Universitetet
i Stavanger
FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study program/specialization: Information Technology - Automation and Signal Processing	Spring semester, 2018 Open/~~Confidential~~
Author: Ørjan Malkenes	Ørjan Malkenes (signature of author)
Instructor: Professor Kjersti Engan Supervisor: Professor Kjersti Engan	
Title of Master's Thesis: Image processing on histopatological images of urothelial carcinoma – assessment of immune cells Norwegian title: Bildebehandling på histopatologiske bilder av urotelial karsinom - vurdering av immunceller	
ECTS: 30	
Subject headings: Urothelial carcinoma, Histogram features, Local Binary Pattern, Chi-Squared, Synthetic Minority Over-sampling Technique, K-means clustering, Support Vector Machine	Number of pages: 53 + supplemental material/other: 6 +embedded files Stavanger, 15th of June 2018

Image processing on histopathological images of urothelial carcinoma – assessment of immune cells

Ørjan Malkenes

June 2018

MASTER'S THESIS

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

Supervisor: Professor Kjersti Engan

Abstract

Bladder cancer is the 6th most common cancer in the world, where urothelial carcinoma is the most common one. Bladder cancer is one of the most economically expensive cancers to treat, as follow up is needed over a long period of time. Through extensive research, it has been indicated that the amount of tumor infiltrating lymphocytes(TIL) can have a positive impact on the relapse rate in conjunction with treatment.

This paper concentrates on image processing to identify, and analyze the amount of TIL cells in histological images of bladder tissue. The objective of this thesis is to locate all cells in a histological image, and to train a classifier to predict if a cell is a TIL or not. The end goal is to automatically determine the amount of TILs in an image which in turn can be used to predict the effectiveness of cancer treatment. A sub set of microscopic tissue samples has been derived from digitized samples, made available by Stavanger Universitetssykehus, to be able to analyze the quantitative performance of the proposed system.

Using a distance transform, in conjunction with pre-processing methods, to 93% of the cells in the histological images were found. A side effect was that there were wrongly located multiple cell centers for some cells, in addition to other non-cell objects in the histological images. Prediction of the located cells, using histogram features, was able to achieve 92% accuracy. Using local binary pattern features, the prediction accuracy was reduced to 73%. Synthetic over-sampling was introduced as the prediction showed a higher accuracy for correctly predicted non-TILs, but this proved to decrease the quantitative performance.

Preface

This thesis marks the end of two great years at the University of Stavanger.
I would like to thank Professor Kjersti Engan for her much appreciated advise and feedback throughout this period.
I would also like to thank Emiel Janssen and Vebjørn Kvikstad at Stavanger University Hospital for medical input and evaluation of the result.
Finally want to thank my family, and especially my partner Veronica for motivating and supporting me.

Contents

List of Figures

List of Tables

Abbreviations

BCG - Bacillus of Calmette and Guerin
HES - Haematoxylin-Erythrosine-Saffron
FRS - Fast Radial Symmetry
NMIBC - Non-Muscle-Invasive Bladder Cancer
PDF - Probability Density Function
SVM - Support Vector Machine
TIL - Tumor Infiltrating Lymphocytes

1 Introduction

With estimated 429.000 new cases in 2012, bladder cancer is the 6th most common cancer in the world[1]. In the US, the estimated number of new cases of bladder cancer in 2018 alone are 81.190, and 17.240 deaths [2]. There are several types of bladder cancer, where urothelial carcinoma is the most common [3].
Bladder cancer is one of the most economically expensive cancer type to treat, as follow up is needed over a long period. Almost half of the patients with NMIBC experience recurrence which need repeated treatment[4].

As treatment for bladder cancer, BCG is most commonly used which induces a general immunal response [5]. There has been done extensive scientific research on this area, and the amount of tumor infiltrating lymphocytes are indicated to have a positive impact on the recurrence and occurrence of bladder cancer [6].

Presently, the assessment of tumor infiltrating lymphocytes is done visually. By an evaluation of histological images, an overall grade is given from 0-100%, and the number of tumor infiltrating lymphocytes is guessed. In an image that contain thousands of cells, this procedure lacks accuracy.

The motivation of this thesis is to create a quantitative measure of tumor infiltrating lymphocytes in histological images with the use of image processing and machine learning. To get this procedure automated would save both time and resources in addition to increase the accuracy in the assessment. By using segmentation methods to locate the cells, and derive characteristics from each cell, the objective of this thesis is divided into two main parts:

- Locate the cells in histological images.
- Train a classifier to predict if a located cell is TIL or not.

Figure 1.1 illustrates the difference between a tumor infiltrating lymphocyte and a epithelial cell in a histological image.

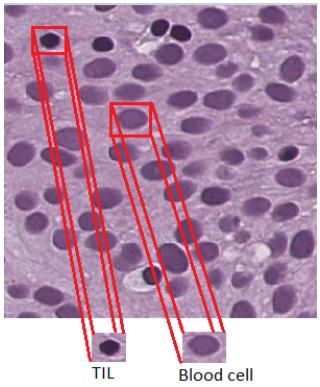


Figure 1.1: Illustration of a tumor infiltrating lymphocyte(TIL) and a epithelial cell in a histological image

An analysis of the cells will be performed before classification based on their features. The end goal is to be able to calculate the amount of tumor infiltrating lymphocytes in relation to epitheilial tumor cells.

1.1 Data material

This section describes the material used in this thesis.

1.1.1 Histological images

The available material for this thesis are histological images of bladder tissue from patients with bladder carcinoma. Samples of bladder tissue from patients with carsionoma has been digitized by The Department of Pathology at Stavanger University Hospital, using a SCN400 histological slide scanner from Leica [7], which then are stored as SCN files.

To view the .SCN files it is necessary to use ImageScope SCN viewer by Leica, or other SCN viewer programs. By using "Aperio Imagescope"[1], the digitized samples can be viewed as shown in figure 1.2, where one can zoom in to 40x magnification.

The samples available have a various degree of contrast and range of color, from sample to sample as illustrated in figure 1.3, which is a result of the amount of HES [2]used in the digitization step. Also, the thickness of the cells can vary, which impacts the color variations.

1.1.2 Extracting desired areas

As this thesis depend on distinguishing between cells, a high magnification of the histological slides is needed. A set of steps has been performed to get the set of images that are used in this thesis. By firstly using the mentioned "Aperio Imagescope", a new set

[1]Aperio Imagescope, developed by Leica, is an open available program for viewing SCN files,
[2]Used as plasma volume expanders because of their colloidal properties.

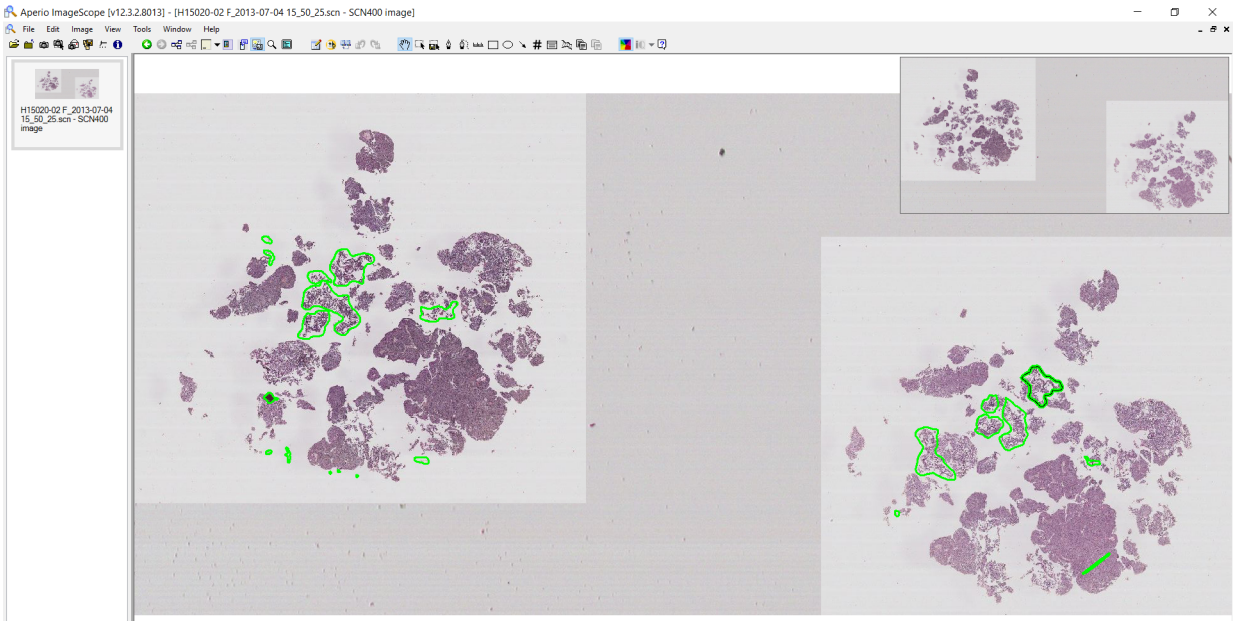
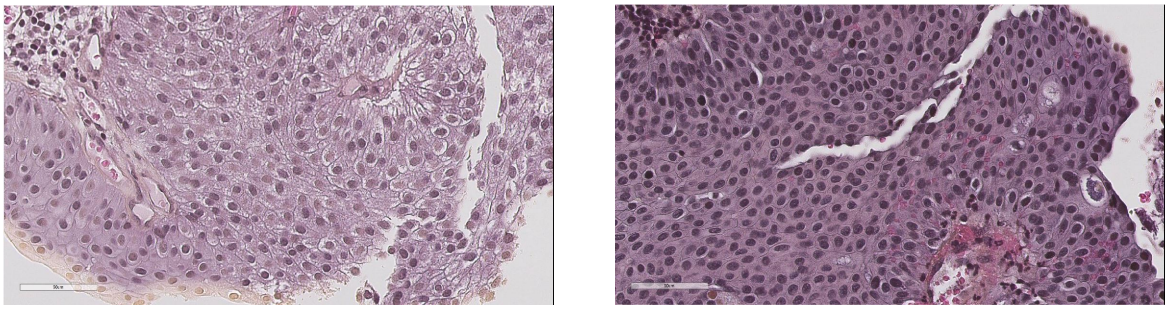


Figure 1.2: Histological images in "Aperio Imagescope"

of images has been derived from a selection of the histological tissue samples. Areas are zoomed in on, at a 40x magnification, and snapshots of these areas are stored as JPEG files.

The resulting set of of the zoomed in images are of size 938x1716 pixels, and are detailed enough to separate the cells. To ensure a representative data set, the selected images has varying contrast and darkness, and also a varying amount of TILs. As the images are stored as JPEG's the images can be processed in MATLAB.



(a) Histological image of bladder tissue

(b) Histological image of bladder tissue

Figure 1.3: 40x magnitude view of histological slides extracted with ImageScope.

As can be seen from figure 1.3, the new set of images contain a large amount of cells. To be able to count and label the cells, a second step of cropping the 938x1718 sized images. Region of interests of 300x300 is extracted from the top left corner from each of the 938x1716 pixel images as shown in figure 1.4.

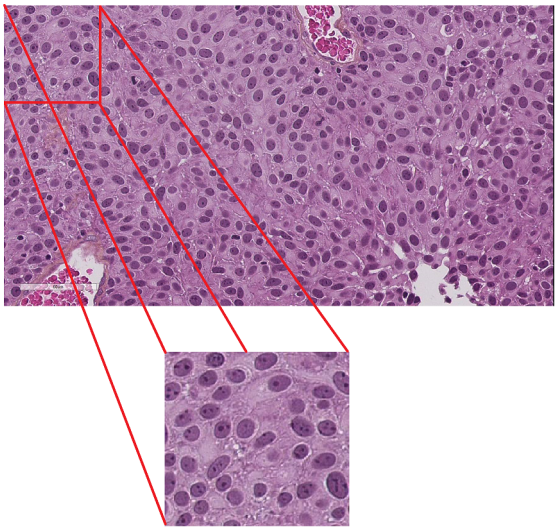


Figure 1.4: Image processing, going from 938x1716 to 300x300

It is from these 300x300 pixel images that the cells are located, and further classified.

1.2 Thesis Outline

Chapter 2 - Background:
Background theory of urothelial carsinoma and some of the functions used in this thesis.
Chapter 3 - Method:
The proposed system is explained. This chapter also contain the implementation of the proposed system.
Chapter 4 - Experiments and results:
Presentation of the experiments and results of the segmentation and classification.
Chapter 5 - Discussion:
Discussion of materials, methods and results.
Chapter 6 - Conclusion:
A conclusion made based on the results.

Appendix A - Matlab code:
The codes are found in the embedded file matlabfunctions.7z, in addition to csv files of features.
Appendix B - Training images:
Images used for testing can be found in the embedded file "testimages.7z".
Appendix C - Test images:
Images used for testing can be found in the embedded file "testimages.7z".
Appendix D - Result experiment 4:
Visual result of the experiment 4.
Appendix E - Training images:
Evaluation of the accuracy given by pathologists at Stavanger Universitetssykehus.

2 Background

2.1 Urothelial carcinoma

Carcinoma in the urothelial arises from the urothelial lining. The tumors are most common along the lateral walls, though it can be found anywhere within the bladder [8]. Different stages of the cancer is described with the Tumor Node Matastasis classification system (TNM). The tumors is described with pTa and pT1 for non-muscular invasive stage, and pT2, pT3 and pT4 for muscular invasive stage. The T describes the tumor size, and the prefix after T describes how far the invasion into the tissue is. pT2 is invasion to muscle, pT3 is invasion to fat and connective tissue and pT4 is invasion to other surroundong organs [3].

Urothelial carcinoma is the most common type of bladder cancers, although it can differ between sexes and in regions [3]. For more developed regions the age standarised rate (ASR) of 16.9 per 100 000 inhabitants is reported for men, campared to 3.7 for women. In 2012, 1347 new cases of bladder cancer was found in Norway [9]. This results in a higher ASR for men, relative to other comparable regions, was registered of 21.6. Figure 2.1 illustrates the ASR of the worls regions, with the darkest blue is where the highest ASR.

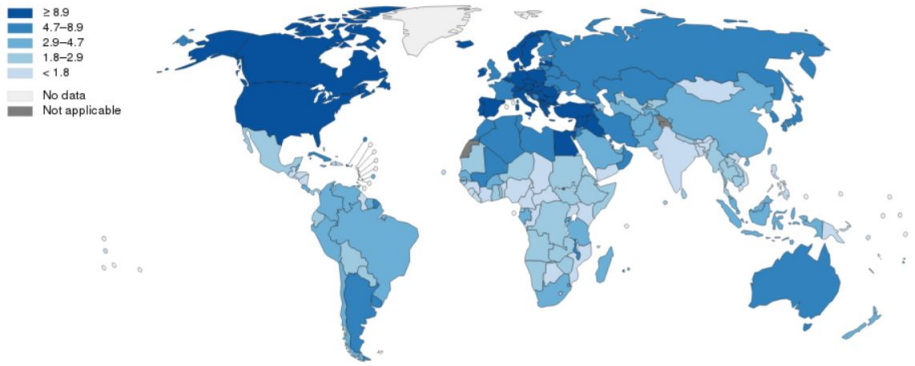


Figure 2.1: Bladder cancer incidence
[10]

2.1.1 Treatment

The treatment of urothelial carcinoma is commonly done by BCG [6]. With the treatment, risk of local recurrence is reduced by about 60%, and 5-year survival rate of about 90% in certain cancer patients. The basis for the effectiveness is thought to be the introduction of TILs, to help induce a immune response.

TILs is a type of white blood cells, of the hosts immune system found around between the tumor cells. In a new treatment, the amount and type is an important factor

to modulate the immune system such that the tumor cells are found an faught [11].

In a histological image they can be seen as smaller and darker in comparison to epithelial cells, illustrated by figure 1.1.

2.2 Histogram equalization

In image processing, histogram equalization is a widely used image enhancement method used for increasing contrast in an image. By stretching out the histogram of the image, It is used in various applications, e.g. medical image processing and radar signal processing [12].

For a given image X(i,j), with L gray levels denoted as $(X_0, X_1, ..., X_{L-1})$, a probability density function $p_X(k)$ for the distribution of the gray levels in the image can be defined as

$$p_X(k) = \frac{n_k}{n}, \qquad k = 0, 1, ..., L-1 \tag{2.2.1}$$

where n is the total number of pixels in image X, and n_k is the number of pixels with value k. For a normal gray level image, L are 256. The cumulative density function for X, $c_X(x)$, can then be defined from the probability density function as

$$c_X(x) = \sum_{j=0}^{k} p(j) \tag{2.2.2}$$

The gray levels of X are then mapped into the entire dynamic range, $(X_0, X_1, ..., X_{L-1})$ by using a transform function, f(x). The cumulative distribution function are used to define f(x) as

$$f(x) = X_0 + (X_{L-1} - X_0)c_X(x) \tag{2.2.3}$$

The equalized output image **Y(i,j)** can then be expressed as

$$Y(i,j) = f(X(i,j)), \qquad \forall X(i,j) \epsilon X \tag{2.2.5}$$

An example of histogram equalization are shown in figure 2.2 and 2.3. Figure 2.2 shows an image before histogram equalizing with its corresponding histogram and cumulative probability distribution, c_X, and figure 2.3 shows the resulting image.

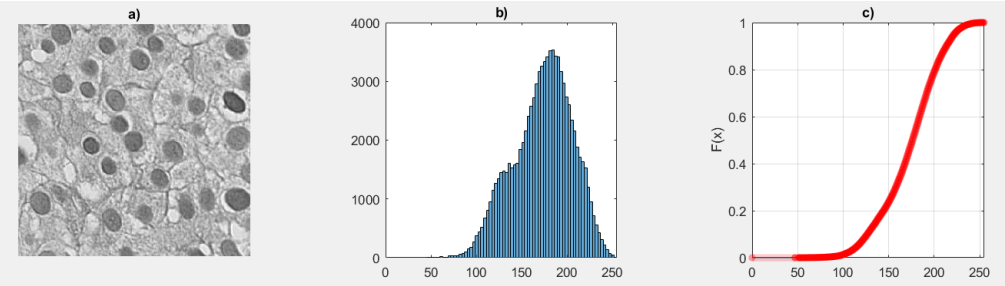


Figure 2.2: Histogram equalizing, example. a): Grayscale image, b) Histogram of the image, c) Cumulative probability distribution

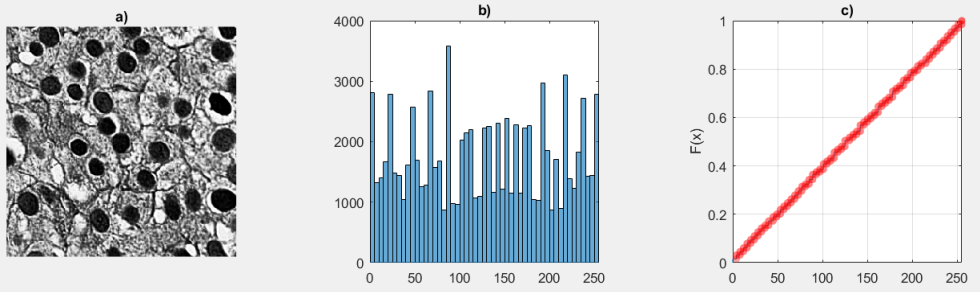


Figure 2.3: Histogram equalizing, transformed image. a): Equalized image, b) Histogram of equalized image, c) Cumulative probability distribution

The results of histogram equalization is an overall enhanced contrast, making it easier to distinguish between objects and background as seen in figure 2.3 a). The cumulative distribution function of the equalized image has a linear shape, illustrating that the number of pixels are similar for all gray values.

2.3 Smoothing filter

Smoothing filters are often used as a method of denoising images. There are several methods developed to remove noise, but commonly it is achieved by averaging. A widely used method for smoothing an image is a locally averaging over a set of pixels in an image with the use of a Gaussian filter[13].

2.4 Thresholding of an image

Binarization is a popular and simple way to segment an image. When binarizing an image a threshold value is used to classify the image pixels into one of two classes. The threshold value can be used for the whole image which are effective and often sufficient.

For an input image I a pixel within the image. I(x,y), is assigned to 0 or 1 based on:

$$I_b(x,y) = \begin{cases} 1, & \text{if } I(x,y) >= T \\ 0, & \text{otherwise} \end{cases} \tag{2.4.1}$$

where I_b is the resulting binarized image and T is the threshold value.

The threshold value can be set manually, or one can use Otsu's method to automatically calculate the optimal value [14].
Otsu's method uses exhaustively search to find a optimal threshold as the threshold that maximizes the within-class variance given by:

$$\sigma_\omega^2 = \omega_0\sigma_0^2 + \omega_1\sigma_1^2 \tag{2.4.2}$$

where ω_0 and ω_1 is the the class probability given by:

$$\omega_0 = \sum_{i=1}^{t} p_i, \qquad \omega_1 = \sum_{i=t+1}^{L} p_i \tag{2.4.3}$$

and variance σ^2 for each class is given by:

$$\sigma_0^2 = \sum_{i=1}^{t}(i-\mu_0)^2\frac{p_i}{\omega_0} \qquad \sigma_1^2 = \sum_{i=t+1}^{L}(i-\mu_1)^2\frac{p_i}{\omega_1} \tag{2.4.4}$$

In the equations 2.4.3 and 2.4.3, L is the number of gray levels in the image with a normalized histogram p, t is the threshold value, and μ is the mean of the classes given by:

$$\mu_0 = \sum_{i=1}^{t}\frac{ip_i}{\omega_0} \qquad \mu_1 = \sum_{i=t1}^{L}\frac{ip_i}{\omega_1} \tag{2.4.5}$$

2.5 Distance Transform

The distance transform is a simple method, often used in image segmentation. The method is usually applied on a binary image, consisting of ones and zeros. The method calculates the distance from a non-zero pixel to the nearest boundary, represented by a zero pixel[15], or the other way around.

The distance can be found with various metrics where the method is different. Figure 2.4 illustrates how distance is found with Euclidean distance in green, and a squared distance in red. We see that the length of the green arrow is smaller than the red arrow, showing that Euclidean measures the shortest path.

Figure 2.5[1]. illustrates a distance transform of a binary image, I, of a squared object where the distance is calculated squared. The value of each pixel, I(x,y), is replaced

[1]Reprinted under the terms of the licence GNU Free Documentation Licence, Version 1.2

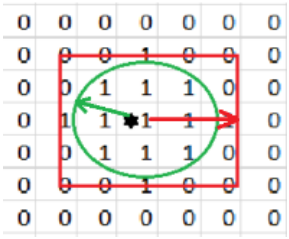


Figure 2.4: Distance metrics, green = Euclidean, red = squared.

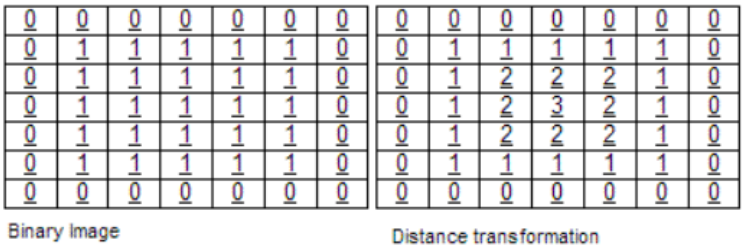


Figure 2.5: Distance transform applied to a image
[16]

by the distance to the nearest boundary, D. Squared distance, D_s counts the number of pixels on each axis resulting in an integer value for each pixel. Euclidean distance, D_e between (x_1,y_1) and (x_2,y_2) is given by :

$$D_e = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{2.5.1}$$

As the transform only locates the closest boundary pixel, a single pixel of this sort can have a great impact on the calculated distance.

2.6 Local Binary Pattern

Local binary pattern is used to describe texture of an image. Originally, it was introduced by Ojala et al.[17] to describe local binary patterns in a texture for a 8 pixel neighbourhood. It assigns a binary number to a pixel, as a measure of texture, by comparing it to each of its neighbourhood pixels. For each comparison, if the neighbourhood pixel has greater value, a 1 is assigned and 0 if not. The assigned values are then added to make a binary number. When using 8 neighbourhood we can get $2^8 = 256$ different values for the texture measurement.

Improvements of the algorithm has been made to deal with larger texture structures. Instead of comparing it to its 8 connected neighbourhood pixels, a circle is set with radius = r and an assigned number of pixels, p, are set on that circle. It is then compared to its p circular neighbourhoods and assigned a binary label. When all of the pixels are combined, the labels that are assigned is put together to make a binary number.

2.7 Chi-squared

Chi-squared [18], denoted as $\tilde{\chi}^2$, is often used to compare if the variation of the data is due to chance, or if it is due to the variable that is tested. It is a test for checking the null-hypothesis as:

$$\tilde{\chi}^2 = \sum_{i=1}^{N} \frac{(O_i - E_i)^2}{E_i} \tag{2.7.1}$$

Where N is the number if bins in the histogram.

From the Chi-squared test statistics we can derive the Chi-squared distance, which can be used to compute the difference between a observed histogram, O, and a model histogram, E [19]. The distance is computed as:

$$d = \sum_{i=1}^{N} \frac{(O_i - ((O_i + E_i)/2)^2}{\frac{O_i - E_i}{2}} \tag{2.7.2}$$

2.8 Classification

When using machine learning, the goal is to train a classifier to be given a data vector, and to recognize what class it belongs to based on the information it holds.

The most beneficial classification method to use varies based on the information in the data set. In cases where the data set are labeled, supervised learning can be utilized by training a classifier, where the features of a sample can be mathed to a label. In other cases the data set are unlabeled, thus no features can be matched to a label. The use of unsupervised learning is needed in these cases.

The following sections explains some of the different methods for supervised and unsupervised learning.

2.8.1 Clustering

Clustering is a method for unsupervised learning which groups samples, denoted by S, in a data set. The goal is to assign a label, or a group, to each sample based on its features.
In an example with two groups, g_1 and g_2, a clustering method assigns the samples to the groups so that all samples within group g_1, $S(g_1)$, are more similar to each other, in comparison to samples in group g_2, $S(g_2)$[20].

Though most clustering methods have the same goal, the algorithm of how it is done varies.

K-means:

.

K-means, also known as Lloyd's algorithm [21], is one of the most common and widely used clustering algorithms. The k-means is a fast and simple algorithm that preforms iteratively, and assigns each sample in a data set to a group or cluster, based on its distance to the cluster centers.

Given a data set where we want to group the samples, S into K groups. Initially K centroids or cluster centers are placed randomly. By computing the Euclidean distance from a sample,S_i, to each cluster center, the sample are assigned a label corresponding to the closest cluster center.

In the next iteration, when all samples are assigned to a cluster, new cluster centers are computed as the average location of the samples in the cluster. With cluster centers, the distance from each sample to the new cluster centers are then computed, and the samples are assigned a label in the same manner as previous.

The algorithm for k-means clustering is shown in algorithm 1.

Algorithm 1 K-means clustering

1: Choose K number of centers,(c1,...,ck), and randomly select their location
2: Assign a label to each sample based on the nearest cluster center, where the distance metric is Euclidean
3: Compute cluster centers based on the samples assigned to each cluster.
4: Re assign the samples to their closest cluster center.
5: Repeat step 2 and 3 until no change in cluster centers.

A down side of Lloyd's algorithm is that it requires many iterations before no change in in cluster centers is achieved. Also, it is sensitive to its initialization when it places the cluster centers. Since it initializes k random centers, it can possibly get two centers

in the same cluster.

K-means++

The *k-means++* [22] is presented by Vassilvitskiiy and Arthur and as an improvement on the *k-means* algorithm. The algorithms works similarly, where the difference is in the method to choose the cluster centers.

K-means++ can be considered as an initialization step for Lloyd's algorithm where it spreads out the cluster centers by placing one center at the time. It chooses a sample from the data set as one center, c_1, and computes the distance, d, from each sample to c_1. The next centers c_i, for $2 <= i <= k$, is chosen as another sample from the data set based on a weighted probability function, to ensure that the centers is not placed in the same cluster. When k centers is chosen, the algorithm proceeds with Lloyd's algorithm [23].

Algorithm 2 K-means++ clustering

1: Given K number of clusters, with cluster centers denoted as C=(c1,...,ck): Choose an initial center, c_1, uniformly at random from dataset.
2: Compute distances from each sample, m, to $c_1$denoted as d(x_m, c_1).
3: For $2 <= i <= k$, select the next centroid, c_i, as a sample from the data set with probability

$$\frac{d^2(x_m, c_p)}{\sum_{h;x_h \epsilon C_p} d^2(x_h, c_p)} \tag{2.8.1}$$

Where Cp is the set of all observations closest to centroid cp, and xm belongs to Cp.
4: Repeat step 3 until k centroids are chosen

2.8.2 Support Vector Machine

Support Vector Machines(SVMs) are a supervised learning method designed for separating two classes, with a low- to moderate-dimensional data sets. It uses a learning approach developed by Vapnic et al.[24], where it computes an optimal hyperplane, with the use of support vectors to separate classes. Thus it is a linear classifier. Figure 2.6[1]. illustrates optimal hyperplane where the samples on the margin are called support vectors.

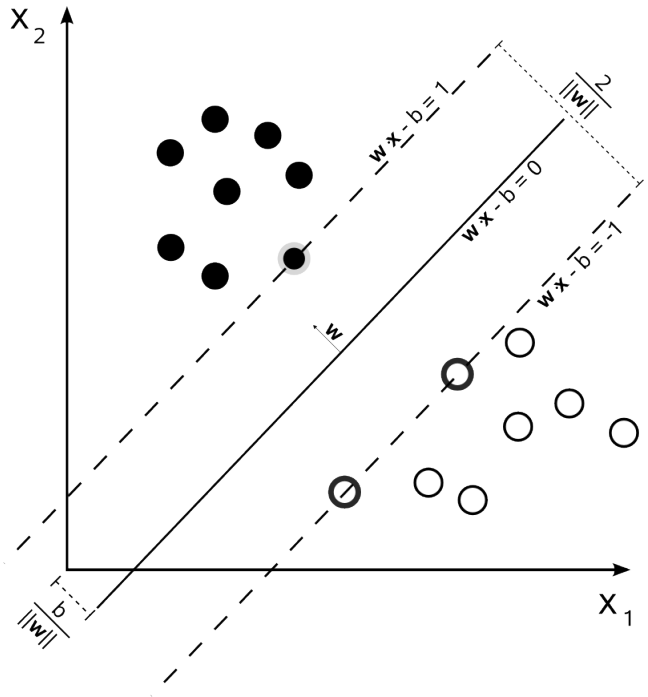


Figure 2.6: Illustration of an optimal hyperplane
[25]

By fixing the margin to 1 on either side of the decision boundary as done in figure 2.6, it can be shown that the optimal hyperplane, defined as the one where the distance between the two outer lines wx-b = 1 and wx-b = -1 are maximalized, is given as:

$$\gamma = \frac{2}{\|w\|} \tag{2.8.2}$$

For linear inseparable classes, some of the samples will overlap, and a linear hyperplane cannot separate all the samples. In this case, the SVM assigns a penalty to the samples on the wrong side of the decision boundary given by:

$$f(x) = x'w + b \tag{2.8.3}$$

where f(x) is the length from an observed sample x to the hyperplane [26].

[1]Reprinted under the terms of the creative commons licence

2.8.3 Class imbalance

As mentioned in the previous chapters, we want to train a model that can predict which class a data vector belongs to. When we train a classifier, we often use a labeled data set where all classes are represented.

However, in machine learning where one class heavily outnumbers another, the learning algorithm can have a problem learning. In these cases the classification tends to be biased, and favour the biggest class, which leads to misclassification [27]. If a data set with two classes represented by 90% of one class and 10% of the other, a simple way of getting 90% accuracy is to set all samples in the data set to the biggest class.

To cope with this problem a number of methods have been proposed.

Over-sampling

One way of tackling the class imbalance problem is to over-sample by duplicating the the minority class samples. This however can lead to over-fitting and is shown to have little effect on the preformance as no additional information is produced [28]. An alternative to over-sampling by duplicating existing ones is SMOTE [29].

SMOTE stands for Synthetic Minority Over-sampling Technique and the aim is to create new synthetic samples of the minority class. The new synthetic samples are created to lie in the same area as the minority class area, rather than duplicating the existing ones. SMOTE operates by taking all samples of the minority class and one by one finding its K nearest neighbour samples of the same class.

Imagine that the SMOTE creates linear lines between a existing minority class samples, $S_{minority}$ and its k-nearest neighbours. The algorithm then loops through these lines, and creates new samples, $S_{minority,s}$ somewhere along it. This is done for each minority class sample, creating new synthesized samples lying in the same area as the minority class samples. SMOTE can vary between the number of neighbours found, and also how many new points along the lines it will create. The resulting data set is balanced between the classes, where the total amount of minority class samples is $S_{minority} + S_{minority,s}$.

Algorithm 3 SMOTE

Given the number of minority instances, T, the number of new synthetic samples to be created, N, and the number of nearest neighbours, k:
for i = 1 to T
    Find the k nearest neighbours for i, an store the indices of these
end for

for N random neighbours of i
    Compute difference for each dimantion between sample i and k_N as d. "d" is a data vector of same size as sample i
    Make a vector r, of same size as i, with random values between 0 and 1
    Synthetic samples = d .* r
end for

Under-sampling

An alternative to over-sampling is under-sampling. A balanced data set can be obtained by neglecting enough of the majority class samples $S_{majority}$. By doing under-sampling to have equal amounts of each class, the data set will be reduced by a factor of $S_{majority}/S_{majority}$.

2.9 Preformance measurements

Evaluation of machine learning algorithm are typically done by using a confusion matrix [29].

	Predicted A	**Predicted not A**
True A	True positive	False negative
True not A	False positive	True negative

Figure 2.7: Confusion matrix for a 2-class problem

Where true positive(TP) and true negative(TN) are the number of correctly classified elements. False positive(FP) and false negative(FN) are the number of falsely classified elements.

From the confusion matrix there are multiple measurements that can be defined. The most common measurement of a machine learning algorithm is the predicted accuracy, defined by:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{2.9.1}$$

We can also define true positive rate (TPR) which is the proportion of correctly identified examples in class A, and true negative rate(TNR) which the proportion of the correctly identified examples in class not A. These are respectively defined as:

$$TPR = \frac{(TP)}{(TP + FN)} \tag{2.9.2}$$

Also known as sensitivity.

$$TNR = \frac{(TN)}{(TN + FP)} \tag{2.9.3}$$

also known as specificity.

3 Method

This chapter presents and explains the methods used in the proposed system.

An overview of the proposed system is shown in figure 3.1. Each of the steps will be explained in more detail. This chapter will first explain the pre-processing steps which is divided into two blocks as they are used for different purposes. Further the seed point extraction, extraction of features and classification is explained, before the final proposed system is presented.

Finally it is explained how the proposed system is implemented in Matlab.

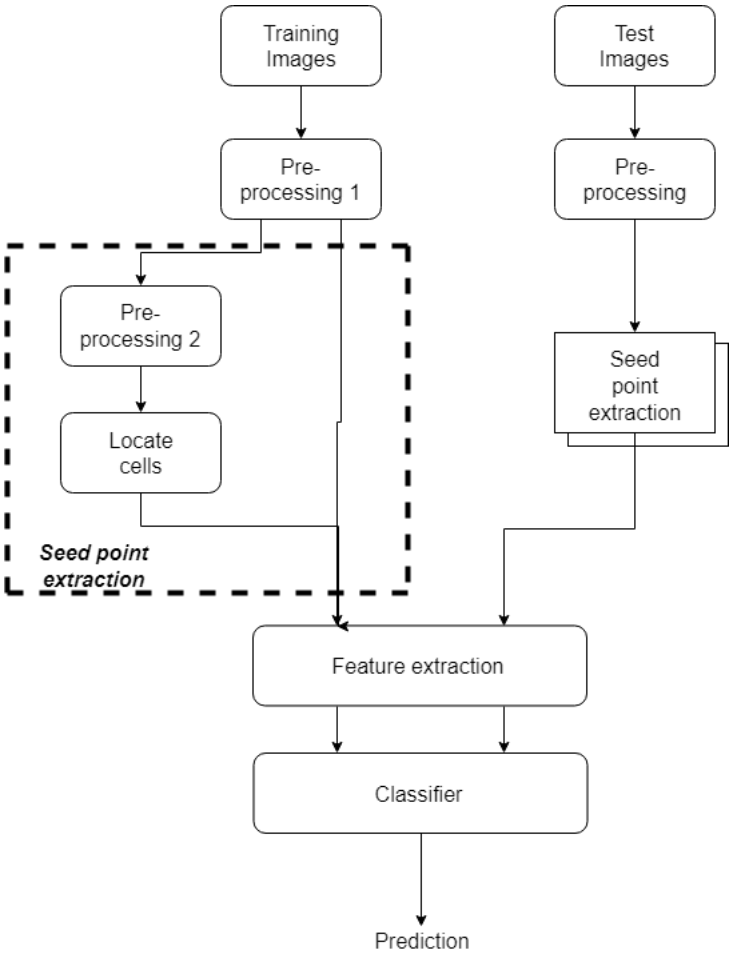


Figure 3.1: Overview of the proposed system

3.1 Pre-processing 1

This chapter explains the "Pre-processing 1" step in figure 3.1.

As illustrated in figure 3.2, we see that the images, I, have variations in brightness and contrast, where the cells in the left image presents as overall darker than the left

image. This can be caused by variations in the amount of HES used, the thickness of the sample, or a combination of the two.

As these images are used to extract features, it is wanted to minimize these variations, but not lose significant information. By converting the images, I, from RGB to grayscale, only the intensity of the images is carried out. The resulting grayscale images, I_g, are used for the feature extraction, but also further processed to locate the cells.

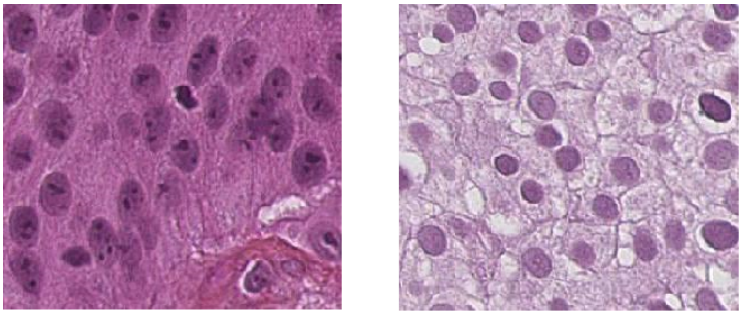


Figure 3.2: Color variations of the images

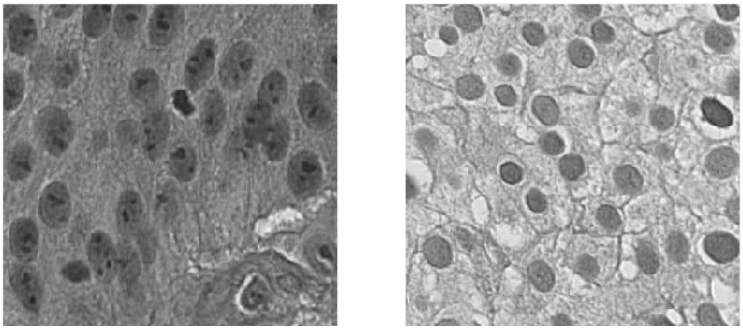


Figure 3.3: Grayscale converted images

For the seed point extraction it is also considered to use the single color channels of the the RGB image I, illustrated in figure 3.4.

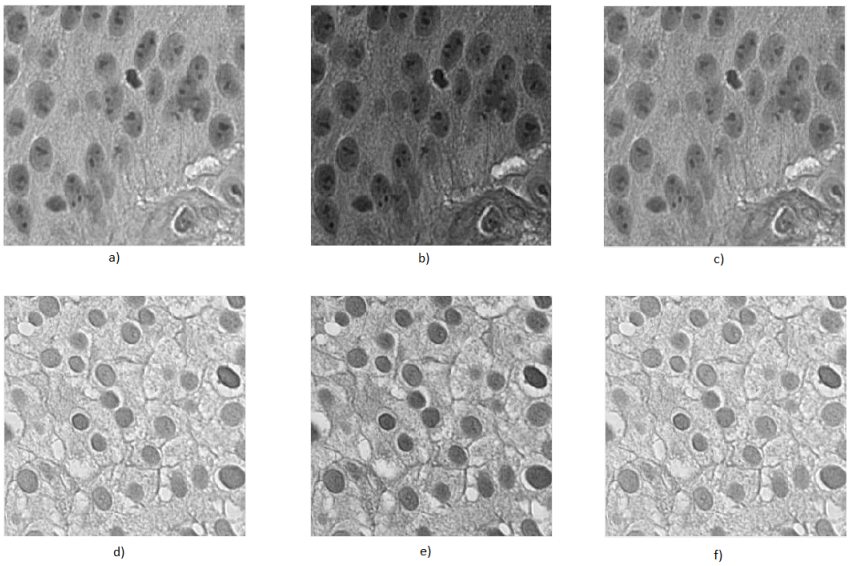


Figure 3.4: Images in the RGB color channels: a) and d) = red channel, b) and e) = green channel, c) and f) = blue channel

3.2 Pre-processing 2

Before any features of the cells can be extracted from the grayscale images, I_g, they first need to be located. This is done by finding seed points, denoted as SP, that represents the centers of the cells. This chapter explains the "Pre-processing 2" block in figure 3.1, which is used in conjunction with the seed point extraction.

3.2.1 Histogram equalization

To locate the seed points, it is first needed to segment the cells from the background tissue. As we can see in figure 3.3 and 3.4 the cells are distinguishable from the background tissue in varying degrees, both within each image, and across all images, because of the factors mentioned in chapter 3.1. To eliminate most of these variations, a histogram equalization(HE) is utilized to normalize the grayscale images I_g.

The histogram equalizing uses equations 2.2.1 to 2.2.5 to enhance the global contrast in the images. As illustrated in figure 3.5 the distinction between cells and background tissue in the histogram equalized images, $I_{g,HE}$, is increased. A distinction between epithelial cells and TILs is not needed in $I_{g,HE}$ as this step is only to locate the cells and no features are going to be extracted here.

As mentioned in chapter 3.1 it has also been considered to look at the histogram equalized images of single color channels in the RGB images. $I_{red,HE}$, $I_{green,HE}$ and $I_{blue,HE}$, are processed in the same way as $I_{g,HE}$ to see if the segmentation and seed point extraction are improved.

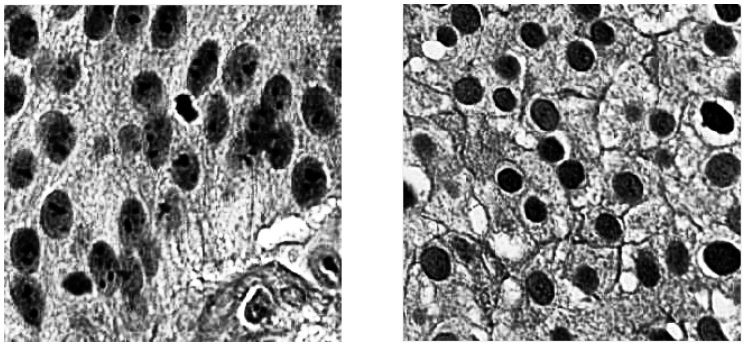


Figure 3.5: Histogram equalized grayscale images

3.2.2 Gaussian smoothing filter

A side effect of the histogram equalization is that potential variations of darkness within each cell is increased. This results in bright pixels within cells as illustrated by the red circles in figure 3.6, which will be segmented away in a binarization.

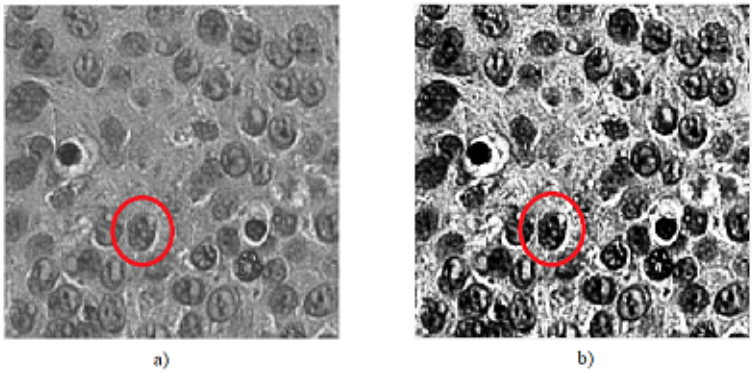


Figure 3.6: Side effect of histogram equalization. a) Grayscale image, b) Histogram equalized image

As explained in chapter 2.5, the distance transform which is used to find the seed point, is sensitive in regards to blank/edge pixels in a binarized image. With this in mind, a Gaussian smoothing filter is run over $I_{g,HE}$ as explained in chapter 2.3. The resulting pre-processed image after the Gaussian smoothing, I_p, is a a blurred version of $I_{g,HE}$, where the bright areas within the cells are evened out. An added benefit by averaging over regions in the image, is that some of the small non-cell objects will be segmented out.

3.3 Locating seed points

The next step is to locate seed points that correspond to the cell centers. To find the actual seed points from the pre-prosessed images I_p, a binarization step is used to segment the cells from the background tissue, before a distance transform is implemented.

3.3.1 Binarization

In I_p, the cells presents as much darker than the backround tissue. To segment the cells, a binarization approach has been used where all pixels in the image, I_p(x,y), are set to zero or one based on a threshold explained in chapter 2.4. To choose the threshold value, both Otsu's method for each individual images, as explained in equations 2.4.3-2.4.5 and a fixed value for all images are considered. As a result, we are left with the binarized images, I_b, where the pixels within a cell are set to zero.

In addition to the cells, there are also areas with various dark non-cell objects in I_p, that are too big for the Gaussian smoothing filter to eliminate, which in turn are set to zero. As a solution to this it has been considered the use of morphological operations of the binary image to get rid of these. The cells have a convex shape, an open/close method of I_b was run over the image. This however was not considered to be a sufficient method, as more cells were not found.

3.3.2 Distance transform

The distance transform calculates the distance from each pixel to its nearest boundary. As the cells have an overall circular/convex shape, the max value will typically be at the center of a cell when calculating the Euclidean distance found by equation 2.5.1. Each pixel in I_b is then replaced by their assigned distance as explained in 2.5. Note that in the Matlab implementation, the distance transform calculates the distance from each zero pixel to its nearest non-zero pixel.

Figure 3.7 illustrates the result of the distance transform, where the brightest pixels represents the pixels in I_b with the longest distance to a boundary.

From the transformed images, I_{dt}, the coordinates of the centers of mass can be found as the local maximums. This is done by comparing each pixel I_{dt}(x,y), to its neighbours. If I_{dt}(x,y) is larger than all of its closest neighbours, it is flagged as a local maximum. The set of local maximums in I_{dt} represents the pixels that are furthest away from a boundary, thus representing the centers of the cells or the seed points, denoted as SP.

3.3.3 Removal of unwanted seed points

As a consequence of how the local maximums are found, it can potentially be seed points close together and therefore in the same cell. To prevent this, a removal function is applied where the seed points coordinates are iteratively run through as shown in algorithm 4.

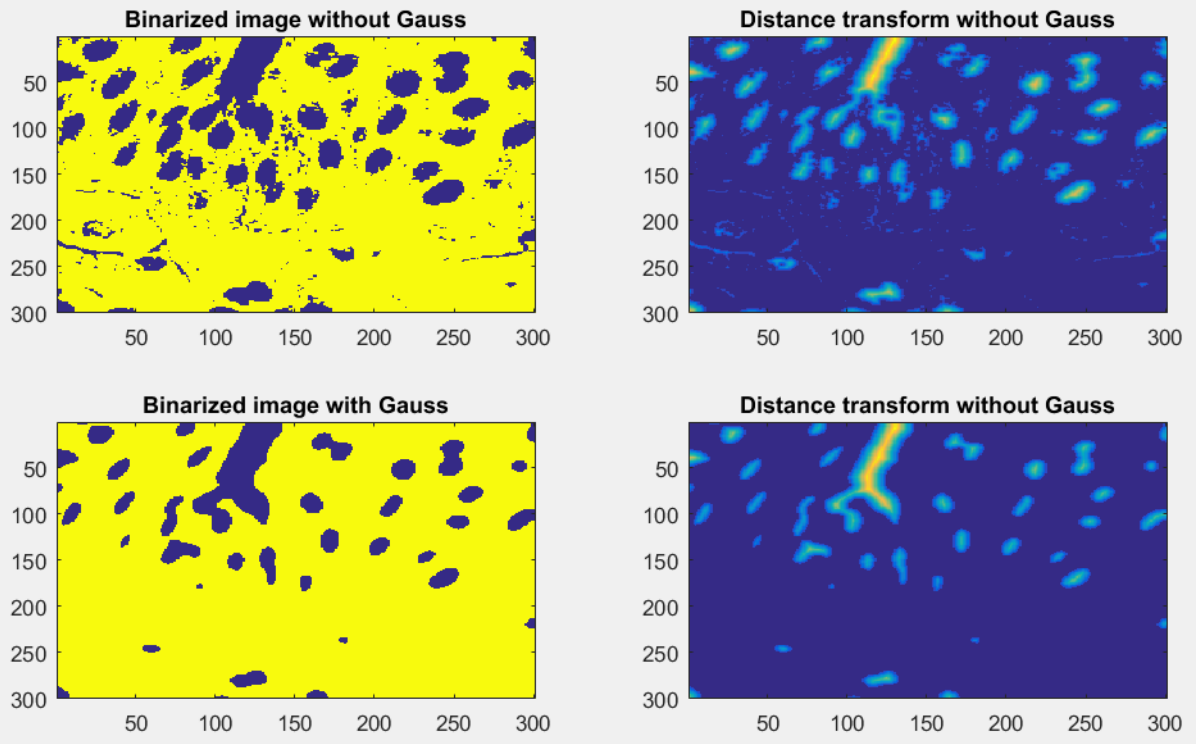


Figure 3.7: Distance transform with and without Gaussian smoothing operation.

Algorithm 4 Removal of close seed points

```
Get coordinates of seed points in I_dt, as SP_i(x, y), where i = 1 → length of SP
if x coordinate of SP_i is closer than (x coordinate of SP_{i+1}) ± 2) then
    if (y coordinate of SP_i is closer than (y coordinate of SP_{i+1}) ± 2) then
        Remove SP_i
    endif
endif
```

In addition, it is beneficial to remove the cells that are at the edge of the image, as these are cropped and in turn are not representative. The cells vary, from about 15 to 30 pixels in both axis' which means that the if a seed point is closer than 15 pixels to any edge can potentially be cropped. To ensure that cropped cells are not accounted for, the seed points that are closer than 15 pixels to the edge are removed. In a 300-by300 image, the removal is done as explained in algorithm 5.

3.3.4 Window of cells

To be able to extract the features a intermediate step is needed. As we now have the SP for each cell sample, a fixed window of size 31-by-31 pixels is placed over the cells with the SP as a center points. Each of these windows are stored to make a data set of cells, denoted as $C_{Ig}(n_{Ig})$, where I_g is the image number, and n_{Ig} is the number of cells in image I_g.

Algorithm 5 Removal of edge seed points

1: Get coordinates of seed points in I_{dt}, as $SP_i(x, y)$, where i = 1 $\rightarrow$ length of SP
2: **if** x coordinate of SP_i <= 15 $\|$ x coordinate of $SP_i(x, y)$ >= 285 **then**
3: Remove SP_i
4: **endif**
5:
6: **if** y coordinate of SP_i <= 15 $\|$ y coordinate of SP_i >= 285 **then**
7: Remove SP_i
8: **endif**

3.4 Feature extraction

After the windows around SP has been stored, the next step is to derive features from the cells. This chapter explains the "Feature extraction" step in figure 3.1.

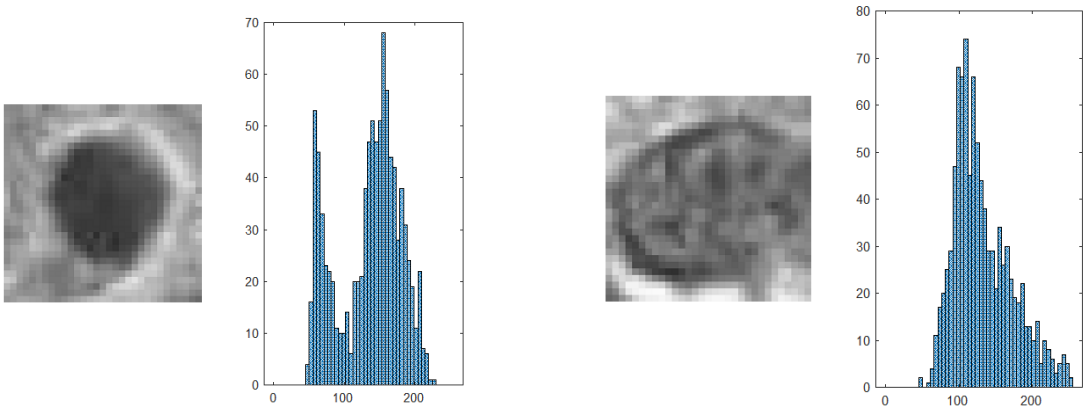
As the cells can vary in size, the fixed window of 31-by-31 pixels can contain varying amount of background tissue which is illustrated by figure 3.8 where the window containing the TIL has more background tissue than the window containing the epithelial cell.

When extracting the features, two different approaches has been considered. One where all pixels of the dilated windows, $C_{Ig}(n_{Ig})$, are included. The other where the background tissue from $C_{Ig}(n_{Ig})$ is segmented out, denoted as $C_{Ig,s}(n_{Ig})$. A binarization of $C_{Ig}(n_{Ig})$ is performed to create $C_{Ig,b}(n_{Ig})$. The pixels in $C_{Ig,b}(n_{Ig})$ that are set to one is the pixels that are removed to create $C_{Ig,s}(n_{Ig})$

By deriving histogram features from the pixel values only belonging to the assigned cell, the background tissue and elements of other cells don't have any effect.

3.4.1 Histogram features

Overall, it can be observed a difference in darkness between TILs and the other cells is the darkness, which is illustrated in figure 3.8. As a result of this, it is possible to assume that features from the histogram of the cells can be efficient, when training a classifier to separate the two classes.



(a) TIL (b) Epithelial cells

Figure 3.8: Histograms of TIL and epithelial cell

Both $C_{Ig}(n_{Ig})$ and $C_{Ig,s}(n_{Ig})$ are rearranged into vectors, denoted as $V_{Ig}(n_{Ig})$ and $V_{Ig,s}(n_{Ig})$ respectively, before deriving the histogram features. The features derived from the histogram is :

Mean:
Mean is a measure of the average value for the histogram, which is derived from $V_{Ig,s}(n_{Ig})$. The mean for vector $V_{Ig,s}(n_{Ig})$ is defined as:

$$Mean = \frac{1}{N}\sum_{i=1}^{N} V_{Ig,s}(n_{Ig})_i \tag{3.4.1}$$

Where N is the number of pixels in $V_{Ig,s}(n_{Ig})$

Variance:
Variance is derived from $V_{Ig}(n_{Ig})$, and is a measure of how far a set of numbers are spread out from their average value [30]. For vector $V_{Ig}(n_{Ig})$ containing N pixels, the variance is defined as

$$V = \frac{1}{N-1}\sum_{i=1}^{N} |V_{Ig}(n_{Ig})_i - \mu|^2 \tag{3.4.2}$$

where μ is the mean of $V_{Ig}(n_{Ig})$ as shown in equation 3.4.1

Skewness:
Skewness is derived from $V_{Ig}(n_{Ig})$, and gives a value for how much the values are spread out to either side. For vector $V_{Ig}(n_{Ig})$, the skewness is defined as

$$S = \frac{E(V_{Ig}(n_{Ig}) - \mu)^3}{\sigma^3} \tag{3.4.3}$$

where σ is the standard deviation of $V_{Ig}(n_{Ig})$, and μ is the mean of $V_{Ig}(n_{Ig})$.

First quartile:
The first quartile value is derived from $V_{Ig}(n_{Ig})$ as the value where the first 25% of the pixels are split from the highest 75%. This can be translated to

$$Q = median(V_{Ig}(n_{Ig})(V_{Ig}(n_{Ig}) < median(V_{Ig}(n_{Ig})))) \tag{3.4.4}$$

Where the median is the value that divides $V_{Ig}(n_{Ig})$ into two parts with equal amount of elements.

Threshold pixels:
As an added histogram feature, the number of pixels below certain threshold in $V_{Ig}(n_{Ig})$, is derived. As illustated by figure 3.8 there are less pixels on the left side of the histogram of the epithelial cells. Through testing, a value of 56 is set as the best discriminative threshold.

3.4.2 Region features:

Region features are used to find regional properties of the cells, by using $C_{Ig,b}(n_{Ig})$, such as the size, shape and axis length of each cell. As seen from figure 3.8, the epithelial cell is larger than TIL, and features describing the size is considered.

Before deriving the region features a intermediate step is done. As the binarization sets the pixels that are within the cell to zero, the complementary image of $C_{Ig,b}(n_{Ig})$ is done. As $C_{Ig}(n_{Ig})$ can contain elements of other cells as a result of overlapping cells and/or the placing of the seed points, a morphological closing is performed, before the areas with connectivity less than 120 is removed.

Area:
Describing the amount of pixels that are within the cells, and are found as the biggest area, in case there are still overlapping cells in $C_{Ig}(n_{Ig})$ that are not segmented out.

Minor/Major axis:
Describing the length of each axis', in pixels.

Differential:
The factor describing the difference between the axis length found as:

$$D = \frac{MajorAxis}{MinorAxis} \tag{3.4.5}$$

3.4.3 Texture features:

There has also been looked at texture features which is used to describe similarity. Due to the size of $C_I(n_I)$, LBP was computed with radius set to one and P=8 circular neighbourhood pixels. From LBP, with a 8 pixel neighborhood we get a histogram with 10 bins for each window, $C_I(n_I)$. A histogram model for both TIL and epithelial cell is computed as the average of all histograms for both classes. Note that this can only be done after the samples are labeled.

With the use of chi-squared distance, the new texture is compared to the computed model, and the difference is calculated as shown in equation 2.7.2.

3.4.4 Normalizing feature vector

The features have values in different ranges and as a result they are weighted differently in a classifier. To cope with this problem, the features are first normalized.

A feature, is normalized as described in equation .

$$f_x(n) - \frac{1}{\sigma_x}(\frac{1}{n}\sum_{i=1}^{n} fx_n) \tag{3.4.6}$$

where σ is the standard deviation and n is the number of features describing the same characteristics. This ensures that all features has the same range and will be weighted equally.

3.5 Classification

After features are extracted from the cells the next step is the classification. This section describes the steps used in the "Classifier" block.

3.5.1 Clustering

As the cells in the images are not labeled beforehand, unsupervised classification with *kmeans++* is utilized.

The k-means++ algorithm is used for two main reasons. The clustering algorithm is used as a help for labeling the data set, but also to get a indication of which features that are suitable to classify the cell samples. A good clustering result will most likely mean that the cells are distinguishable as described with the used feature vector.

With varying feature vectors, the *kmeans++* algorithm is run on the training set and test set separately. As the problem consists of finding which cell are TIL and which is not, the clustering divides the samples described with a feature vector as explained with algorithm 2 with K = 2. Due to that some of the training images does not contain both epithelial cells and TILs, it is needed to implement the clustering algorithm on the entire set of samples to such that the algorithm is not forced to divide same class samples in an image where one of the classes are absent.

When an overall good result is achieved, the labels are overseen visually and any misclassified cell samples is corrected to make a true labeled training- and test set. The features are stored with their corresponding label, image number and coordinates of their seed points.

3.5.2 Support Vector Machine

With the problem is narrowed down to a two-class problem, the SVM is considered to be a suitable classifier for this thesis. Before the SVM can predict if a located cell is a TIL or not, it has to be trained with feature vectors that are assigned a label. With varying combinations of features, the SVM is trained to be able to predict it a new feature vector describes a TIL or not-TIL. As the results from the clustering showed that the samples was not linearly separable, the support vector machine finds the best hyperplane where most of the samples are divided, and assigns a penalty for the misclassified samples.

...

The data set are imbalanced between the two classes, where the normal cells are represented more than TILs. As an imbalanced data set can have a negative impact on the classifier, a sampling method is considered.

Sampling

Due to that the data set has a somewhat limited amount of TILs, an under-sampling

is considered to result in too much loss of samples. As an alternative an oversampling considered . As explained in chapter 2.8.3, an oversampling by duplicating existing samples is shown to not be as effective, so a synthesized set of TIL samples is created with SMOTE as shown in algorithm 3.

3.6 Proposed system

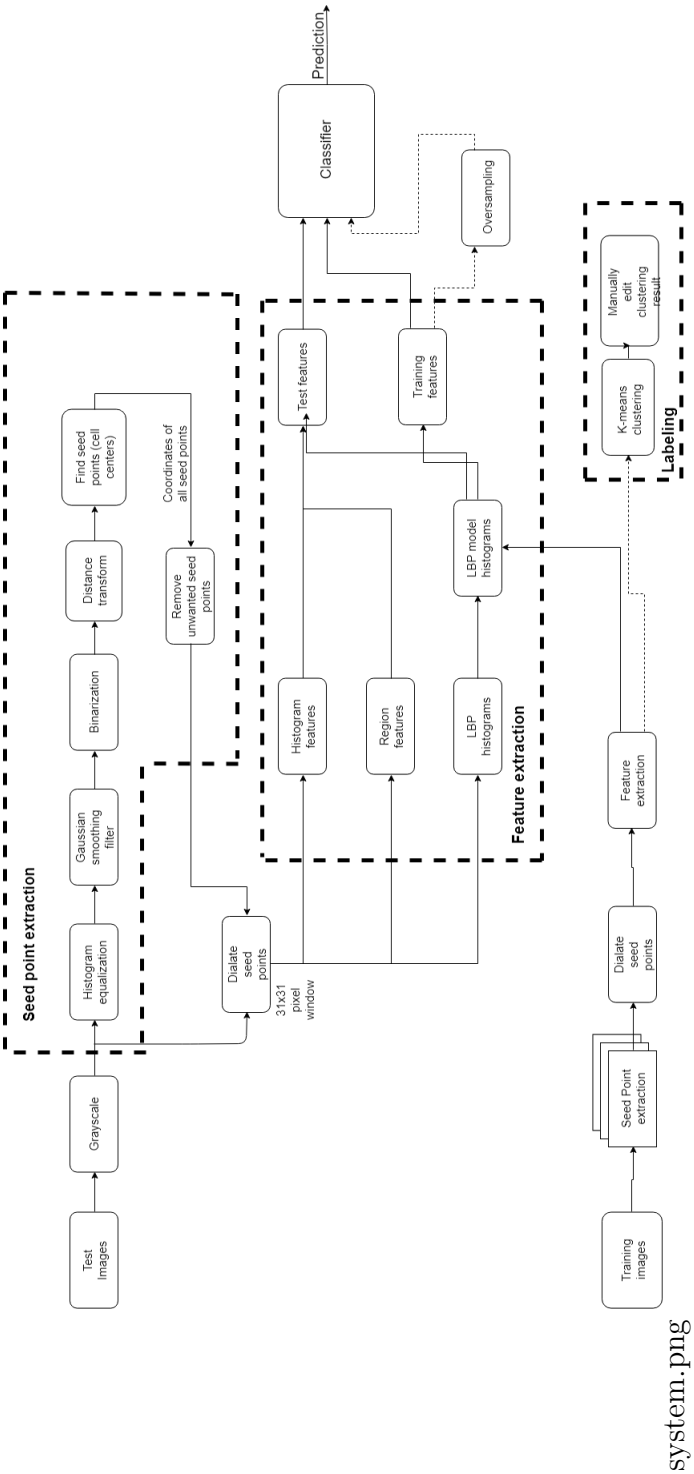


Figure 3.9: Proposed system

3.7 Matlab implementation

The proposed system was implemented in Matlab using a combination of embedded functions, external functions and novel functions.

3.7.1 Pre-processing and seed point extraction

Table 1: Overview of functions that are used for pre-processing and seed point extraction. For the embedded functions, the syntax of the function is in parenthesis

Method	**Embedded**	**External**	**Self made**
Grayscale conversion (rgb2gray)	x		
Histogram equalization (histeq)	x		
Gaussian smoothing filter (imgaussfilt)	x		
Otsu's method (graythresh)	x		
Binarization (imbinarize)	x		
Distance transform (disttrans)	x		
Removal of seed points			x
Morphological opreation (imclose)	x		

All of the pre-processing steps where implemented from the library embedded in Matlab. The images was converted by using the function *rgb2gray* and histogram equalizing by *histeq*. For the Gaussian smoothing filter, *imgaussfilt*, one has to specify the kernels standard deviation which is the size of the window to be averaged. The binarization of the image is done by using "imbinarize". The function lets one specify a threshold value. Otsu's method are done with the embedded Matlab function *graythresh*.

3.7.2 Feature extraction

Table 2 shows the functions that has been used for feature extraction. For the embedded functions, the name is in parenthesis.

Table 2: Overview of functions that are used for feature extraction

Method	Embedded	External	Novel
Histogram features (mean, var, skewness, median)	x		x
Region features (regionprops)	x		
lbp		x	
Chi-squared		x	

The lbp histograms has been calculated as explained in chapter 2.6 using an external function made by University of Oulu [31]. It is implemented with a 8 pixel circular neighbourhood and a radius = 1. To calculate the Chi-squared distance as explained in equation 2.7.1, a function by B. Schuerte was utilized [32].

3.7.3 Clustering

The implementation of the clustering is done with the use of existing Matlab function *kmeans*, which by default uses the *kmeans++* algorithm. As explained in section 2.8.1 it iterates through the data set and assigns a class based on the distance to the found cluster center.

The labeling of the samples is done after the result of the clustering.

3.7.4 Classification

Table 3: Overview of functions that are used for classification. For the embedded functions, the name is in parenthesis.The function that are crossed in two places with x* signify that the function has been made with guidelines from external part, with the use of embedded functions

Method	Embedded	External	Novel
Clustering (kmeans)	x		
SMOTE	x		x*
SVM (fitcsvm)	x		

The classification is conducted with the use of support vector machine with the embedded function called *fitcsvm*. The function lets you specify multiple parameters to optimize the results [33].

The SMOTE algorithm is implemented as suggested by Dr. Kunert [34]. The algorithm gets the feature vector of the samples, and for each sample is finds its k-nearest points by using MATLABs embedded *knearest* function.

4 Experiments and results

This chapter presents the preformed experiments and the results achieved. Some of the experiments is the foundation of which the following experiments are done from. Thus, the setup and result are shown for each experiments, where the best result is used in the proceeding experiments.

4.1 Experiment 1: Seed point extraction

Since the distance transform are sensitive to variations in binary images, different pre-processing steps were tested and evaluated. A series of experiments were set up to determine the best method of pre-processing the images to extract the seed points, and locate the cells. To get a robust representation, all experiments were done to the whole set of training images, where the entire set of images are processed in the same way for each method.

The experiments has been done on gray-scale images in addition to the red-, green- and blue color channels. The parameters that are altered between the experiments are the σ in the Gaussian smoothing filter and the threshold value. As histogram equalization is crucial to find all the cells, it was included in all experiments. The method that provided the best result for the training images were further implemented for the test set.

The number of cells to be located for the training set has been counted to be 881. For each experiments it was counted the number of cells that were located correctly, denoted as true positive, and the number of cells that were not located, denoted as false negative. In addition the number of non-cell objects located and/or duplicates of the same cell were counted and denoted as false positive.

Grayscale

Firstly the experiments are preformed on the histogram equalized grayscale images. Figures 4.1 illustrates how a resulting binarized image, from which the distance transform is run on vary with different parameters from which the distance transform

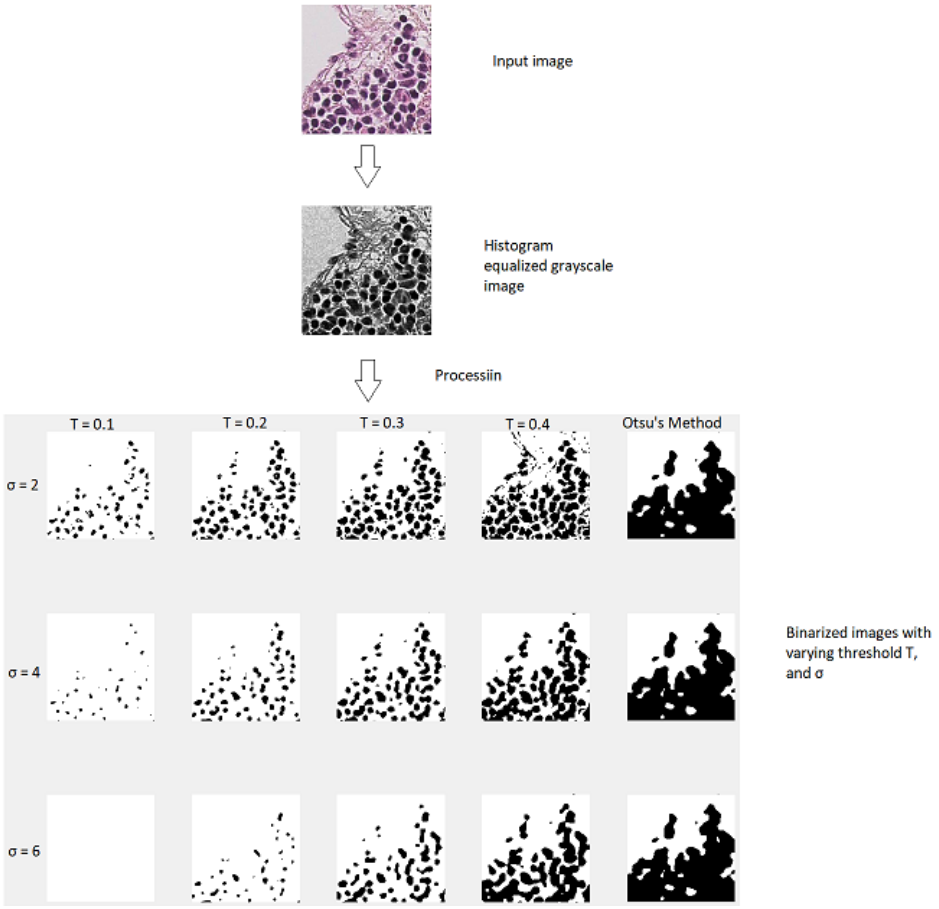
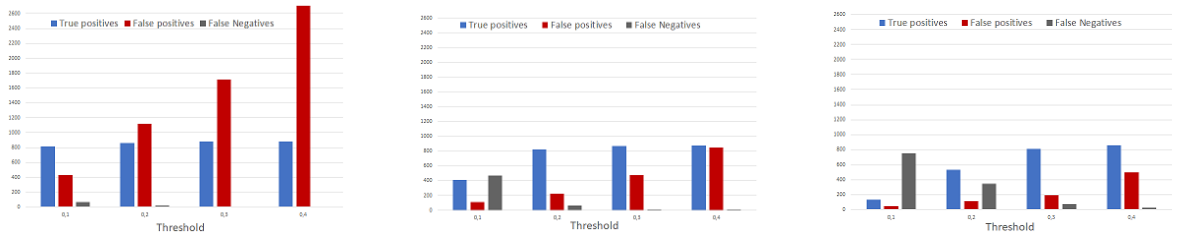


Figure 4.1: Processing of image before seed point extraction

Figure 4.2 shows the result of how many seed points that was found from the total set of training images with the use of grayscale images.

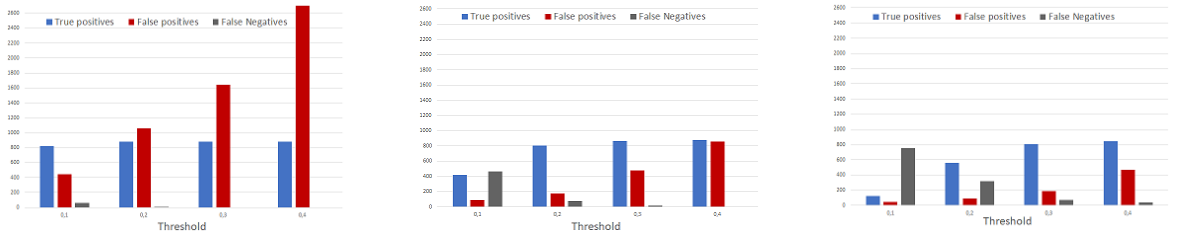


(a) $\sigma = 2$ (b) $\sigma = 4$ (c) $\sigma = 6$

Figure 4.2: Results experiment 1 with histogram equalized grayscale image with varying threshold. "σ" denotes the standard deviation of the Gaussian kernel

Further, the distance transform has been tested on the different color channels to see if the results would improve. The approach was the same as shown in figure 4.1, where the histogram equalized grayscale image were substituted with the input images' histogram equalized red-, green- and blue channel.

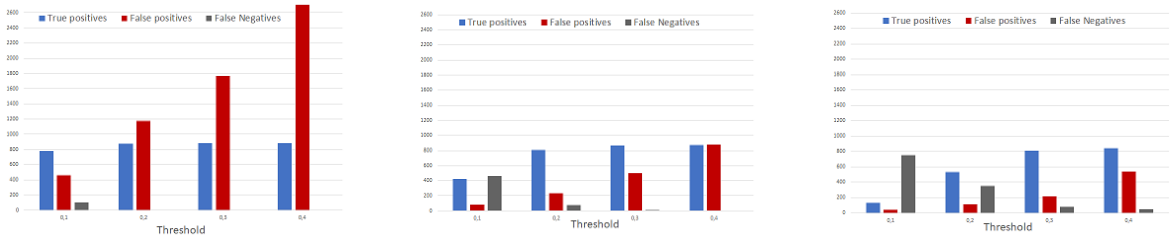
Red channel



(a) $\sigma = 2$ (b) $\sigma = 4$ (c) $\sigma = 6$

Figure 4.3: Results experiment 1 with histogram equalized red channel image with varying threshold. "σ" denotes the standard deviation of the Gaussian kernel

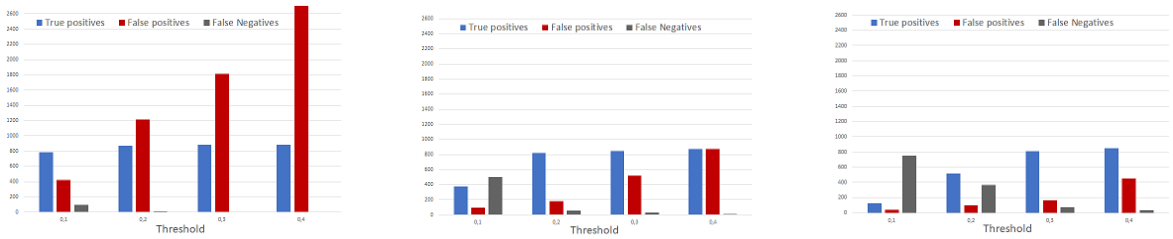
Green channel



(a) $\sigma = 2$ (b) $\sigma = 4$ (c) $\sigma = 6$

Figure 4.4: Results experiment 1 with histogram equalized green channel image with varying threshold. "σ" denotes the standard deviation of the Gaussian kernel

Blue channel



(a) $\sigma = 2$ (b) $\sigma = 4$ (c) $\sigma = 6$

Figure 4.5: Results experiment 1 with histogram equalized blue channel image with varying threshold. "σ" denotes the standard deviation of the Gaussian kernel

Pre-processing on the color channel images gives a similar results as on the grayscale image. With a low σ most of the seed points are found, but also the amount of false positives are overwhelming. We also see that with a high σ there is a substantial amount of false negatives as a result of too much blurring and the binarization step, and a higher threshold is needed.

The threshold value calculated by using Otsu's method for each image was higher than needed, as illustrated by figure 4.1. The amount of false positives drastically increased, as a result of background tissue that are not segmented away. Testing showed that a relatively low threshold value was needed to be left with only the cells.

It was evaluated that the best seed point location is achieved by using the histogram equalized gray-scale image with a fixed threshold of 0.2, and a Gaussian smoothing kernel of $\sigma = 4$. With this method, 1044 cells from the training images were found, where where 820 of the cells are located correctly, and 224 of the seed points are mostly duplicates within a same cell.

The same method was performed on the test images as for the training images, and these sets of cells are the ones proceeded with in the following experiments.

4.2 Experiment 2: Clustering

After experiment 1 it was decided to proceed with the grayscale images, where the Gaussian kernel factor σ was set to 4, combined with a fixed threshold of 0.2. After creating a data set that consist of 31x31 pixel images containing the cells as explained in chapter 3.3.4, the features are extracted as explained in chapter 3.4 and the next step was to predict if they are tumor infiltrating lymphocytes or not.

This experiment was set up for two main reasons:
1. To get an indication of the usability of the features.
2. To help with the labeling of the cells.

As there are no labeling done to the samples beforehand, the analysis of the clustering results were only done visually. For each clustering, each seed point were plotted and viewed with their respective assigned label, TIL or not-TIL. A set of tests were conducted with different feature vectors, with the parameters set to their default values from the embedded function.

A labeled set was derived by manually editing the misclassified samples from the clustering. The clustering indicated that the historgram features were suitable to separate the two classes.

4.3 Experiment 3: Classification with support vector machine

This experiments was conducted to evaluate the performance of the support vector machine(SVM) with different feature vectors. The SVM is first trained with the feature vectors derived from the cell samples in the training images, and tested with the feature vectors derived from the cell samples in the test images. As the SVM was tested with various combinations, this section is divided into subsections where the groups of feature vectors are tested separately. The performance of the classifier with the different combinations of feature vectors were done with the equations shown in equations 2.9.1-2.9.3.

4.3.1 Pre-experiment: Hyperparameters

For the SVM to work optimally a set of parameters needs to be determined. As mentioned in chapter 3.7.4 the *fitcsvm* is used, which finds the optimal parameters based on the features that are selected. For each length of the feature vectors, the best features are determined based on a criterion value, and their respective optimal parameters are found by minimizing the k-fold cross-validation loss.

In addition to the optimal parameters, the experiments are devised with default parameters, with only outlier fraction altered. Based on the clustering results showing an overlap between the two class types, it is assumed that an outlier rate needs to be specified.

4.3.2 Feature selection

The best features where chosen from the optimization function of the SVM, embedded in Matlab. The optimal features was found with different specified length, found both within each feature type and a combined set where histogram features, region features and LBP features are combined. Due to that SVMs is considered to work best with low- to moderate-dimensional data sets, the maximum length of the feature vectors are set to 4.

Histogram features:
There are 5 features in total that are derived from the histogram:

$$features_h = \begin{bmatrix} Quartile \\ DarkPixel \\ Mean \\ Variance \\ Skewness \end{bmatrix} \tag{4.3.1}$$

The optimal histogram features for each length were found by the SVM as shown in table 4.

Table 4: Results, selected optimal histogram features

Feature length	Calculated optimal featurevector
1	$[0\ 1\ 0\ 0\ 0]' .* features_h$
2	$[0\ 1\ 1\ 0\ 0]' .* features_h$
3	$[0\ 1\ 1\ 0\ 1]' .* features_h$
4	$[1\ 1\ 1\ 0\ 1]' .* features_h$

Region features:

$$features_r = \begin{bmatrix} Area \\ minorAxis \\ majorAxis \\ Differensial \end{bmatrix} \tag{4.3.2}$$

Table 5: Results, selected optimal region features

Feature length	Calculated optimal featurevector
1	$[1\ 0\ 0\ 0]' .* features_r$
2	$[1\ 1\ 0\ 0]' .* features_r$
3	$[1\ 1\ 1\ 0]' .* features_r$
4	$[1\ 1\ 1\ 1]' .* features_r$

LBP features:

The two features derived from LBP describes the similarity between the new texture, and the computed texture model for both epithelial cells, E, and TILs. In the experiment with LBP features, both are used.

$$features_{lbp} = \begin{bmatrix} \tilde{\chi}^2_{LBP,E} \\ \tilde{\chi}^2_{LBP,TIL} \end{bmatrix} \tag{4.3.3}$$

Combined features

A combination of all features derived was considered. The optimization function from the *fitcsvm* was used to find the best combinations from the combined set of features, $features_C$:

$$features_C = \begin{bmatrix} \tilde{\chi}^2_{LBP,E} \\ \tilde{\chi}^2_{LBP,TIL} \\ Quartile \\ DarkPixel \\ Mean \\ Variance \\ Skewness \\ Area \\ minorAxis \\ majorAxis \\ Differensial \end{bmatrix} \tag{4.3.4}$$

Table 6: Results, selected optimal combined feature vector

Feature length	Calculated optimal featurevector
1	$[0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]' .* features_C$
2	$[0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0]' .* features_C$
3	$[1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0]' .* features_C$
4	$[1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0]' .* features_C$

4.3.3 Experiment 3.1: Classification with histogram features:

This experiment was conducted to evaluate the performance of the SVM with histogram features. The experiment for the histogram features are divided into two parts:

1. With optimal histogram features as shown in table 7, with its corresponding parameters calculated with the optimization function as explained in 4.3.1.

2. With all combinations of features, with default parameters as explained in chapter.

The experiments has also been devised with and without over-sampling. Note that the synthesized minority samples created by SMOTE, though lying in the same area as the existing minority samples, will vary in location for each time it is run due to that it uses a random variable when placing new samples.

Optimal histogram features and parameters

Table 7 shows the results where the SVM has been trained with the optimal histogram features found in chapter 4.3.1, table 4. The best results from the classifier are highlighted in gray.

Table 7: Results from SVM, with optimal histogram features

Without SMOTE				
	Feature length	**Accuracy**	**TPR**	**TNR**
	1	0.9147	0.8112	0.9503
lightgray	2	0.9168	0.8233	0.9489
	3	0.9147	0.7912	0.9572
	4	0.9116	0.7631	0.9627
With SMOTE				
	1	0.9106	0.9157	0.9088
lightgray	2	0.9147	0.8835	0.9254
	3	0.9126	0.8675	0.9282
	4	0.9116	0.8514	0.9223

Combinations of histogram features and default parameters

Table 8 - table 11 shows the results where the SVM has been trained with various combinations of the histogram features. The best results from the classifier are highlighted in gray.

Feature notations:
Q = Quartile,
DP = Pixels below threshold,
M = Mean,
V = Variance,
S = Skewness.

Table 8: Results from SVM, trained with one histogram features

Without SMOTE				
	Features	**Accuracy**	**TPR**	**TNR**
	Q	0,8684	0,5783	0,9682
lightgray	DP	0,9126	0,8112	0,9475
	M	0,8828	0,6506	0,9627
	V	0,3936	0,7229	0,2804
	S	0,5159	0,5141	0,5166
With SMOTE				
	Q	0,3361	0,0321	0,4406
lightgray	DP	0,9116	0,9157	0,9102
	M	0,7842	0,9518	0,7265
	V	0,3998	0,7028	0,2956
	S	0,5128	0,5141	0,5124

Table 9: Results from SVM, trained with two histogram features

Without SMOTE				
	Features	**Accuracy**	**TPR**	**TNR**
	Q + DP	0,9157	0,8434	0,9406
	Q + M	0,8890	0,6747	0,9627
	Q + V	0,9013	0,6988	0,9710
	Q + S	0,8941	0,6586	0,9751
lightgray	M + DP	0,9198	0,8514	0,9434
	M + V	0,9106	0,7390	0,9646
	M + S	0,9013	0,7028	0,9696
	DP + V	0,9147	0,8353	0,9420
	DP + S	0,9178	0,8434	0,9434
	V + S	0,7441	0	1
With SMOTE				
	Q + DP	0,9116	0,9197	0,9088
	Q + M	0,8900	0,8635	0,8992
	Q + V	0,8993	0,8313	0,9227
	Q + S	0,9034	0,8273	0,9296
lightgray	M + DP	0,9156	0,9157	0,9116
	M + V	0,9123	0,8635	0,9309
	M + S	0,9115	0,8755	0,9378
	DP + V	0,9065	0,8966	0,9116
	DP + S	0,9147	0,8795	0,9268
	V + S	0,5036	0,4418	0,5249

Table 10: Results from SVM, trained with three histogram features

Without SMOTE				
	Features	**Accuracy**	**TPR**	**TNR**
	Q + DP + M	0,9168	0,8474	0,9406
	Q + DP + V	0,9188	0,8394	0,9461
lightgray	Q + DP + S	0,9188	0,8474	0,9434
	Q + M + V	0,9024	0,7068	0,9699
	Q + M + S	0,8993	0,6948	0,9696
	Q + V + S	0,9003	0,7028	0,9682
	DP + M + V	0,9147	0,8394	0,9406
	DP + M + S	0,9168	0,8394	0,9434
	DP + V + S	0,9188	0,8394	0,9461
	M + V + S	0,9137	0,7470	0,9710
With SMOTE				
	Q + DP + M	0,9137	0,9157	0,9130
	Q + DP + V	0,9160	0,8795	0,9213
lightgray	Q + DP + S	0,9188	0,8916	0,9282
	Q + M + V	0,9065	0,8434	0,9282
	Q + M + S	0,9188	0,8394	0,9461
	Q + V + S	0,9044	0,8394	0,9268
	DP + M + V	0,9085	0,8876	0,9157
	DP + M + S	0,9157	0,8795	0,9282
	DP + V + S	0,9065	0,8715	0,9185
	M + V + S	0,9188	0,8594	0,9392

Table 11: Results from SVM, trained with four histogram features

Without SMOTE				
	Features	**Accuracy**	**TPR**	**TNR**
	Q + DP + M + V	0,9168	0,8434	0,9420
	Q + DP + M + S	0,9168	0,8394	0,9434
lightgray	Q + DP + V + S	0,9198	0,8434	0,9461
	Q + M + V + S	0,9065	0,7189	0,9710
	DP + M + V + S	0,9188	0,8394	0,9461
With SMOTE				
	Q + DP + M + V	0,9075	0,8835	0,9157
	Q + DP + M + S	0,9065	0,8635	0,9213
lightgray	Q + DP + V + S	0,9126	0,8755	0,9254
	Q + M + V + S	0,9106	0,8474	0,9323
	DP + M + V + S	0,9085	0,8755	0,9199

The results show that most of the features vectors, derived from the histogram, results in a good accuracy. Also, the accuracy without SMOTE are somewhat better than with the use of SMOTE. However, the TPR are noticeably higher with the use of

SMOTE, and the TNR worse, meaning that more of the TILs are labeled correctly, but less of the epithelial-cells are labeled correctly.

4.3.4 Experiment 3.2: Classification with region features features:

This experiment was conducted to evaluate the performance of the SVM with region features. For this experiment it was tested only with the optimal combinations of features found in in table 5. Same as for the histogram features, both with and without SMOTE is tested.

Table 12: Results from SVM, with optimal region features

Without SMOTE				
	Feature length	**Accuracy**	**TPR**	**TNR**
	1	0,7440	0	1
	2	0,7440	0	1
	3	0,7440	0	1
	4	0,7440	0	1
With SMOTE				
	1	0,6752	0,5221	0,7279
	2	0,6670	0,5663	0,7017
	3	0,6578	0,6386	0,6644
	4	0,6670	0,6426	0,6754

4.3.5 Experiment 3.3: Classification with LBP features:

This experiment was conducted to evaluate the performance of the SVM with similarity features from the LBP texture description.

Table 13: Results from SVM, with LBP features

Without SMOTE				
	Features	**Accuracy**	**TPR**	**TNR**
	$\tilde{\chi}^2_{LBP,E} + \tilde{\chi}^2_{LBP,TIL}$	0,7390	0,1205	0,9517
With SMOTE				
	$\tilde{\chi}^2_{LBP,E} + \tilde{\chi}^2_{LBP,TIL}$	0,6608	0,6345	0,6699

4.3.6 Experiment 3.4: Classification with combined features:

This experiment was devised to evaluate the performance of the SVM where the optimal features are selected as in table 14. The optimal parameters of the SVM is computed by the embedded function in the fitcsvm.

Table 14: Results from SVM, with optimal combined features

Feature length	**Accuracy**	**TPR**	**TNR**
1	0,9147	0,8112	0,9503
2	0,9147	0,8153	0,9489
3	0,9178	0,8153	0,9530
4	0,9178	0,8193	0,9517

4.4 Experiment 4: Calculating TILs:

This experiment was conducted to test the quantitative properties of the proposed system. The best accuracy in experiment set 4.3 was achieved with the features, mean and DarkPixels. By using these features, it has been tested how many TILs is located relative to actual amount of TILs present from image to image present to see if differences occur.

Table 15: Results, Number of TILs in the images

Image	**Counted**	**Predicted w/o SMOTE**	**Predicted w/ SMOTE**
1	1	0	2
2	5	0	0
3	0	0	0
4	0	0	0
5	8	9	13
6	0	3	5
7	16	28	30
8	29	39	41
9	26	65	66
10	10	14	25
11	0	0	0
12	14	22	24
13	19	31	33
14	0	0	1
15	9	7	10
16	3	3	4
17	2	2	2
18	13	16	21
19	10	2	2
20	6	4	4
21	5	4	4
22	5	4	4
23	0	0	0
24	0	0	0
Total:	181	253	291

The resulting images are shown in appendix D, and an evaluation is done by pathologists at Stavanger Universitetssykehus, shown in appendix E.

5 Discussion

The best result for locating seed points was decided with Gaussian smoothing kernel with $\sigma = 4$ and a fixed threshold of 0.2, from the histogram equalized grayscale image. Over 93% of the cells is found by in experiment 1, although additional false positives are located as a side-effect.

With the use of SVM, a 91.98% accuracy classification of the located cells was achieved with histogram features while texture- and region features resulted in a lower performance of the system.

This chapter discuss the material, approach and results for this thesis.

5.1 Material and data set

The images used for this thesis was selected so that both TILs and epithelial cells were represented. It was also a focus point to have varying degree of color and contrast across the images to get a representative data set. As there were no labeling done beforehand, the images were cropped down to 300-by-300 pixel images to more easily create a labeled data set of cells.

The size should not affect the results as the methods used in the proposed system will work in the same way for larger images as for smaller images. As the method aim to locate the cells, and features are extracted from these, classification results should be scaleable. The only important thing in selecting the images is that they must be detailed enough so that the cells are separable from the tissue.

5.2 Seed point extraction

A distance transform was considered as a fast and simple method for finding seed points as cell centers. Due to the variations between each cell, where some had transparent areas in comparison with the background tissue, some of the located seed points were not placed at the center of the cells. It was considered to use morpholigical opening an closing to cope with this, but preliminary testing showed that this was at the expense of locating other cells. The Gaussian smoothing filter was considered to achieve sufficient results in relation to even out most of these cases.

Also, it was found that, as a result of the shape variances of the cells, the distance transform yielded multiple local maximums within the same cell which was stored as separate seed points. It can be thought that this affects the classification accuracy, as the features that are derived for the same cell will be similar, and one misclassified cell will be counted as multiple misclassified cells. On the other hand, the same can be said for the correctly classified cells, which can be considered of as a form for over-sampling.

A removal function was implemented to cope with the closest grouped seed points. However, a significant amount of such grouped seed points remained as they were outside of the removal window. To deal with these cases, a larger window was considered but due to overlapping cells this could result in loss of correct placed seed points. In addition, as the function iterates from the top left seed point to the bottom right, the most suitable seed point can be removed instead of the less suitable.

5.3 Feature extraction

In the early stages it was considered that features describing the darkness in a grayscale image should yield promising results. To capture the cells, a fixed window was introduced around all seed points to capture the cell. It was considered that the size of these windows could be larger, but as the overall variation of size between the cells where from 15-30 pixels in any axis, it was decided to proceed with a 31x31 pixel window.

Due to variations in darkness between the grayscale images, some overlap in feature space occurs, as an epithelial cell in one image can be similar to a tumor infiltrating lymphocyte in another. The histogram features derived is all a measure of the darkness of the cells, but it was evaluated that some bonus features could help with the separation of the classes.

Features describing the size of the cells was proven to yield insufficient results. The size of the cells vary in size and shape, such that no clear distinction could be made.

LBP was used to create histograms of each window containing a cell, and chi-squared distance was used to calculate the difference between these windows and computed models. By using a circular neighbourhood of radius = 1, it was found that the computed models were quite similar to each other. A larger radius could be used to capture larger texture structures but this not tested. As the windows created is of a set size of 31x31 pixels it was considered that a limited radius and neighbourhood pixels was sufficient.

Also as a result of that some seed points are located close to the edge of a cell, the fixed window around it can capture other cells as well, which in turn yields misleading features.

5.4 Results

Various combinations of feature vectors has been tested, and it is shown that histogram features overall yielded the best result. The predicted labeling from experiment 4 has been overviewed by pathologists at Stavanger Universitetssykehus, and the feed-back was positive, and their evaluation can be seen in appendix E.

5.4.1 Location of the cells

Since there are created additional seed points in the form of duplicates or other non-cell objects, they still have to be labeled. The same cell can be extracted features from multiple times, as a result of the windows are placed around the seed points. There are windows, containing the same cell, where the it is shifted slightly in either direction.

5.4.2 Classification

Compare the system to the manual process. The labeling of the data set where not done by someone with medical background. The manual labeling of the cells are done as good as possible by analyzing the cells based on information given by pathologists from Stavanger University Hospital. With this in mind, the results are .
The labeling were done semi-automatic by clustering the located seed points from the distance transform. As the classification is the done with the basis of the seed points located, some problems occurred.

6 Conclusion

This thesis has proposed a system to locate cells in a histological image and to decide if a located cell is a tumor infiltrating lymphocyte or not.

The use of distance transform to locate cell centers as seed points detected 93% of all the cells in the data set, but it is found that this can vary from image to image. The down-side of the method is that there are located a significant amount of false positives as a result of the distance transform.

A Support Vector Machine is proven to be a suitable classifier for this thesis as it is narrowed down to a two-class problem. Overall the histogram features showed a good accuracy, with a 92% accuracy of the classification of the located cells is achieved from the SVM with two histogram features.

The performance measures of the results revealed that 85% of the TILs, and 94% of the epithelial cells are classified correctly. By implementing an over-sampler to create synthesized samples of TILs, the TPR is marginally improved. Providing a 2% increase in correctly classified TILs, but at the cost of the overall accuracy, the performance were better without oversampling.

Due to the duplicate seed points that are found with the distance transform the proposed system did not prove to be quantitative. Though the proposed system achieves a good accuracy in the classification, some cells are counted twice as a result of the fixed window around the seed points. It is concluded that histogram features are suitable for separating TILs and non-TILs, but the distance transform does not provide sufficient location of the cells.

6.1 Future work

The data set this thesis used, can to be expanded to make the data set even more representative. The method should be tested properly on larger images. This would requires a new labeling process.

The proposed system has its flaws when it comes to locating the cells and improvements on this section is advised to create a quantitative measure. The features that are used results in an overall good classification accuracy, but it is dependant on the seed point locations. As it stands, the system it is sensitive of disturbances and dark non-cell objects in the histological image, resulting in fake positives detected, in addition to multiple seed points for one cell. Advancements are continuously done on the detection of cell nucleons. Yousef Al-Kofahi et al. has proposed a method by using graph-cuts-based binarization and multiscale multiscale Laplacian-of-Gaussian filtering which has proved good results[35]. Also, an alternative can be to use fast radial symmetry [36]. As an alternative, to comprehend with the fake positives, adding a class describing non-cell objects could be considered. This however, will not deal with the duplicates of the same cell.

References

[1] L. A. Torre, F. Bray, R. L. Siegel, J. Ferlay, J. Lortet-Tieulent, and A. Jemal, "Global cancer statistics, 2012," *CA: a cancer journal for clinicians*, vol. 65, no. 2, pp. 87–108, 2015.

[2] Key statistics for bladder cancer. [Online]. Available: https://www.cancer.org/cancer/bladder-cancer/about/key-statistics.html

[3] O. M. Mangrud, "Identification of patients with high and low risk of progresson of urothelial carcinoma of the urinary bladder stage ta and t1," 2014.

[4] M. Mossanen and J. L. Gore, "The burden of bladder cancer care: direct and indirect costs," *Current opinion in urology*, vol. 24, no. 5, pp. 487–491, 2014.

[5] P. L. Ho, S. B. Williams, and A. M. Kamat, "Immune therapies in non-muscle invasive bladder cancer," *Current treatment options in oncology*, vol. 16, no. 2, p. 5, 2015.

[6] P. Sharma, Y. Shen, S. Wen, S. Yamada, A. A. Jungbluth, S. Gnjatic, D. F. Bajorin, V. E. Reuter, H. Herr, L. J. Old *et al.*, "Cd8 tumor-infiltrating lymphocytes are predictive of survival in muscle-invasive urothelial carcinoma," *Proceedings of the National Academy of Sciences*, vol. 104, no. 10, pp. 3967–3972, 2007.

[7] L. Microsystems. The scanning system for superior whole slide digital images. [Online]. Available: https://www.leica-microsystems.com/products/light-microscopes/clinical-microscopes/details/product/leica-scn400/

[8] L. Cheng and D. Bostwick, *Urologic Surgical Pathology.* Mosby Elsevier, 2008.

[9] J. Ferlay, I. Soerjomataram, R. Dikshit, S. Eser, C. Mathers, M. Rebelo, D. M. Parkin, D. Forman, and F. Bray, "Cancer incidence and mortality worldwide: sources, methods and major patterns in globocan 2012," *International journal of cancer*, vol. 136, no. 5, 2015.

[10] WHO. (2018) Cancer Today. [Online]. Available: http://gco.iarc.fr/today/online-analysis-map?mode=population&mode_population=who&population=900&sex=0&cancer=22&type=0&statistic=0&prevalence=0&color_palette=default&projection=natural-earth

[11] E. Janssen, private communication, 2018.

[12] Y.-T. Kim, "Contrast enhancement using brightness preserving bi-histogram equalization," *IEEE transactions on Consumer Electronics*, vol. 43, no. 1, pp. 1–8, 1997.

[13] M. Lindenbaum, M. Fischer, and A. Bruckstein, "On gabor's contribution to image enhancement," *Pattern Recognition*, vol. 27, no. 1, pp. 1–8, 1994.

[14] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[15] A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing," *Journal of the ACM (JACM)*, vol. 13, no. 4, pp. 471–494, 1966.

[16] Wikipedia. Distance transform. [Online]. Available: https://en.wikipedia.org/wiki/Distance_transform

[17] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[18] K. Pearson, "X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 50, no. 302, pp. 157–175, 1900.

[19] B. Clarke and D. Sun, "Reference priors under the chi-squared distance," *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 215–231, 1997.

[20] A. Trevino. (2016) Introdiction to k-means clustering. [Online]. Available: https://www.datascience.com/blog/k-means-clustering

[21] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[22] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding."

[23] Mathworks. (2018) Kmeans. [Online]. Available: https://se.mathworks.com/help/stats/kmeans.html#bues5gz

[24] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[25] Wikipedia. Support vector machine. [Online]. Available: https://en.wikipedia.org/wiki/Support_vector_machine

[26] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[27] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2009.

[28] C. Drummond, R. C. Holte *et al.*, "C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in *Workshop on learning from imbalanced datasets II*, vol. 11. Citeseer, 2003, pp. 1–8.

[29] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[30] MathWorks. Variance. [Online]. Available: https://se.mathworks.com/help/matlab/ref/var.html#bundkwe-1

[31] U. of Oulu. "A general Local Binary Pattern (LBP) implementation for Matlab". Lbp.m ver 0.4, getmapping.m ver 0.2. [Online]. Available: http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab

[32] B. Schauerte. (2013) Histogram distances. [Online]. Available: https://se.mathworks.com/matlabcentral/fileexchange/39275-histogram-distances

[33] Mathworks. Support vector machines for binary classification. [Online]. Available: https://se.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html

[34] R. Kunert. (2017) Smote explained for noobs - synthetic minority over-sampling technique line by line. [Online]. Available: http://rikunert.com/SMOTE_explained

[35] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam, "Improved automatic detection and segmentation of cell nuclei in histopathology images," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 4, pp. 841–852, 2010.

[36] S. Zafari *et al.*, "Segmentation of overlapping convex objects," 2014.

A - Matlab

The following files is embedded in matlab.7z.

colorchannels.m
used to extract the red,green and blue color channel from RGB input image.
seedPointExtraction.m
Script to locate the seed points.

removeSeedPoints.m:
Removes seed points that are close together

removeSeedPointsedge.m:
Removes the seed points at close to edge

clustering.m
The script developed for k-means clustering of the cells.

getMapping.m
Returns a structure containing a mapping table for LBP codes. source: University of Oulu.

lbp.m
returns the local binary pattern image or LBP histogram of an image. source: University of Oulu.

LBPHistograms.m
Script to make LBP feature vector derived from lbp.m

traininglbp.csv
csv file of features derived from LBP histogram of cells in training images, found in LBPHistograms

testlbp.csv
csv file of features derived from LBP histogram of cells in test images, found by lbp.m

ModelLBPChi.m
Script to make LBP histogram models and compute Chi-squared distance

lbp.m
returns the local binary pattern image or LBP histogram of an image. source: University of Oulu.

SMOTEegen
Function for creating synthesized samples of minority class

SVMclassifier.m
Script for training SVM with fixed parameters to predict cell class. Change feature vector each time.

SVMclassifierHyperParameters.m
Script for training SVM with optimal parameters to predict cell class. Change feature vector each time.

combinedfeaturestrening.csv
csv file of features derived from cells found in the training images

combinedfeaturestrening.csv
csv file of features derived from cells found in the test images.

B - Training images

Images used for training are embedded in "trainingimages.7z"

C - Test images

Images used for training are embedded in "testimages.7z"

D - Results experiment 4

The result of own labeling, predicted labeling w/o SMOTE and predicted labeling w SMOTE with mean and DarkPixels used as features, is embedded in "resultexp4.7z"

E – Evaluation of experiment 4, by pathologists at SUS

Image number	Venstre %	Høyre %
1	100	95
2	100	85
3	100	100
4	100	100
5	50	60
6	Veldig vanskelig uten kontekst	Veldig vanskelig uten kontekst
7	95	80
8	90	85
9	50	20
10	95	60
11	100	100
12	95	85
13	95	85
14	100	95
15	95	95
16	100	100
17	100	100
18	90	90
19	85	75
20	98	85
21	85	88
22	100	100
23	100	100
24	100	100