Image Based Characterization of Circulating Tumor Cells

Katrine Brandt Albrektsen

Kongens Lyngby 2014

DTU Compute
Technical University of Denmark
Matematiktorvet, building 303B,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3351
compute@compute.dtu.dk
www.compute.dtu.dk

CytoTrack Aps
Gl. Lundtoftevej 1D st
2800 Kongens Lyngby
Denmark
Phone +45 7027 4200
info@cytotrack.com
www.cytotrack.dk

Abstract

The assessment of circulating tumor cells (CTCs) in blood samples from cancer patients can help in determining the prognosis for the patient and can help in personalized treatment. The CytoTrack is a fluorescent microscope, which can be used to image possible CTCs within a blood sample. These images are manually looked through by a trained operator, in order to determine which images are of CTCs and which are false positives. This is time consuming and tedious work, and reducing this scoring time is the topic of this thesis.

In this thesis images from the CytoTrack are automatically scored. For this different classification methods are tested including random forest and support vector machines. The algorithms are tested on data from breast cancer patients and on data from spiked samples. The performance on the spiked data are significantly better compared with the patient data. This is explained by bigger variations within the patient samples compared with the spiked samples.

Through cross validation both high sensitivities and specificities are computed. For this work the sensitivity is weighted over the specificity since it is important not to miss any true positives. To avoid missing true positives, thresholds are determined based on the receiver operating characteristic (ROC) curves. These thresholds are chosen so the true positive rate is equal to one.

After choosing a threshold the algorithms are tested on a unknown test set. From this testing it is shown that it is possible to completely avoid false negatives and still classify a significant part of the data as negatives. That is, the amount of data to be scored manually is reduced and hence the scoring time is reduced.

Resumé (Danish)

Cirkulerende tumorceller (CTC'er) i blodprøver fra kræftpatienter kan detekteres og dette kan hjælpe til med at bestemme prognosen for patienten og kan hjælpe til med at tilrettelægge individuel behandling. CytoTrack er et fluorescerende mikroskop, som kan anvendes til at tage billeder af mulige CTC'er i en blodprøve. Disse billeder kigges manuelt igennem af en uddannet operatør, for at afgøre hvilke billeder der er af CTC'er og hvilke der er falske positive. Det er tidskrævende og kedeligt arbejde, og at reducere denne scoringstid er emnet for denne afhandling.

I denne afhandling bliver billeder fra CytoTrack automatisk klassificeret. Til dette er forskellige metoder testet, inklusiv random forrest og support vector machines. Algoritmerne er afprøvet på data fra brystkræftpatienter og på data fra spikede prøver. Baseret på denne test er klassificeringen af spiket data signifikant bedre end klassificeringen af patient data. Dette kan forklares ved store variationer inden for patientprøverne, sammenlignet med de spikede prøver.

Gennem krydsvalidering blev der beregnet både høje sensitiviteter og specificiteter. Da det er vigtigt ikke at fejlscore nogle af de sande positive, er der i dette arbejde lagt mest vægt på sensitivitet frem for specificitet. For at undgå at misse nogle af de sande positive, fastlægges en threshold på baggrund af receiver operating characteristic (ROC) kurver. Threshold værdierne er valgt således at den sande positive rate er lig med én.

Efter at have valgt en threshold, testes algoritmerne på et ukendt test sæt. Fra denne test er det vist, at det er muligt helt at undgå falsk negative og stadig klassificere en betydelig del af data som negativ. Det vil sige at mængden af data der skal scores manuelt reduceres og dermed reduceres scoringstiden.

Preface

This thesis was conducted as a collaboration between the Department of Mathematics and Computer Science (DTU compute) at the Technical University of Denmark (DTU) and CytoTrack Aps in the period from January 2014 to June 2014. The thesis was conducted as a partial fulfillment of the requirements for acquiring a Master of Science Degree in Engineering M.Sc.Eng and the work amounts to 32.5 ECTS points.

The thesis was supervised by Professor Rasmus Larsen(DTU), Professor Knut Conradsen (DTU), Postdoc Mark Lyksborg (DTU), and Tom Hede Markussen PhD (CEO of CytoTrack Aps).

The undersigned is a Master student in Medicine and Technology at the Technical University of Denmark (DTU) and the Faculty of Health and Medical Sciences at the University of Copenhagen.

Lyngby, 20-June-2014

Katrine Brandt Albrektsen

Acknowledgements

I would like to express my deep gratitude to my DTU supervisors Professor Rasmus Larsen, Professor Knut Conradsen and Postdoc Mark Lyksborg for their inputs during the Friday meetings.

A special thanks to the whole CytoTrack team, especially Tom Hede Markussen PhD CEO, Anders Frandsen R&D Engineer and Henrik Stender PhD CCO, for guidance and suggestions during the project.

I would like to thank the Department of Clinical Biochemistry at Hillerød Hospital, especially Thore Hillig PhD, for providing data for the thesis.

I would also like to thank Peer Horn Cand. Scient (PhD student), for helping me find some excellent literature.

Contents

Abbreviations

In this section the abbreviations used in this project are given.

Abbreviation	Explanation
BLOB	Binary Large Object
CD45	Leukocyte Common Antigen (the CD stands for Cluster of Differentiation)
CTC	Circulating Tumor Cells
DAPI	4',6-Diamidino-2-Phenylindole
EpCAM	Epithelial Cell Adhesion Molecule
FITC	Fluorescein Isothiocyanate (Anti-pan-Cytokeratin)
fn	False Negative
fp	False Positive
MCF7	Michigan Cancer Foundation-7 (Name of a cell line)
oob	Out-Of-Bag

rbf	Radial Basis Function
ROC	Receiver Operating Characteristic
ROI	Region of Interest
SkBr3	(Memorial) Sloan–Kettering Cancer Center-3 (Name of a cell line)
SVM	Support Vector Machine
tn	True Negative
tp	True Positive

Motivation and Outline

Motivation

Cancer is one of the most common causes of death world wide with around 7.6 million cancer deaths each year [JBC+11]. The majority of the cancer related deaths are attributed to blood-borne metastatic cancers, since this gives limited treatment options and thus an unfavorable prognosis [H+13][Lig12]. The cells responsible for the metastases move through the blood vessels and are known as *circulating tumor cells* (CTCs).

The presence of CTCs within a blood sample can be used as a prognostic factor and give an indication of whether the cancer is actively spreading or not. By monitoring the number of CTCs during a course of treatment, the effectiveness of the treatment can be assessed much quicker compared with classical imaging modalities such as PET, CT and MRI. Furthermore CTCs can be characterized for different cancer mutations and thus be used for personalized treatment [C+04][Lig12].

Different techniques are available for detecting CTCs and they are generally based on the same principles. First an isolation of the CTCs based on differences in the characteristics of blood cells and CTCs, next the cells are marked with e.g. fluorescent markers to make it possible to distinguish them from the rest of the cells. The cells are then imaged and the images are manually assessed and classified by a trained operator. This manual scoring can take up to several hours and is time consuming and tedious work. The scoring is made qualitatively and both inter- and intra- operator variability is introduced. Automatic scoring of CTCs based on images is the topic of this thesis in the hope of overcoming

some of these difficulties.

Outline

This section contains an overview over the chapters included in this thesis

- **Chapter 1 - Objectives:** In this chapter a short description of the objectives of this thesis is given
- **Chapter 2 - Introduction:** In this chapter an introduction to the problem at hand is given. This includes a description of cancer, especially metastasizing cancer, diagnosis and treatment of cancer and description of CTCs and how they are detected.
- **Chapter 3 - Existing Technology:** In chapter 3 some earlier attempts of classifying CTCs and cells in general are shortly discussed. Furthermore a short description of the basic principles behind the competing technology CellSearch® is included.
- **Chapter 4 - Data:** In this chapter a description of the data used is given.
- **Chapter 5 - Methods:** The methods used for this thesis are introduced in chapter 5. A general description of the methods at hand are given and some preprocessing methods are tested.
- **Chapter 6 - Results:** This chapter contains the results of using the different scoring algorithms. The different algorithms are tested and their performance are compared.
- **Chapter 7 - Discussion:** The final results are discussed.
- **Chapter 8 - Conclusion:** A short conclusion of the entire thesis is given in this section.
- **Appendix A - CytoTrack Images:** Some of the images used for this thesis are shown in this appendix
- **Appendix B - Feature Histograms:** The histograms of all the features introduced in chapter 5 are shown in this appendix.
- **Appendix C - Specifications:** The specifications made for the program is given in this appendix

Chapter 1

Objectives

The objective of this thesis is to develop an automatic algorithm for scoring of cell images obtained using the CytoTrack.

The CytoTrack generates a catalog of images of potential CTCs. These images contains a lot of false positive images and only a few true positives, since CTCs are extremely rare. The current method is for trained operators to manually look through these images to determine which are true positives and which are false positives. The goal for this automatic scoring is to reduce the number of false positives without losing any of the true positives. Thus a high sensitivity is weighted over a high specificity. The manual scoring can take up to several hours per sample and by reducing the number of images to look through, the scoring time can be reduced. The goal is to at least score 10% of the true negatives as negatives.

Chapter 2

Introduction

2.1 Cancer

Cancer is one of the most common causes of death world wide with around 7.6 million cancer deaths each year [JBC+11]. The rate of cancer incidences are increasing in the economically developing countries and the cancer burden is thus getting bigger each year. In the future cancer should be a chronic disease that you live with and not a disease that you die from, and in order to fulfill this goal more research is necessary.

Most of the cancer related deaths are attributed to blood-borne metastatic cancers, since the metastases often attacks vital organs such as the lungs or the brain [H+13][Y+11]. Furthermore when the cancer has started metastasizing the treatment options are limited. The cells that move through the blood and start the secondary tumors are known as *circulating tumor cells* (CTCs) and assessment of these can be a step in the right direction for cancer research.

The existence of CTCs has been known since the 19th century and these have been linked to metastasizing [Lig12]. In the circulation most of the CTCs get destroyed, but some cells survive, escape the circulation and starts proliferation at a new site, see figure 2.1.

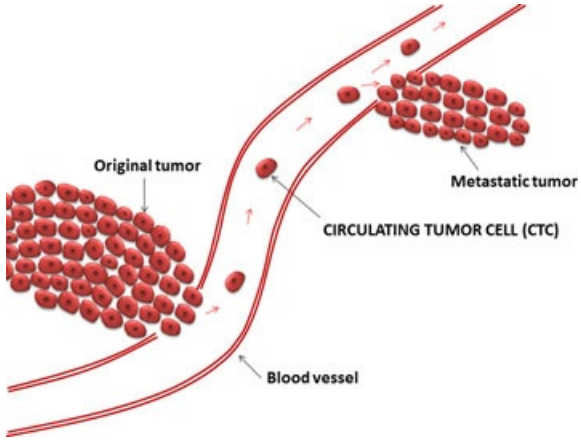


Figure 2.1: Illustration of how CTCs leave the primary tumor and starts metastasizing [H+13].

2.2 Cancer - Diagnosis and Treatment

Cancer is generally diagnosed using different imaging modalities such as PET, CT and MRI and confirmed with a biopsy. After detection of malignancies a risk analysis is performed to determine the severity of the disease. This risk analysis is based on tumor size, whether or not there are metastases, and whether or not the cancer has spread to lymph nodes [NH04]. The assessment of CTCs could possibly be a supplement to this risk analysis.

For the treatment the first step is generally surgical removal of the tumor, followed by chemotherapy, radiation therapy, immunotherapy and/or targeted therapy [Lig12]. The effectiveness of the therapy is detected through the different imaging modalities where typically a scan is made before start of therapy and again 3-6 month after ending therapy. There is thus a long waiting before it is possible to tell whether the therapy is working. Furthermore it is only possible to detect tumors of a certain size using classical imaging modalities and thus very small tumors can easily be overlooked. By measuring the number of CTCs a real-time monitoring of the treatment is possible. For an effective treatment the number of CTCs found in a blood sample will decrease, but if the number stays the same or even increase it is possible to conclude that the treatment is ineffective [C+04]. With this knowledge is it possible to change the treatment much faster and thereby give the patient a much better prognosis.

2.3 Ciculating Tumor Cells

The knowledge of CTCs is still very limited and looking at these can give rise to a lot of new knowledge. CTCs are very low in number with as few as one among billions of other cells [Y+11]. Generally the number of CTCs is higher for patients with metastatic cancers than for local cancers, and the number of CTCs will generally decrease during an effective treatment [H+13].

The CytoTrack system can be used to detect CTCs. Different fluorescent markers are used in the detection, and for finding CTCs the most widely used marker is for cytokeratin. By using other markers it is possible to characterize the individual CTCs. This information can be used for personalized therapy and can help to identify new therapeutic targets to help suppress metastasis [Y+11].

The current definition of a CTC is based on preclinical studies made using the CellSearch® system[1] [Lig12]. For a cell to be classified as a CTC it has to be positive for cytokeratin, it has to have a nucleus, it should not be positive for CD45[2], it should have a cell-like morphology and it should have a diameter larger than 4 μm [H+13][Lig12].

2.4 Sample Preparation

The first step in CTC detection is collection of the blood sample. To avoid skin cells contaminating the blood, the first draw should not be used. Before a blood sample is scanned in the CytoTrack, some preparation steps are necessary. The first step is to isolate the buffy coat[3] from the blood sample, see figure 2.2. The blood sample is centrifuged and hereby it is possible to remove the plasma in order to get to the buffy coat. The buffy coat is transferred to a new tube and to make sure all erythrocytes are removed, the sample is lysed[4] [Cyt14a]. This process is very crucial and it is possible that some cells are lost in this step.

The next step is to stain the cells and for this three different types of fluorescent markers are used. A fluorescent marker is a molecule that can be attached to a specific biomolecule and thereby help in the detection of that biomolecule.

[1] The CellSearch® system is another system which can be used for CTC detection, see more in chapter 3

[2] CD45 is positive for some leukocytes

[3] The part of the blood sample that primarily contains the leukocytes and platelets after centrifuging the blood sample

[4] Lysis means breaking down erythrocytes

To visualize a labeled molecule the fluorescent marker is excited by a light source and light of lower frequency, and thus higher wavelength, is emitted [WW07][YFF08]. When more than one marker is used it is important that these emit light of different wavelengths and thus are distinguishable.

For this thesis the first fluorescent marker used is Anti-pan-Cytokeratin (*Fluorescein Isothiocyanate* (FITC)), which stains cells containing cytokeratin and emits light in the green spectrum [Cyt14a]. Cytokeratin is expressed in cells of epithelial origin and thus epithelial CTCs can be visualized using FITC. The second staining is a nuclear stain (*4',6-Diamidino-2-Phenylindole* (DAPI)), that emits light in the blue spectrum. DAPI stains all cells having a nucleus, meaning all the remaining cells in the tube should be stained with DAPI. The last staining is Anti-human CD45 (*leukocyte common antigen* (CD45)), which is expressed in the cell membrane of some leukocytes, i.e. negative for CTCs [Cyt14a]. CD45 emits light in the red spectrum. Besides these three staining the phenomenon of auto fluorescence can appear in a sample. That is, other molecules than the stained ones can emit light and thus give rise to false positives.

For a cell to be classified as a CTC it should be positive for both FITC and DAPI, but negative for CD45, in order to be categorized as a CTC.

After the staining the cells are washed and smeared out on a CytoTrack disc where the cells are immobilized.

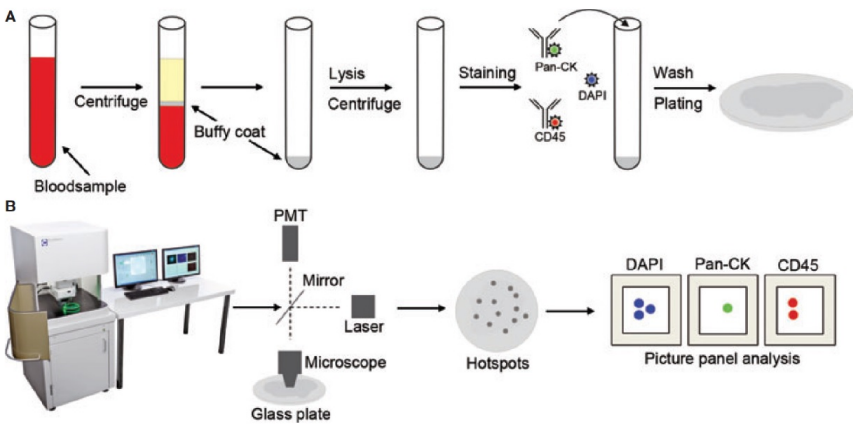


Figure 2.2: Illustration of the sample preparation and scanning procedure [H+13].

2.5 CTC Detection an Enumeration

The CytoTrack is a system for detection of rare cells, such as CTCs and an illustration of the CytoTrack can be seen in figure 2.3.

Figure 2.3: Illustration of the CytoTrack from [Cyt14b]

After the disc has been prepared it is inserted in the CytoTrack. Here a laser is used to scan the sample, in order to find FITC positive areas. These are saved as hot spots, i.e. possible CTCs. After the scanning the CytoTrack retrieves the hot spots and these are imaged using the camera.

For the setup used for this project, i.e. CTC detection, three images are taken of each hot spot. The three images are obtained using three different filters in order to image the three different types of staining. That is, CD45 in the red channel, FITC in the green channel, and DAPI in the blue channel, see figure 2.4.

After the scanning the images are manually scored by a trained operator, i.e. a bunch of the hot spots are false positives and only a few are true positives. The scoring is based on the CTC definition described in section 2.3. The operator is presented with three images of each hot spot, i.e. an image from the CD45-, FITC- and DAPI- channel, and from these it is determined if the criteria for a CTC is met.

The number of hot spots from one sample varies significantly from patient to

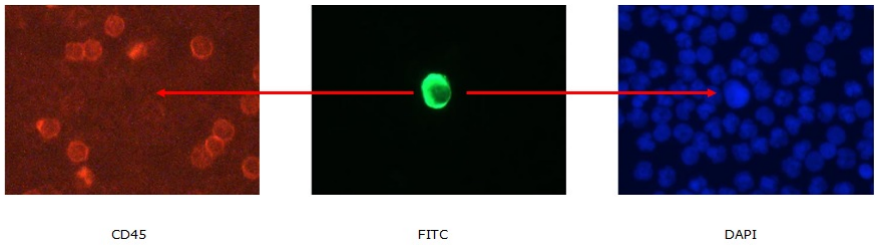


Figure 2.4: Example of the images obtained using CytoTrack

patient. The scoring can take up to several hours for large data sets, where there can be up to thousands of hot spots. As this scoring is a manual process both inter- and intra- operator variability is introduced. Furthermore this manual scoring is time consuming, tedious and costly, and it would be preferable to reduce or even eliminate this step. Reduction in scoring time or even elimination of manual scoring could possibly be obtained by use of an automatic scoring algorithm.

Chapter 3

Existing Technology

CTC detection is a novel technology and to my knowledge only one attempt of automatic classification of CTC images is publicly documented, i.e. [Lig12]. In [Lig12] the images are obtained from the CellSearch® system. CellSearch® is the only available FDA-cleared technology for CTC detection and hence a short description of the procedures involved in CTC detection when using the CellSearch® follows.

In the preparation of the sample an enrichment step based on magnetic separation is added. That is, the CTCs are captured by adding an EpCAM [1] ferrofluid and placing the sample between magnets. After this enrichment the cells are stained with fluorescent markers similar to those used with the CytoTrack. This results in a 300 μl sample which is placed in a semi-automatic epi-fluorescent microscope, where it is scanned in four fluorescent channels [Lig12]. These channels includes one for each of the three fluorescent markers and a control channel used to verify auto fluorescence or used for a fourth biomarker. After the scan the CellSearch® software finds objects that are positive in the cytokeratin channel (also known as the PE-channel) and in the nuclear channel (also known as the DAPI-channel). An example of the image gallery presented to the CellSearch ® operator is given in figure 3.1.

For the classification in [Lig12] the patient survival is used as a training parameter

[1] EpCAM = Epithelial Cell Adhesion Molecule, which is expressed in epithelial cells

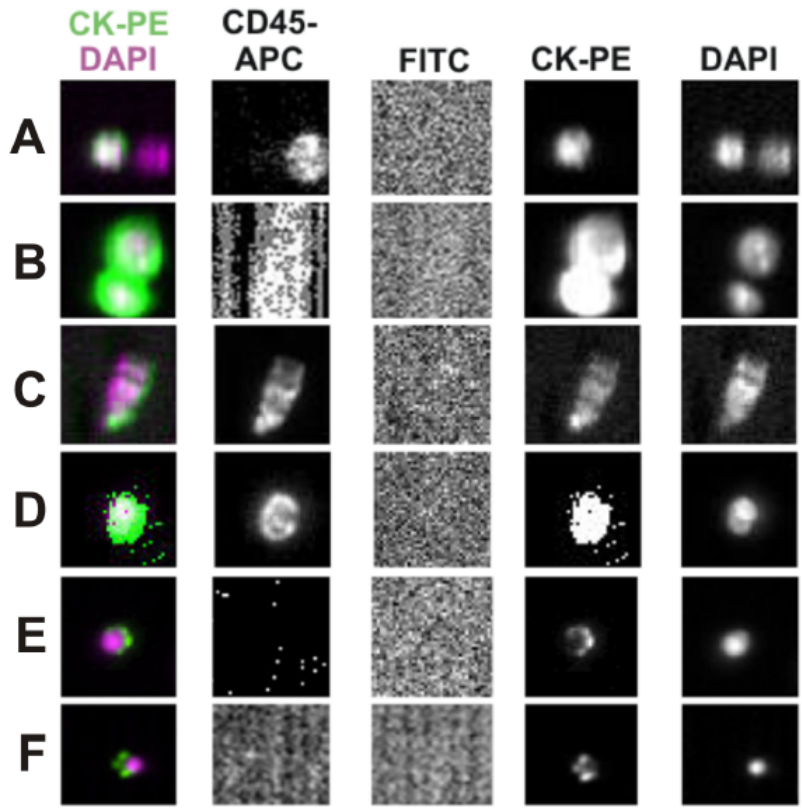


Figure 3.1: Examples of the CellSearch® image gallery [Lig12]. The first column (CK-PE DAPI) is overlap images of the cytokeratin and nuclear channel, the second column (CD45-APC) is the CD45 channel, the third column (FITC) is the control channel, the fourth column (CK-PE) is the cytokeratin channel and the fifth column (DAPI) is the nuclear channel.

instead of a marked training set and furthermore [Lig12] studies the impact of changing the CTC definition to see the effect on patient survival.

The process of scoring the images is divided into segmentation, features extraction and classification. In [Lig12] for the segmentation three histogram based methods are tested, these are Zack's triangle threshold, Otsu's method and the isodata method. From the segmented images 24 basic features are extracted and from these 4 are selected through univariate analysis. The classification is based on these 4 features, and different threshold levels for each feature are tested. This results in 16464 different classifiers, which are tested. The final results from [Lig12] are comparable to manual scoring, and the inter- and intra- operator variability is eliminated.

Besides [Lig12] the identification and characterization of different types of cells have been researched thoroughly, e.g. in [HEH+06] [PSF+04] [BACP07].
In [HEH+06] different methods for cell classification are reviewed. Among all methods tested the random forest and support vector machines algorithms outperform all other algorithms, with both the highest mean classification accuracy and the lowest standard deviation.

In this thesis different methods are tested for both the segmentation and for the classification. For the segmentation Otsu's method and a fixed threshold are tested, and for the classification support vector machines and random forest are tested.

Chapter 4

Data

The images are obtained as described in section 2.5. Three images are obtained of each hot spot, i.e. one for the CD45-, FITC- and DAPI staining. The images are of size 1280x960 pixels and each pixel corresponds to 0.375x0.375μm. The images are rgb images and are obtained in jpeg format. Two sets of data are used for this thesis. The first set consist of images obtained from cell lines mixed with humane blood, and the second set consist of images obtained from blood samples of mammary cancer patients. In this chapter cropped versions of the original images are shown, for the uncropped versions the reader if referred to appendix A.

4.1 Spiked Samples

The first data set consist of images obtained from cell lines mixed with humane blood. These images are generally easier to score since the cells have had the change to grow big and have not been subjected to the stress and strain from the human circulatory system.

Data set	No. of positives	No. of negatives
MCF7 1	129	51
MCF7 2	162	45
MCF7 3	105	57
MCF7 4	129	45
MCF7 5	123	123
MCF7 6	132	135
MCF7 7	174	57
MCF7 8	114	90
SKBR3 1	315	285
Total	1383	888

Table 4.1: Number of image set scored positive and negative by the operator in the spiked data.

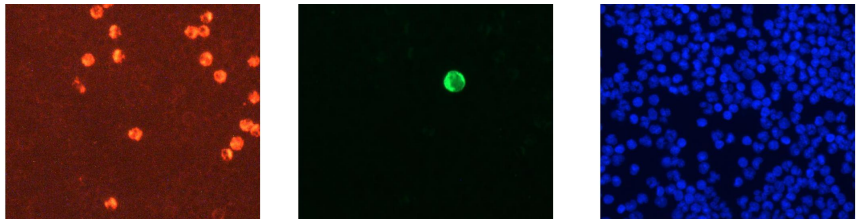
Two types of cell lines are used, MCF7[1] and SkBr3[2]. These are chosen since they have a high cytokeratin expression, i.e. a high homogeneous FITC signal (see figure 4.1b and 4.2b), and their nuclei are increased in size as seen in figure 4.1c and 4.2c. These characteristics makes them easy to score and thereby a good starting point.

Both the MCF7 and SkBr3 cell line originates from mammary cancer patients and are both isolated in 1970 from Caucasian women [ATC14b] [ATC14a]. Examples of how the cells look when imaged using CytoTrack can be seen in figure 4.1 and 4.2. Furthermore some examples of false positive images (i.e. images that are scored negative in the manual scoring) are shown in figure 4.3 and 4.4.

For training of the algorithm eight data sets obtained from MCF7 and one set from the SkBr3 cell line are used. The number of positives and negatives for each data set can be seen in table 4.1. For the testing 2 data sets from the SkBr3 cell line are used.

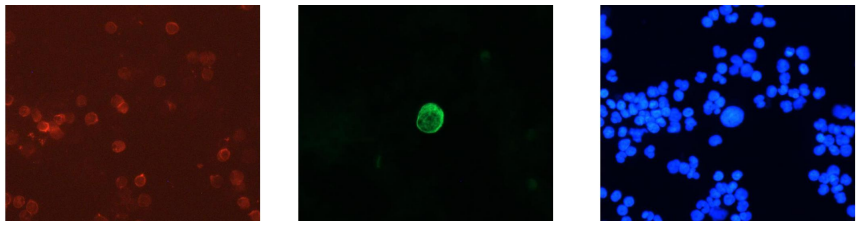
[1] The acronym MCF7 refers to *Michigan Cancer Foundation-7* which is who established the cell line in 1973

[2] The acronym SkBr3 refers to *Memorial Sloan–Kettering Cancer Center* which is who isolated the cell line



(a) CD45 (b) FITC (c) DAPI

Figure 4.1: MCF7 cell imaged using CytoTrack (the images are cropped out from the original)



(a) CD45 (b) FITC (c) DAPI

Figure 4.2: SkBr3 cell imaged using CytoTrack (the images are cropped out from the original)

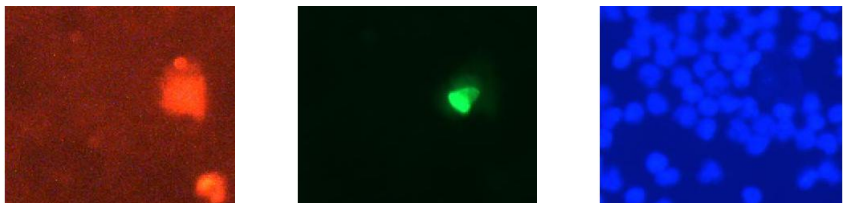


Figure 4.3: Example of a false positive image from the spiked samples (the images are cropped out from the original)

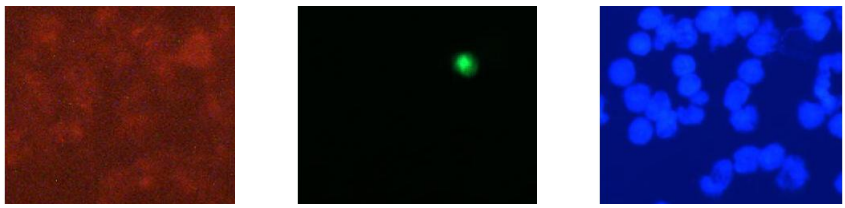


Figure 4.4: Example of a false positive image from the spiked samples (the images are cropped out from the original)

4.2 Patient Samples

Since it is of more clinical relevance to look at real patient data, some patient samples are also included in the data for this thesis. These images can be a bit harder to score than the images of spiked samples, since there are more variation in the appearances of the cells. In figure 4.5 and 4.6 some images are shown of CTCs from patient samples.

One data set stands apart from the rest, i.e. data set B171. In this sample the CTCs are generally bigger and have a high homogeneous FITC signal. An example is illustrated in figure 4.7. This difference is explained by B171 being a fresh sample, where the others are older samples taken from the freezer.

There are also some alignment issues in some of the images as seen in figure 4.8. This misalignment is a result of the instrument being heated when running for a long time. In the imaging process, the three types of images are taken separately, i.e. first all the DAPI images are taken, second the FITC images and third the CD45 images. This results in a significant time difference between the three images of one hot spot.

In figure 4.9 and 4.10 there are some examples of false positive images (i.e. images scored negative in the manual classification process). It is clear that there are great variations among both the positive and negative images.

The patient samples are all from mammary cancer patients and are all from patients enrolled in the XeNa project. The XeNa project is a clinical trial, which studies the drug Xeloda on invasive breast cancer patients [Lan13].

The data is divided into a training set and a testing set. The training set consist of images of all hot spots from 10 patients from the XeNa project, 5 who are CTC positive and 5 who are CTC negative, and images of some of the hot spots from 4 other patients from the XeNa project (B95, B129, B130 and B134). The test set consist of 7 data sets with images of all hot spots.

As can be seen from table 4.2 there are a lot more negative images than positives, which gives rise to a severe unbalanced data set. A balanced data set has thus been generated by using all positive images and an equal number of randomly selected negative images. This balanced data set are used for the training.

Data set	No. of positives	No. of negatives
B95	5	16
B97	12	269
B102	0	292
B113	5	60
B127	0	187
B129	1	13
B130	25	17
B133	8	225
B134	6	1
B137	0	167
B143	0	208
B161	0	233
B163	48	115
B171	15	219
Total	125	2022

Table 4.2: Number of image sets scored positive and negative by the operator in the patient data.

Figure 4.5: Example of a CTC positive image set from the patient data (the images are cropped out from the original)

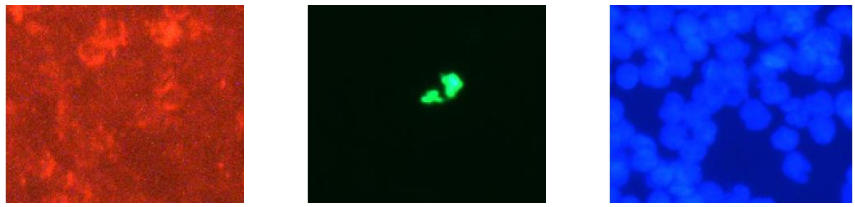


Figure 4.6: Example of a CTC positive image set from the patient data (the images are cropped out from the original)

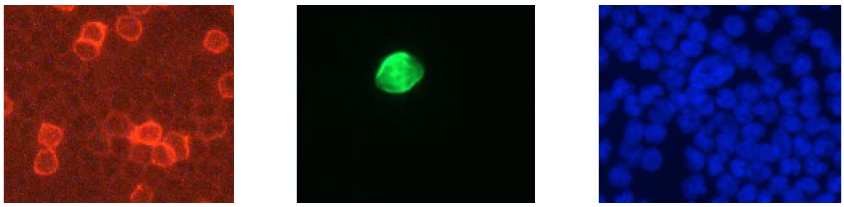


Figure 4.7: Example of a CTC positive image set from the B171 data set (the images are cropped out from the original)

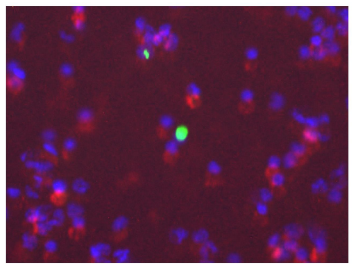


Figure 4.8: Image example of misalingment, the image is an overlay between the three channels and the images used for the overlay are cropped out from the original

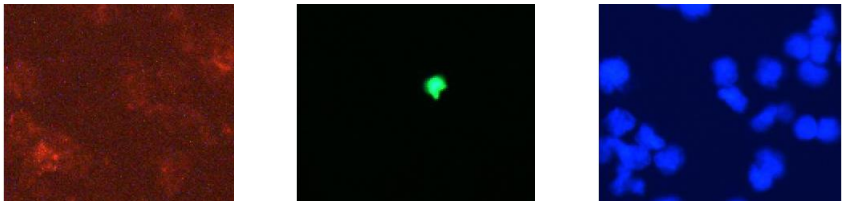


Figure 4.9: Example of a false positive image from the patient data (the images are cropped out from the original)

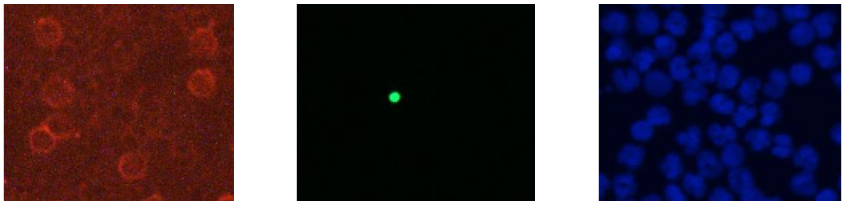


Figure 4.10: Example of a false positive image from the patient data (the images are cropped out from the original)

Chapter 5

Methods

All computations are implemented in Matlab (The Mathworks Inc.) in version 8.3.0.532 for windows. Matlab is chosen since it is excellent for image analysis and it provides ease of programming. Furthermore Matlab offers various toolboxes and for this thesis both the *image processing toolbox* and the *statistics toolbox* are used. The specifications for the program are given in appendix C.

The different steps involved in the classification of the CTC images are illustrated in the flowchart in figure 5.1.

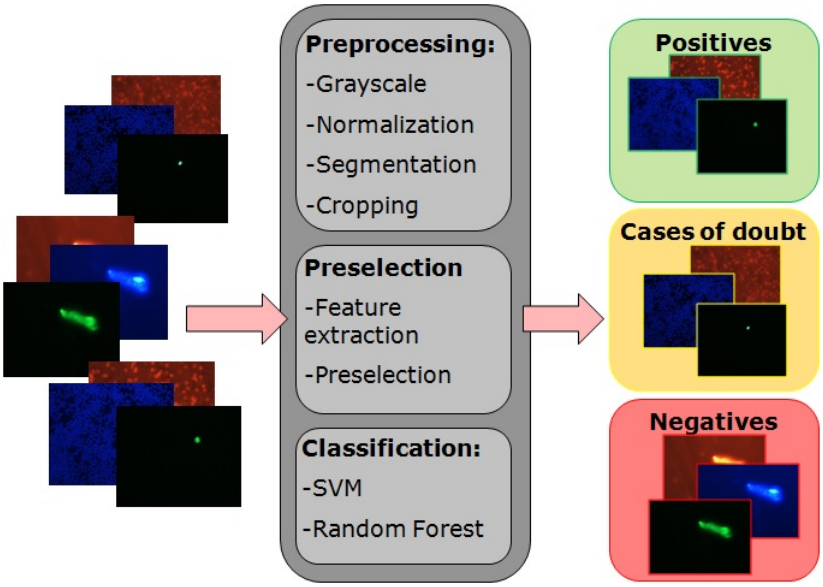


Figure 5.1: Flowchart over the steps involved in the classification of the CTC images

5.1 Preprocessing

Before the classification, some preprocessing steps are necessary. Features have to be extracted from the images. These features should preferably be based on the cell morphology and hence the cells have to be segmented. In order to segment the images, they are first converted into gray scale images.

5.1.1 Gray Scale

For the conversion of rgb images into gray scale, two methods are tested.

1. **Method 1:**
   The use of Matlab's function `rgb2gray`, which uses a weighted sum of the red, green and blue component:

   $$I_{gray} = 0.2989 \cdot R + 0.5870 \cdot G + 0.1140 \cdot B \tag{5.1}$$

   where R, G and B are the red, green and blue component, respectively.

2. **Method 2:**
   Only looking at one of the components, i.e. for CD45 only the red component, for FITC only the green component and for DAPI only the blue component is taken into account.

In figure 5.2 the two gray scale conversion methods are compared. In the first row are the CD45 images, in the second the FITC images and in the third row the DAPI images. In the first column of figure 5.2 the original images are shown, and in the second and third column method 1 and 2 are shown, respectively.

The comparison in figure 5.2 clearly show a higher contrast for the images obtained using method number 2 compared with method number 1. Since a high contrast will make the segmentation easier method number 2 is chosen and used for the final scoring program.

After the conversion the pixel values are normalized to achieve consistency in the pixel range, i.e. to make the individual images more comparable. The normalization is done by use of 5% and 95% of the maximum value, i.e.

$$I_{norm} = 255 \cdot \frac{I_{gray} - max_{5\%}}{max_{95\%} - max_{5\%}} \tag{5.2}$$

where I_{gray} is the gray scale image, and $max_{5\%}$ and $max_{95\%}$ are 5% and 95% of the maximum values, respectively.

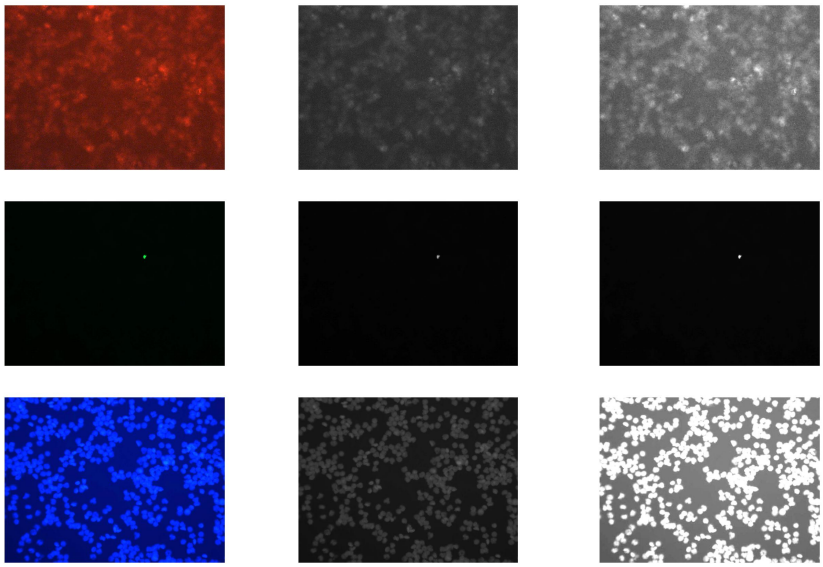


Figure 5.2: Comparison of two gray scale methods. Top: CD45, middle: FITC and bottom: DAPI. First column: Original images, second column: `rgb2gray` applied to images, third column: Only one channel.

5.1.2 Segmentation

The images have to be segmented to make it possible to extract morphological features from the cells. The first step in the segmentation is to only segment the FITC images, and from this determine a *region of interest* (ROI). This ROI is defined as the box that exactly contains the hot spot found in FITC, plus 10 pixels in each direction. The ROI is cropped out of the three images, i.e. CD45, FITC and DAPI. The segmentation of the CD45 and DAPI images is only performed on the cropped images.

There are many different approaches to image segmentation, and it is important to find the right method for the right images. All the cells in the images are objects of interest, these are all clearly visible from the background and a simple histogram-based method should be sufficient. Two methods have been tested,

a fixed threshold where a threshold is determined for each type of image, and Otsu's method for generating an automatic threshold.

1. **Fixed Threshold**
   A fixed threshold is determined for each type of image. The optimal threshold is determined based on the histograms of several images. A threshold was chosen for the CD45-, FITC- and DAPI images independently from each other. The thresholds that gave the best results on a scale from 0 to 255 were 200 for CD45 and DAPI, and 60 for FITC. Some results of using this method can be seen in figure 5.3 in the third column.

2. **Otsu's Method**
   In Otsu's method for applying a threshold to an image, the histogram is normalized and assumed to be a probability distribution, i.e. [Ots79]

   $$p_i = n_i/N, \quad p_i \geq 0, \quad \Sigma_{i=1}^{L} p_i = 1 \tag{5.3}$$

   where n_i is the number of pixels at level i, N is the total number of pixels and L is the gray levels.

   The optimal threshold for separating the two classes, C_0 and C_1, is defined as the one that gives the lowest within-class variance σ_W^2

   $$\sigma_W^2 = \omega_0 \sigma_0^2 + \omega_1 \sigma_1^2 \tag{5.4}$$

   where ω_0 and ω_1 are the probabilities of class C_0 and C_1, respectively, and σ_0^2 and σ_1^2 are the variances of class C_0 and C_1, respectively [Ots79].

   This method is implemented using Matlab's function `graythresh` from the image processing toolbox. In order to be as sensitive as possible and to avoid segmenting a lot of noise, an offset of 20 is added to the FITC and CD45 images. The offset is not added to the DAPI images since these are often saturated and thus have a threshold of 255. Some results from this method can be seen in figure 5.3 in the second column.

For both methods all objects that have a size of 100 pixels ($14\mu m^2$) or less are considered noise and are removed from the images. Furthermore to be sure that even the smallest objects are segmented in the FITC images, a dilation operation is applied to the gray scale image and after segmentation an erosion is applied, in order not to overestimate the sizes of the objects. For the structuring element a disk shape with a radius of 5 and 3 are used, respectively. All hot spots imaged

will be found close to the center of the images, and hence all borders are cleared for objects in the uncropped binary FITC images.

The two methods are compared in figure 5.3. In the first column are the original gray scale image for CD45, FITC and DAPI, respectively. Otsu's method are applied to the images in column two, and in the third column the results of applying a fixed threshold is illustrated.

For some images the fixed threshold performed excellent, as seen in the FITC images of figure 5.3, but as can be seen in the DAPI and CD45 images of figure 5.3 that did not apply to all images. The background and the intensity of the cells varies significantly throughout the data, and hence a fixed threshold does not give an optimal segmentation. Otsu's method on the other hand performed much more robust and thus this was the method of choice for the segmentation.

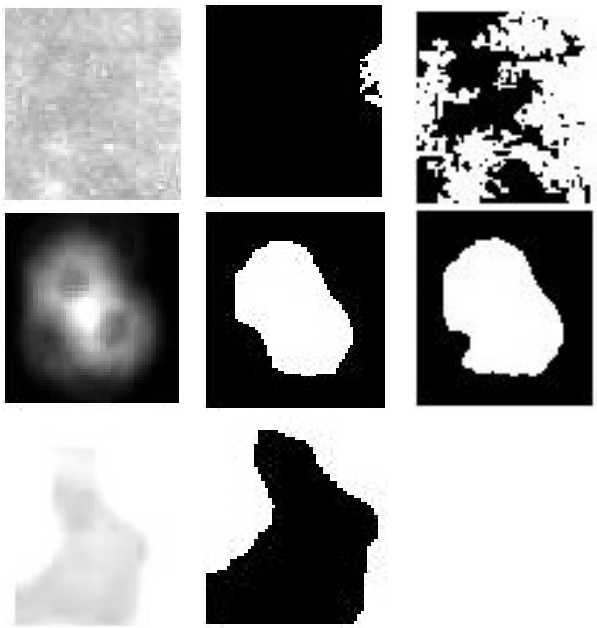


Figure 5.3: Comparison of two threshold methods. Top: CD45, middle: FITC and bottom: DAPI. First column: Original images, second column: Otsu's method, third column: Fixed threshold.

5.1.3 Feature Extraction

As a first step in the feature extraction some obvious negatives are classified. These are classified based on the area and convex area of the *binary large object* (BLOB) found in the FITC images. Furthermore the overlap between the BLOB in the FITC image and a BLOB in the CD45 image is also used for this preselection. The criteria for obvious negatives are image sets where no BLOBs are found in FITC, where the BLOB has an area or convex area larger than 10,000 pixels ($\tilde{1}406\mu m^2$) in FITC and image sets where there are 100% overlap between the BLOB found in FITC and a BLOB found in CD45. Besides these obvious negative images, all image sets where the FITC contains more than one object are classified as cases of doubts. This is done to avoid classifying the same image twice.

After this preselection 13 basic morphological features are extracted from the images. These are area, convex area, solidity, eccentricity, diameter, perimeter, minor axis/major axis, contrast, overlap between FITC and DAPI, overlap between FITC and CD45 and mean intensity in FITC, DAPI and CD45. These features are chosen since they e.g. describe the shape of the cell, which should have an impact on the classification. Furthermore both intensities and contrast are thought to possibly have an effect on the classification. Generally a relatively high intensity is observed for the CTCs in the FITC images, and a bit lower intensity is observed for the nucleus of CTCs (lower intensity for the blob in DAPI compared with other DAPI blobs). The first eight features are measured in the FITC images. The first seven features and the intensities are measured using the `regionprops` function which is directly available from the *image processing toolbox* in Matlab. The contrast is defined as the standard deviation of the pixel values. The overlaps are defined as the area where both images have values of 1, i.e. where the BLOB overlaps, divided by the entire area of the BLOB in FITC. The overlap is thus a value between 0 and 1, where 1 is defined as 100% overlap.

The distribution of these features are plotted as cumulative histograms. This is done to see how well the chosen features separate the positive and negative image sets. The histograms for both the patient data and the spiked data can be seen in appendix B. From these histograms it is clear that the positive and negative class are more well separated for the spiked data than for the patient data.

5.2 Classification Methods

After the preprocessing the unclassified images have to be classified based on the computed features. Classification is the task of dividing unseen data into the right classes. For this a classification algorithm is used, which is trained on training data where the classes are already known. In this case a data set marked by a trained operator is used, since there is no ground truth. There are a number of different classification algorithms to choose from and some of the popular choices include *random forest* and *support vector machines*. These two algorithms generally performs well, which e.g. is shown in [HEH+06], and are the methods of choice for this thesis.

5.2.1 Random Forest

Random forest is a classification algorithm that is built from decision trees. A decision tree is a tree that classifies the input data by sorting based on the values of the features extracted from the data. The features are represented by *decision nodes*, where the topmost are called *root nodes*. The root nodes are connected to the *internal nodes* through branches. Each branch do a binary split and ultimately end out in *leaf nodes* which holds the classification of the data [HGFO14]. An example of a decision tree can be seen figure 5.4.

In a random forest many decision trees are grown. Each tree is trained from a

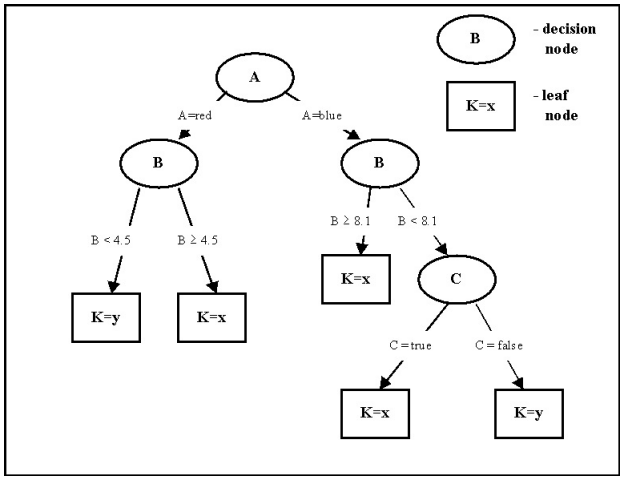


Figure 5.4: Example of a decision tree from [HGFO14]

randomly chosen subset of the input features $mtry$, which is a way of reducing the dimensionality of the features

$$\mathbb{R}^n \rightarrow \mathbb{R}^{mtry}, \quad mtry \ll n \tag{5.5}$$

where n is total the number of features.

From a random forest it is possible to associate a distribution with each of the internal nodes as well as the leaf nodes. These distributions can be used to choose the split with highest confidence. A probabilistic predictor model is associated with the leaf nodes of all the trees

$$p_t(y|v) \tag{5.6}$$

where y is the class label, i.e. either CTC or *not a* CTC, v is the feature values, and $t = \{1, 2, ..., T\}$ is the trees [BC] [BS09].

An average over all the trees can give the full prediction for the forest

$$p(y|v) = \frac{1}{T} \sum_{t=1}^{T} p_t(y|v) \tag{5.7}$$

From these probabilities it is possible to compute a *Receiver operating characteristic* (ROC) curve, which can be used to choose optimal thresholds to e.g. avoid false negatives.

Different variables should be taken into consideration when constructing a random forest. An important variable is the size of the forest, which is given by the number of trees. By adding more trees to a forest it is possible to reduce the variance of the model and keep the bias of the trees. Other important variables are the minimum number of leaf nodes, which is used as a stopping criterion in the training for each tree, and the number of features to select from each split ($mtry$) [BC] [BS09].

Random forest can also be used for estimating the features importance, by randomly permuting the *out-of-bag* (oob) features. The oob features are the ones not incorporated in the construction of the tree. After having trained each tree in the in-bag samples, the oob samples are tested and the number of votes for the correct class is counted. For each of the n features their values in the oob samples are randomly permuted and predictions are computed. The number of votes for the correct class of the permuted oob samples is subtracted from the number of votes for the correct class of the non-permuted oob samples. The average over all trees in the forest is used as a raw importance score for the feature n. If this importance score is low it implies that changing this feature will not have a significant effect on the scoring of the data and vice versa [BC] [BS09].

5.2.2 Support Vector Machine

Support vector machines (SVMs) are supervised learning models which can be used for binary classification of data. The input vectors are mapped into a high dimensional feature space called Z. The mapping is done using non-linear mapping which is chosen prior to the classification. The data is separated into two classes by use of a linear separating plane called a *hyperplane*. There will be an infinite amount of hyperplanes that can separate the training data, but not all will generalize well. The optimal hyperplane is defined as the plane that gives the maximal margin between vectors of the two classes as can be seen in figure 5.5. Only a small amount of the training data have to be taken into account for this approach, i.e. the support vectors [CV95] [TM14].

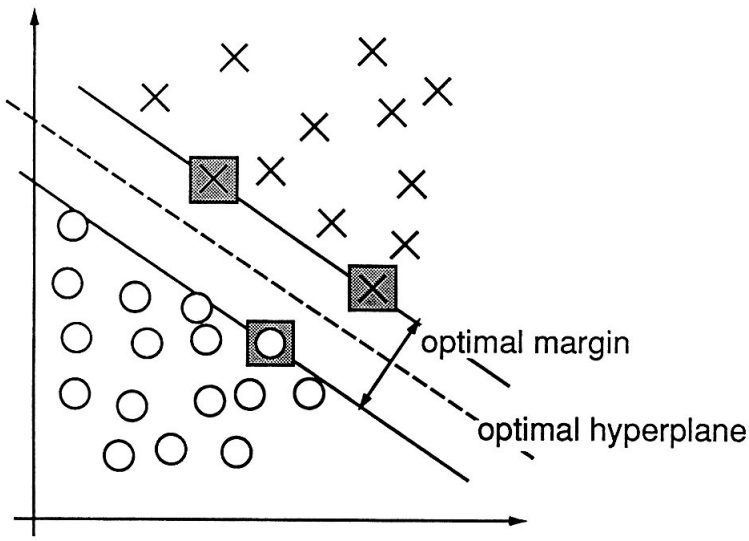


Figure 5.5: Illustration of the principle in SVM [CV95]

The algorithm is first trained using a marked training set, $\{\boldsymbol{x}_i, y_i\}(i = 1, ..., n)$, where $\boldsymbol{x}_i \in \boldsymbol{R}^d$ and $y_i \in \{-1, +1\}$ (corresponding to the two classes *not a CTC* and *CTC*). The equation for the optimal hyperplane is thus,

$$\boldsymbol{w}_0 \cdot \boldsymbol{x} + b_0 = 0 \tag{5.8}$$

where $\boldsymbol{w}_0$ is the weight vector and b_0 is the bias [CV95].

The decision function is given as

$$I(\boldsymbol{z}) = sign\left(\sum_{support\ vectors} \boldsymbol{w}_0 \cdot \boldsymbol{z} + b_0\right) \tag{5.9}$$

which determined the classification of the vector z.

For inseparable data, i.e. can not be separated without error, the goal is to reduce the number of errors. For this a *soft margin* is used, which is given as

$$min\left(\frac{1}{2}\boldsymbol{w}^2 + C\left(\sum_i \xi_i\right)\right) \tag{5.10}$$

under the constraints

$$\begin{aligned} y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i + b) &\geq 1 - \xi_i \\ \xi_i &\geq 0 \end{aligned} \tag{5.11}$$

Increasing C will allow a more strict separation between classes and reducing C towards zero will make a wrong classifications less important [CV95] [TM14].

Not all data can be well separated using a simple hyperplane. For these cases a *kernel function* $K(\boldsymbol{x}, \boldsymbol{y})$ is used to map the data into higher dimensional spaces. In a linear space S, the function φ maps $\boldsymbol{x}$ to S.

$$K(\boldsymbol{x}, \boldsymbol{y}) = \varphi(\boldsymbol{x}) \cdot \varphi(\boldsymbol{y}) \tag{5.12}$$

Different functions can be used as kernel functions and in this thesis two different kernels, besides the linear, are tested, i.e. a polynomial and a radial basis function (rbf). The polynomial kernel is of the form

$$K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}\cdot\boldsymbol{y} + 1)^d \tag{5.13}$$

where d is the order of the polynomial. The rbf is given as

$$K(\boldsymbol{x}, \boldsymbol{y}) = exp\left(-\frac{|\boldsymbol{x} - \boldsymbol{y}|^2}{\sigma^2}\right) \tag{5.14}$$

where σ^2 is the variance [CV95] [TM14].

The output of SVM is not simple scores as it is for random forest. Instead the output is given by

$$f(\boldsymbol{x}) = h(\boldsymbol{x}) + b \tag{5.15}$$

where

$$h(\boldsymbol{x}) = \sum_i y_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) \tag{5.16}$$

where α_i is the weight of a support vector in the feature space and $\boldsymbol{x}_i$ is the image of a support vector in input space [CV95]. From this output a posterior probability can be calculated by fitting a sigmoid function to $f(x)$

$$P(y = 1|f) = \frac{1}{1 + exp(Af + B)} \tag{5.17}$$

where the parameters A and B are trained discriminatively using the training set [Pla99].

Chapter 6

Results

In order to find the optimal method for classifying the data, 4 different algorithms are tested. These are the random forest and SVM with 3 different kernel functions (linear, polynomial and radial basis function (rbf)). For this the feature importance is tested followed by cross validation and testing of the 4 algorithms.

6.1 Feature importance

The importance of the features is tested using *random forest*, see section 5.2.1. In figure 6.1 each feature is given as a number along the x-axis (see table 6.1) and the feature importance is given along the y-axis. From this it is clear that not all features contribute equally to the separation of the data. Based on these histograms the 3 and 6 most important features for both patient and spiked data are found and used for further testing.

In figure 6.1 it can be seen that there are big variations in the importance of the individual features for the spiked data. There are also some variations for the patient data, but these are not as big as for the spiked data.

The 6 most important features for the patient data given in descending order are:

1. Mean intensity measured in FITC
2. Convex area
3. Area
4. Diameter
5. Perimeter
6. Contrast

The 6 most important features for the spiked data given in descending order are:

1. FITC/DAPI overlap
2. FITC/CD45 overlap
3. Contrast
4. Convex area
5. Mean intensity measured in FITC
6. Area

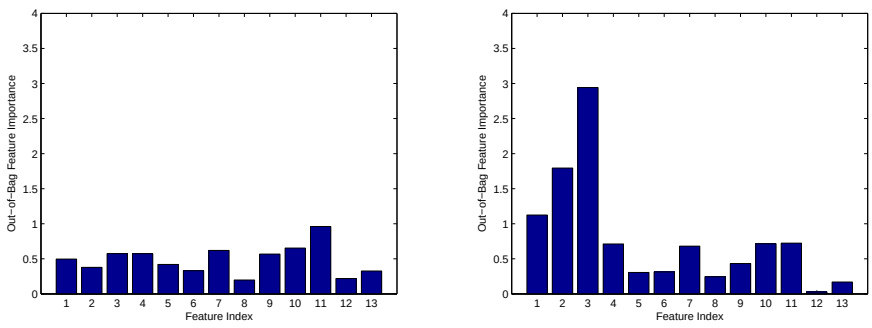


Figure 6.1: Illustration of the importance of the different features for both patient (left) and spiked (right) data.

Number	Feature
1	Contrast
2	FITC/CD45 overlap
3	FITC/DAPI overlap
4	Area
5	Minor-/major axis
6	Eccentricity
7	Diameter
8	Solidity
9	Perimeter
10	Convex area
11	Intensity FITC
12	Intensity CD45
13	Intensity DAPI

Table 6.1: Features corresponding to the numbers in figure 6.1.

6.2 Cross Validation

The 4 algorithms are cross validated using 10-fold cross validation. In 10-fold cross validation the training data is randomly divided into 10 equal sized subsamples and the training is performed on 9 of the 10 subsamples, the last subsample is then used for validation. This is repeated 10 times using a new subsample for validation each time, i.e. all subsamples are used for validation exactly once. The 10 results are then averaged to give an estimate of the performance of the algorithm.

6.2.1 Random Forest

For random forest 3 variables have to be determined, these are the number of trees, the minimum number of leaf nodes and the number of variables to sample (*mtry*). The first variable, *number of trees*, are tested using the *out-of-bag error* (oob error). The oob error is plotted as a function of the number of trees as seen in figure 6.2. It can be seen that the error does not change much after reaching about 200-300 trees, but to be on the safe side 500 trees are used.

Furthermore it can be seen that for the patient data, using all features give significantly lower oob error compared with using 3 or 6. For spiked data, using 6 features gives the lowest oob error. It is also worth noticing that the oob error in general is much lower for the spiked samples compared with the patient samples.

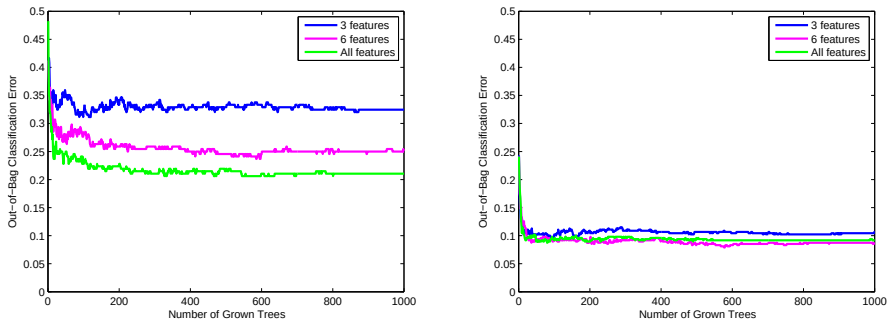


Figure 6.2: Oob error as a function of the number of trees for patient data (left) and spiked data (right).

After the number of trees is determined, a 10 fold cross validation is performed where the minimum number of leaf nodes and the number variables to sample

are varied. The results from the cross validation are illustrated in figure 6.3 and 6.4 where the sensitivity and specificity is computed for each combination. The sensitivities are shown to the left and the specificities are shown to the right. In figure 6.3 the patient data is used and in figure 6.4 the spiked data is used. For both figures the top row is including 3 features, the middle is including 6 features and the bottom row is including all features. From these figures it is clear that both higher sensitivities and specificities are reached for the spiked data compared with the patient data.

The goal is to reach as high sensitivity and specificty for one of the combinations. For this thesis the sensitivity is weighted higher than the specificity, since the goal is to completely eliminate false negatives.

From the two figures it is noticed that the sensitivities are generally higher than the specificities, and relatively high sensitivities are found for all combinations. There are bigger variations in the specificities, e.g. the specificities are clearly higher using all features for the patient data compared with using 3 or 6 features. For the spiked data the variations for the specificities are not as clear, but they are generally higher using 6 features compared with using all or 3 features. The best combination for both patient and spiked data is given in table 6.2, and these combinations are used for the testing.

	Patient	Spiked
No. of features	All	6
No. of variables to sample	9	4
Minimum number of leaf nodes	6	2
Sensitivity	**0.8518**	**0.9572**
Specificity	**0.8178**	**0.8907**

Table 6.2: Summary of the variables that give the best combination of sensitivity and specificity for random forest.

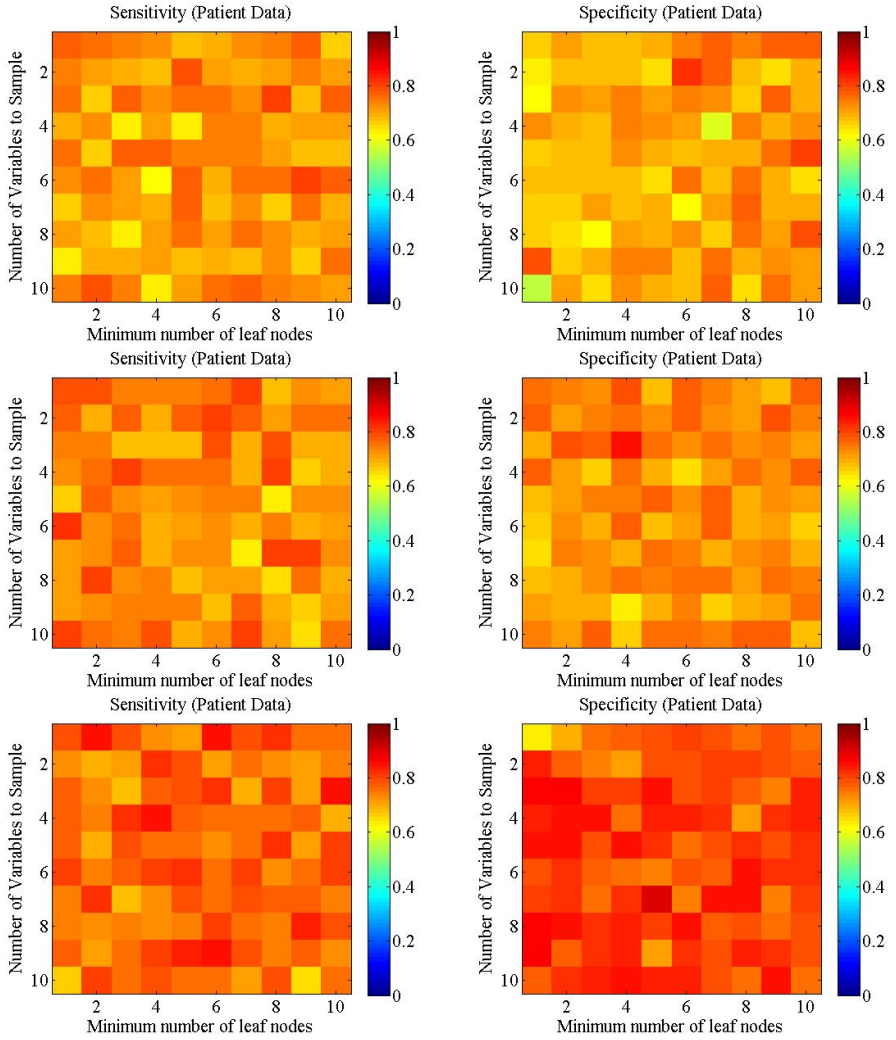


Figure 6.3: Sensitivities and specificities computed from 10-fold cross validation for the patient data. In the top row the 3 most important features are included, in the middle row the 6 most important features are included and in the bottom row all features are included.

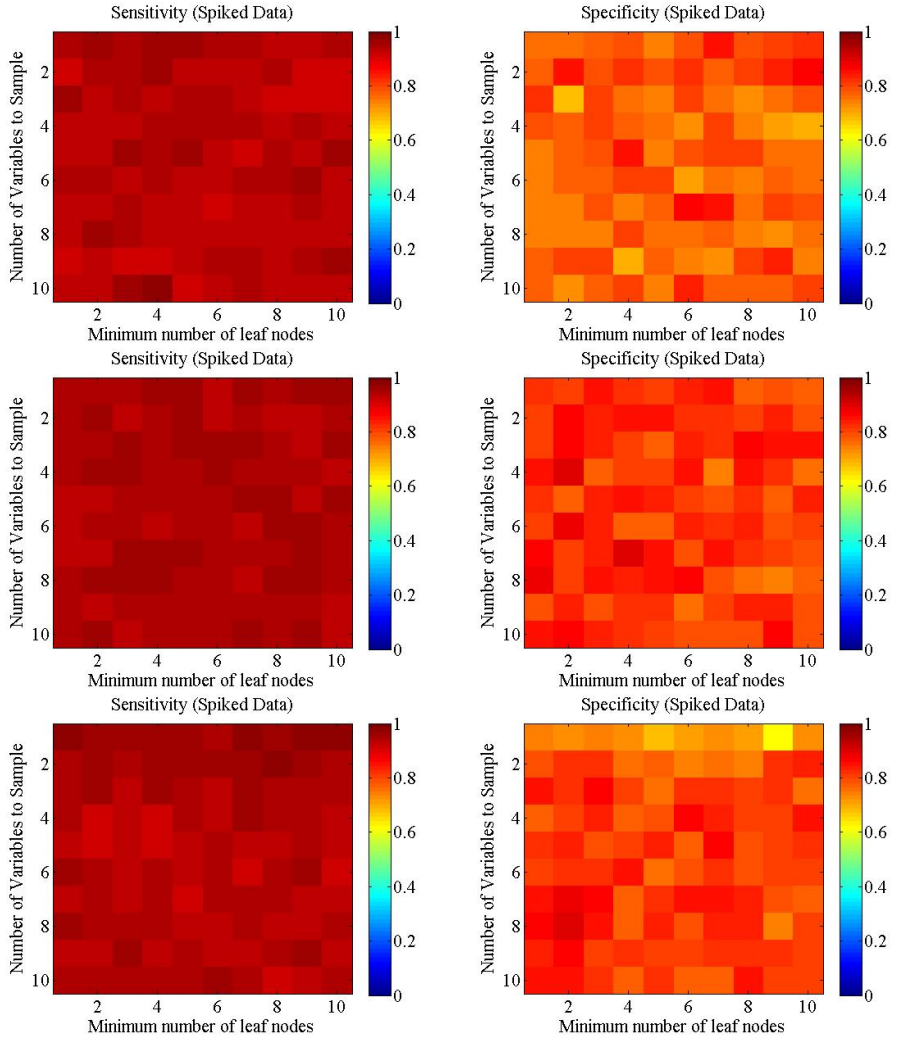


Figure 6.4: Sensitivities and specificities computed from 10-fold cross validation for the spiked data. In the top row the 3 most important features are included, in the middle row the 6 most important features are included and in the bottom row all features are included.

6.2.2 SVM Linear

For the linear kernel of SVM a 10 fold cross validation is performed where the variable C and the kernel scale is varied. The results from the cross validation are illustrated in figure 6.5 and 6.6 where the sensitivity and specificity is computed for each combination. The sensitivities are shown to the left and the specificities are shown to the right. In figure 6.5 the patient data is used and in figure 6.6 the spiked data is used. For both figures the top row is including 3 features, the middle is including 6 features and the bottom row is including all features. From these figures it is clear that both higher sensitivities and specificities are reached for the spiked data compared with the patient data, as was also seen from the random forest.

As for the random forest the sensitivities are higher than the specificities and there are bigger variations in the specificities. In this case the highest specificities are achieved using 6 features for both patient and spiked data. The best combinations for both patient and spiked data are given in table 6.3, and these combinations are used for the testing.

It is worth noticing that the specificities are generally lower using the linear kernel compared with using random forest for both patient and spiked data. Furthermore for the patient data the computed sensitivities are also lower compared with the results from using random forest.

	Patient	Spiked
No. of features	6	6
Kernel scale	1	1
C	100	10
Sensitivity	**0.7982**	**0.9548**
Specificity	**0.6491**	**0.8175**

Table 6.3: Summary of the variables that give the best combination of sensitivity and specificity for SVM with a linear kernel.

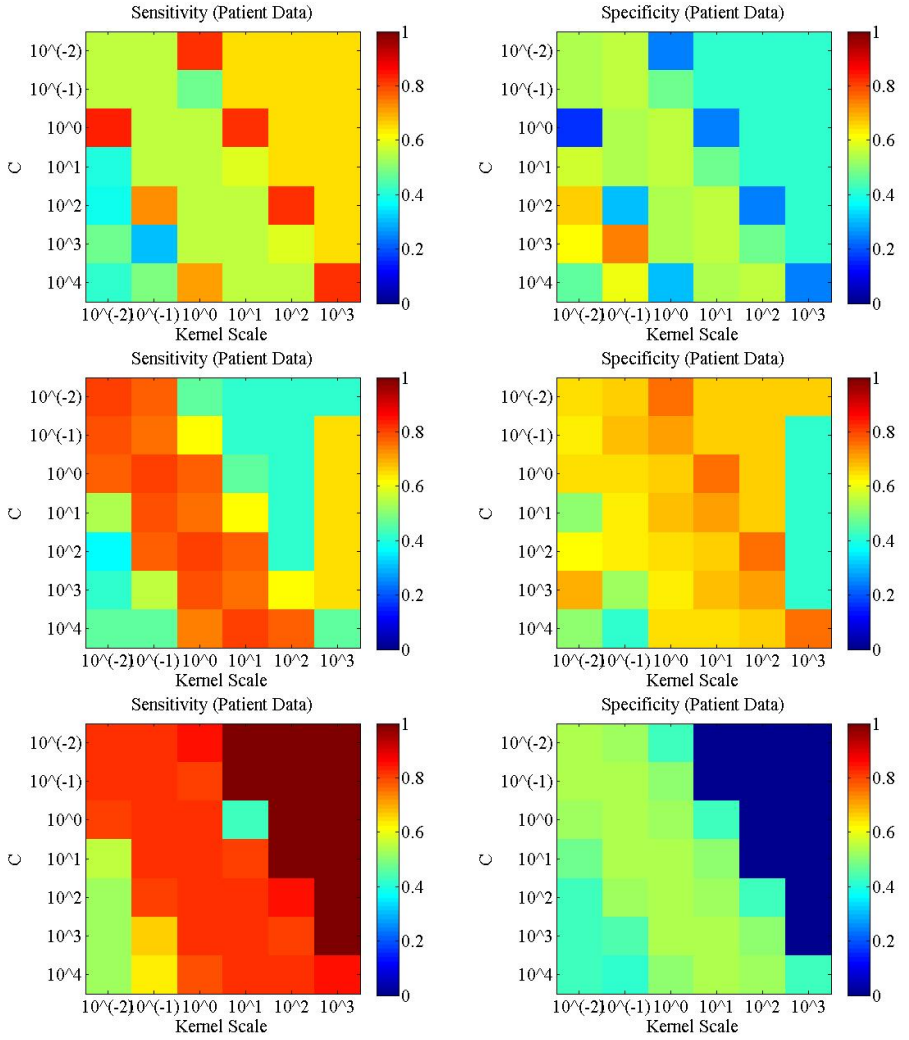


Figure 6.5: Sensitivities and specificities computed from 10-fold cross validation for the patient data. In the top row the 3 most important features are included, in the middle row the 6 most important features are included and in the bottom row all features are included.

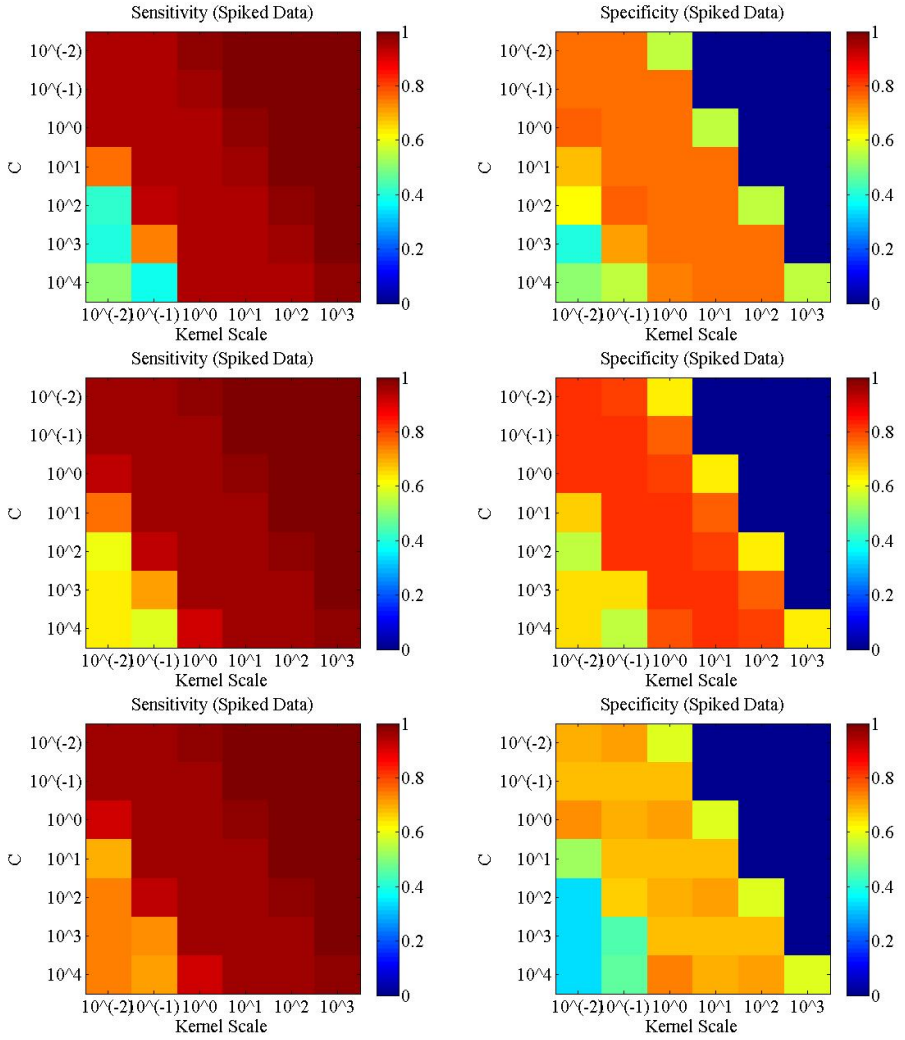


Figure 6.6: Sensitivities and specificities computed from 10-fold cross validation for the spiked data. In the top row the 3 most important features are included, in the middle row the 6 most important features are included and in the bottom row all features are included.

6.2.3 SVM Polynomial

For SVM using the polynomial kernel a 10 fold cross validation is performed where the variable C and the order of the polynomial is varied. The results from the cross validation are illustrated in figure 6.7 and 6.8 where the sensitivity and specificity is computed for each combination. The sensitivities are shown to the left and the specificities are shown to the right. In figure 6.7 the patient data is used and in figure 6.8 the spiked data is used. For both figures the top row is including 3 features, the middle is including 6 features and the bottom row is including all features.

For the polynomial kernel the sensitivities are higher than the specificities, as seen for both random forest and SVM with a linear kernel. As for the linear kernel the highest specificities are reached using 6 features for both patient and spiked data. The best combination for both patient and spiked data are given in table 6.4, and these combinations are used for the testing.

The sensitivities and specificities for the spiked samples are comparable to the ones computed using the linear kernel and using random forest. For the patient data the results from using the polynomial kernel are comparable to using the linear kernel and hence the results are worse for the polynomial kernel compared with using random forest.

	Patient	Spiked
No. of features	6	6
Order	3	2
C	1	10
Sensitivity	**0.7719**	**0.9578**
Specificity	**0.7105**	**0.8467**

Table 6.4: Summary of the variables that give the best combination of sensitivity and specificity for SVM with a polynomial kernel.

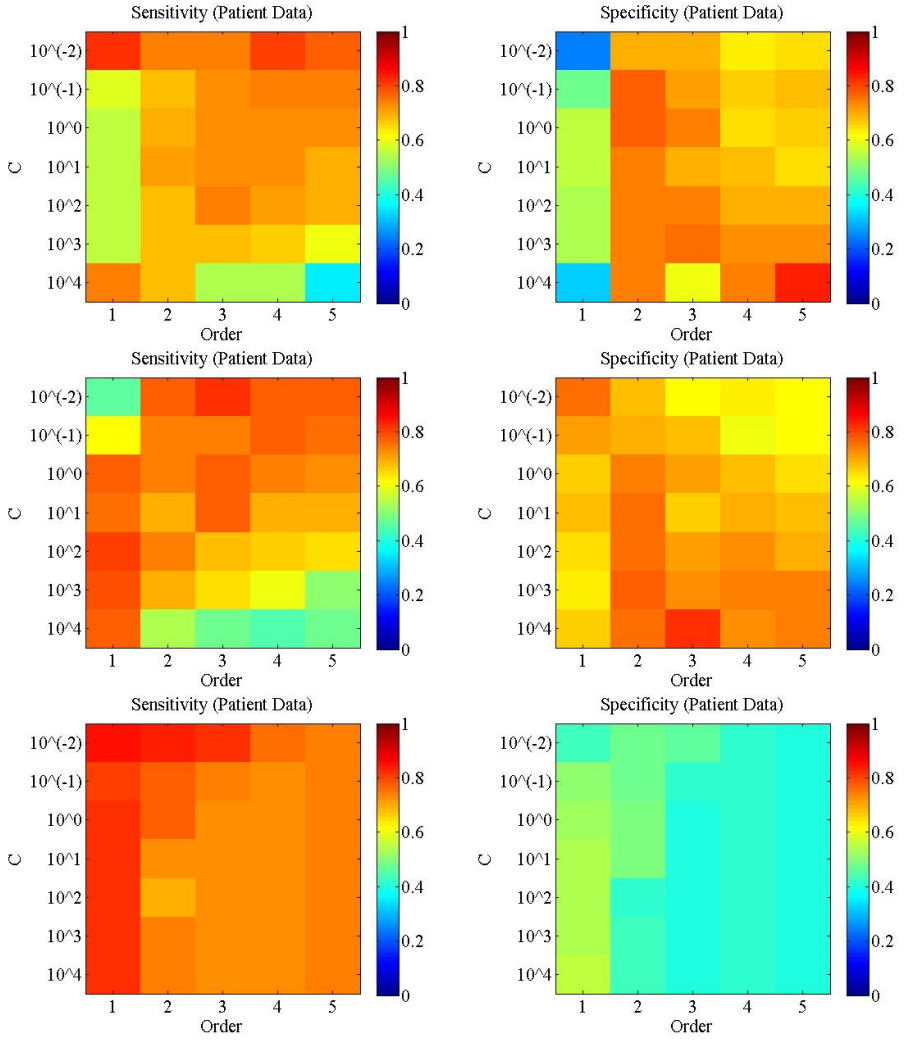


Figure 6.7: Sensitivities and specificities computed from 10-fold cross validation for the patient data. In the top row the 3 most important features are included, in the middle row the 6 most important features are included and in the bottom row all features are included.

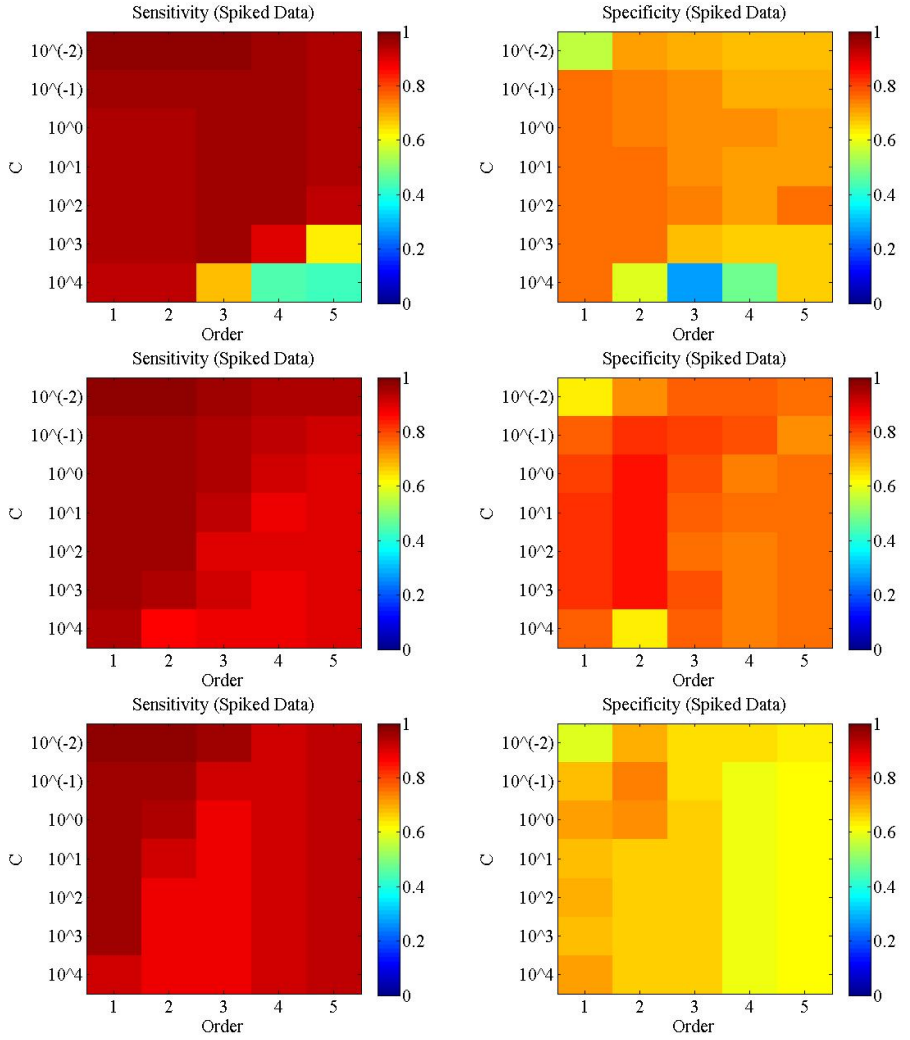


Figure 6.8: Sensitivities and specificities computed from 10-fold cross validation for the spiked data. In the top row the 3 most important features are included, in the middle row the 6 most important features are included and in the bottom row all features are included.

6.2.4 SVM rbf

For SVM with an rbf kernel a 10 fold cross validation is performed where the variable C and the kernel scale (σ) is varied. The results from the cross validation are illustrated in figure 6.9 and 6.10 where the sensitivity and specificity is computed for each combination. The sensitivities are shown to the left and the specificities are shown to the right. In figure 6.9 the patient data is used and in figure 6.10 the spiked data is used. For both figures the top row is including 3 features, the middle is including 6 features and the bottom row is including all features. As for the other algorithms both higher sensitivities and specificities are reached for the spiked data compared with the patient data.

In this case the highest specificities are achieved using 6 features for both patient and spiked data. The best combination for both patient and spiked data are given in table 6.5, and these combinations are used for the testing.

From table 6.5 it is clear that the three SVM algorithm's performances are comparable for the patient data, but performs a bit worse than random forest. For the spiked data all four algorithm's performances are comparable.

	Patient	Spiked
No. of features	6	6
Kernel Scale (σ)	10	10
C	10	100
Sensitivity	**0.7193**	**0.9639**
Specificity	**0.7193**	**0.8394**

Table 6.5: Summary of the variables that give the best combination of sensitivity and specificity for SVM with rbf kernel.

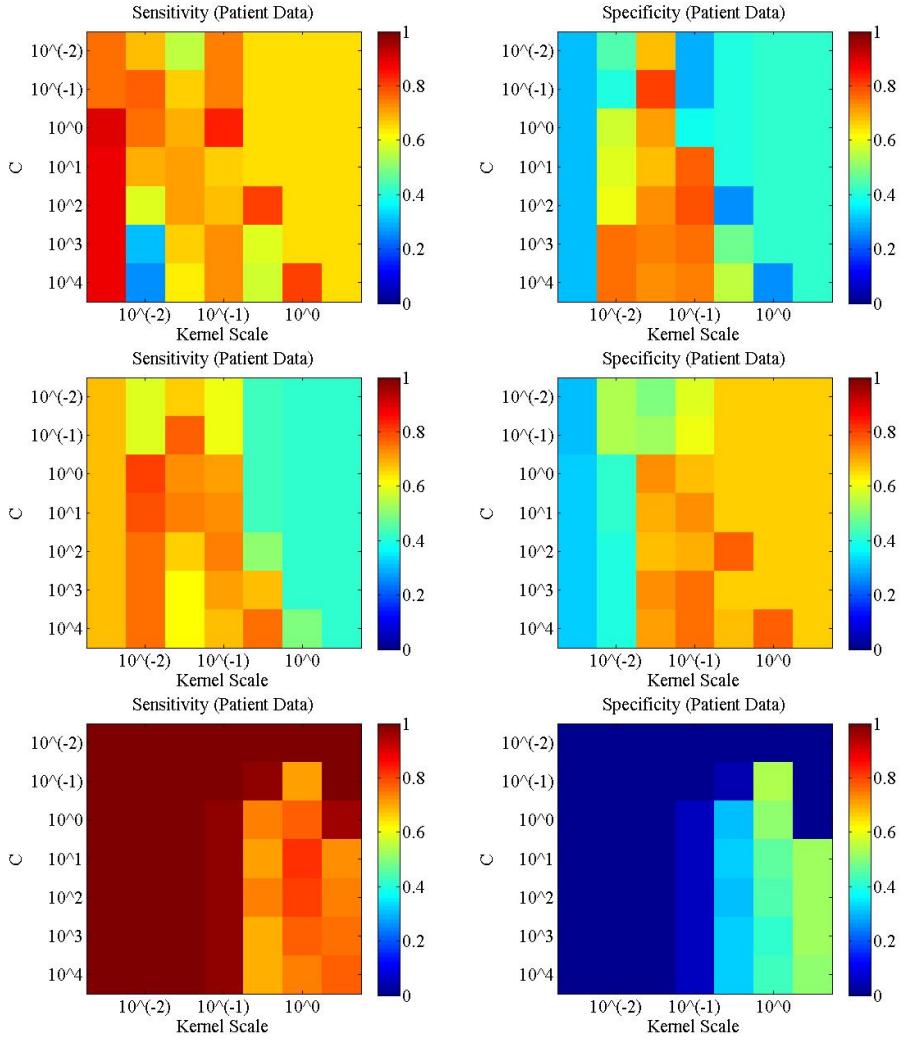


Figure 6.9: Sensitivities and specificities computed from 10-fold cross validation for the patient data. In the top row the 3 most important features are included, in the middle row the 6 most important features are included and in the bottom row all features are included.

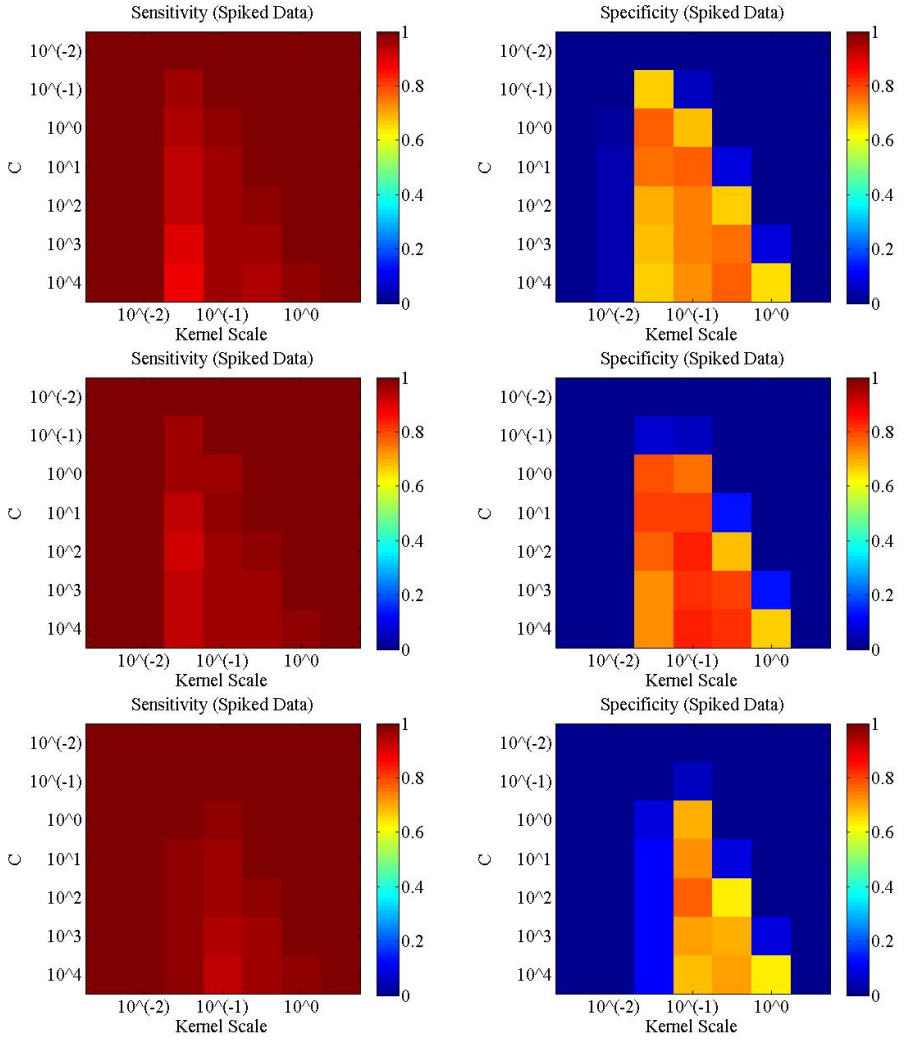


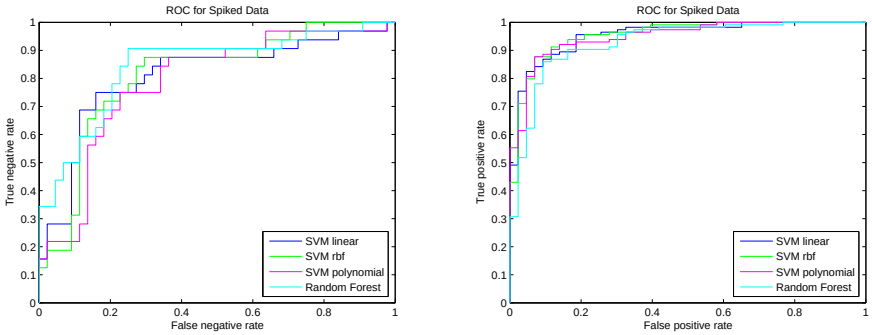
Figure 6.10: Sensitivities and specificities computed from 10-fold cross validation for the spiked data. In the top row the 3 most important features are included, in the middle row the 6 most important features are included and in the bottom row all features are included.

6.3 ROC Curves

A *receiver operating characteristic* (ROC) curve is a graphical representation of the performance of a binary classifier when varying the discriminant threshold. The ROC curve is computed by plotting the true positive rate as a function of the false positive rate.

From the cross validation, the 4 algorithms, for both patient and spiked data, that performs the best are chosen for further testing. In figure 6.11 the ROC curves for these 4 algorithms for both patient data and spiked data are shown. There are no apparent differences in the performance of the 4 algorithms, but it seems like random forest performs a bit better for the patient data, as expected.

Thresholds are found based on the ROC curves. Since it is more important to avoid false negatives than false positives, the thresholds are chosen so that the *true positive rate* is equal to 1. The rest is classified as negatives.



(a) Patient samples **(b)** Spiked samples

Figure 6.11: ROC curves for patient data (to the left) and spiked data (to the right).

6.4 Results From Testing

The four algorithms trained on the spiked data are tested on the spiked test data. The results from this test can be seen in table 6.6. It should be noted that beside the data in the table, 163 image sets are classified as cases of doubts, and hence not incorporated in the table. In table 6.6 it is noticed that for three out of four algorithms there are no false negatives. The one false negative can be seen in figure 6.12.

	Tp	Tn	Fp	Fn	Sensitivity	Specificity
SVM linear	179	35	23	0	1	0.6034
SVM polynomial	178	44	14	1	0.9944	0.7586
SVM rbf	179	39	19	0	1	0.6724
Random forest	179	31	27	0	1	0.5344

Table 6.6: Results from testing the algorithms based on the spiked data on the spiked testing data. Tp = True positive, Tn = True Negative, Fp = False positive and Fn = False negative.

Figure 6.12: Illustration of the false negative image from the spiked test data (the images are cropped out from the original)

Data set B171 looks more like the spiked data than the patient data. This is explained by B171 being a fresh sample. Since the test set is unknown, there is no information about whether the test samples are fresh or not. Therefore patient data is tested using both the algorithms trained on the spiked data and the algorithms trained on the patient data. The results can be seen in table 6.7. Besides the results in table 6.7 209 image sets are classified as cases of doubts and are not incorporated in the table. From table 6.7 it is clear that the algorithms trained on spiked data give higher specificities, but at the same time they give false negatives and thus lower sensitivities compared with the

algorithms trained on the patient data. All in all three image sets are scored false negative and these can be seen in figure 6.13, 6.14 and 6.15.

	Tp	Tn	Fp	Fn	Sensitivity	Specificity
SVM linear (Spiked)	21	236	184	2	0.9130	0.5619
SVM polynomial (Spiked)	20	346	74	3	0.8696	0.8238
SVM rbf (Spiked)	20	321	99	3	0.8696	0.7642
Random forest (Spiked)	22	246	174	1	0.9565	0.5857
SVM linear (Patient)	23	188	232	0	1	0.4476
SVM polynomial (Patient)	23	164	256	0	1	0.3905
SVM rbf (Patient)	23	194	226	0	1	0.4619
Random forest (Patient)	23	186	234	0	1	0.4429

Table 6.7: Results from testing all algorithms on the patient data. Tp = True positive, Tn = True Negative, Fp = False positive and Fn = False negative.

The images which are scored negative are sorted from the data, and the operator will only have to verify the data scored as positive and the data scored as cases of doubts manually. In order to determine how effective the individual algorithms are, the percentage of the true negatives which are actually scored negative can be seen in table 6.8. From this it is noticed that a larger portion of the data is generally scored negative from the patient data compared with the spiked data. The largest percentages are scored negative for the patient data scored with the algorithms trained on the spiked samples, but these are also the ones with most false negatives.

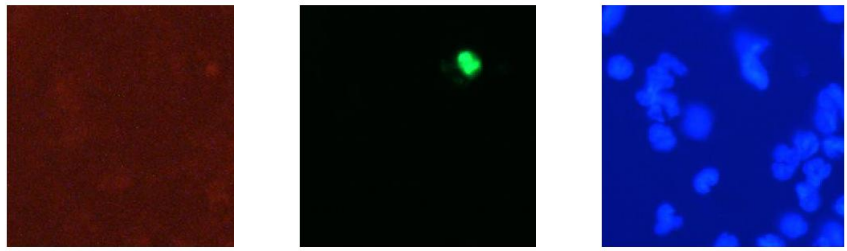


Figure 6.13: Illustration of a false negative image from the patient test data (the images are cropped out from the original)

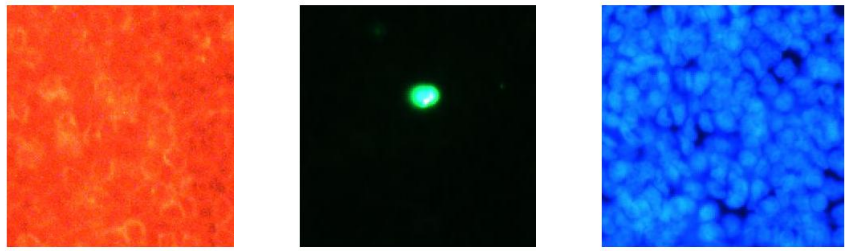


Figure 6.14: Illustration of a false negative image from the patient test data (the images are cropped out from the original)

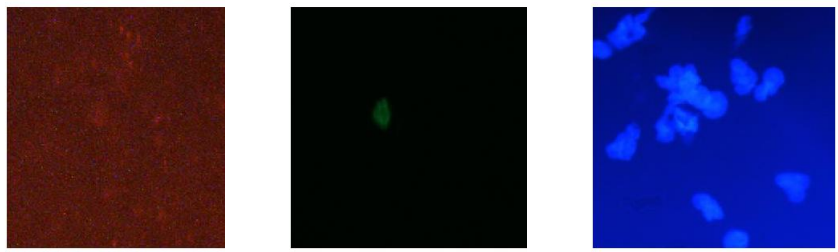


Figure 6.15: Illustration of a false negative image from the patient test data (the images are cropped out from the original)

	Spiked Data	Patient Data
SVM linear (Spiked)	16.59%	37.70%
SVM polynomial (Spiked)	20.85 %	55.27%
SVM rbf (Spiked)	18.48%	51.28%
Random forest (Spiked)	14.69%	39.09%
SVM linear (Patient)	-	30.03%
SVM polynomial (Patient)	-	26.71%
SVM rbf (Patient)	-	30.99%
Random forest (Patient)	-	29.71%

Table 6.8: Percentage of the true negatives that are actually scored as true negatives.

6.5 Time Elapsed

In table 6.9 the time elapsed for running the different algorithm are given. The specifics for the computer used for the scoring can be seen in table 6.10.

From table 6.9 it is clear that there are no remarkable differences between the algorithms in terms of time.

	Spiked Data (400 hot spots)	Patient Data (652 hot spots)
SVM linear (Spiked)	325.57s	608.13s
SVM polynomial (Spiked)	326.61s	605.61s
SVM rbf (Spiked)	325.35s	611.33s
Random forest (Spiked)	333.77s	607.50s
SVM linear (Patient)	-	602.65s
SVM polynomial (Patient)	-	612.53s
SVM rbf (Patient)	-	610.43
Random forest (Patient)	-	612.65

Table 6.9: Time elapsed for the different scoring algorithms.

Processor	Intel core i5-2500k 3.3 GHz
Memory	8 GB RAM
Storage	128 GB SSD

Table 6.10: Computer used for computing the results.

Chapter 7

Discussion

Thirteen different features are tested in this thesis. From the test of feature importance it is clear that not all features have the same influence on the separation of the data. The feature with the lowest importance is the mean intensity in CD45 for both the patient and the spiked data. There are big variations in the quality of especially the CD45 images, which could give rise to this low importance. It is also interesting to notice that the features with the highest importance are different for the patient data and the spiked data. Furthermore the features have roughly the same importance for the patient data compared with the spiked data. This could be due to the fact that there are big variations in the appearances of the CTCs in the patient data compared with the spiked data, as is illustrated in section 4.2.

For the random forest the oob-error is computed for varying number of trees. From this it is clear that the error is significantly lower for the spiked data compared with the patient data. This could probably be explained by the big variations between the CTC images in the patient data. Another reason for the lower error in the spiked data, could be that more positive CTC images were used for the training of the spiked data compared with the patient data. It is also interesting to notice that the number of features does not have a significant impact for the spiked data, whereas for the patient data the number of features seem to have a great impact. An explanation for this could be that the importance of the features are relatively similar for the patient data and

thus all features have an influence on the separation of the two classes.

From the cross validation of the random forest relatively high sensitivities and specificities are achieved. These are generally higher for the spiked data, as expected. By changing the number of variables to sample and the minimum number of leaf nodes, there are observed no big variations. The biggest variations are observed by changing the number of features used. It is not the same variables or the same number of features that give the highest sensitivities and specificities for patient data compared with spiked data. This is not surprising when comparing with the results from the measures of feature importance and the oob-errors.

The cross validation on the SVM with a linear kernel gives very different results depending on the variables used. Furthermore as for the random forest the sensitivities are generally higher than the specificities. The variables that give the highest sensitivities also gives the lowest specificities and a compromise between the two must be chosen. For the spiked data it is possible to choose a combination with high sensitivity that gives relatively high specificity. This combination gives a sensitivity that is almost equal to the one found using random forest, but the specificity is significantly lower. For the patient data both the sensitivity and specificity are significantly lower compared with the results from random forest. This indicates that the random forest performs better in the cross validation compared with the SVM with a linear kernel.

Cross validation is also performed using SVM with a polynomial kernel. The best combination of variables gives sensitivities comparable to the ones obtained using the linear kernel and the specificities are a bit higher. This goes for both the patient and spiked data. This could indicate that the polynomial kernel generally performs better compared with the linear kernel. On the other hand the specificities are again lower compared with the random forest, indicating that the random forest performs better on these data sets.

By cross validating the SVM with an rbf kernel it is clear that there are more variable combinations resulting in all the data being classified as CTC positive when comparing with the other algorithms. The best variable combination for the patient data results in the lowest sensitivity compared with the other algorithms. On the other hand the specificity is comparable to the one computed using SVM with a polynomial kernel. For the spiked data the specificity is also comparable with the one computed from using SVM with a polynomial kernel. The sensitivity computed for the spiked data when using SVM with an rbf kernel is the highest sensitivity computed compared with the other algorithms. Since sensitivity is weighted higher than specificity, this could indicate that using an rbf kernel is the optimal method for the spiked data, but it should be noted that the sensitivity for the rbf kernel is only around 1% higher than the sensitivities

computed for the other algorithms.

The ROC curves for the spiked data are very similar, and hence comparable, which corresponds with the cross validation. An explanation of this could be that the two populations of the spiked data (negative and positive images) are relatively easy to separate, and hence making the classification easier and thereby similar performance of the different algorithms are achieved. For the patient data on the other hand there are bigger variations in the performance of the ROC curves. It clearly looks like the random forest performs the best, which corresponds with the results from the cross validation. Generally the ROC curves computed for the patient data are worse than the ones computed for the spiked data, as expected.

For both patient and spiked data the threshold becomes very low before all true positives are classified as positives. This gives rise to a lot of false positives, but since avoiding false negatives is more important this is accepted.

For the testing of the algorithms the threshold is change based on the ROC curves. By changing the threshold high sensitivities, and thereby low number of false negatives, are reached and for most cases false negatives are avoided completely. The downside to the high sensitivities are low specificities. As can be seen from table 6.6 and 6.7 the highest specificities are reached for the lowest sensitivities, as expected. For the cases where a sensitivity of 1 is achieved the specificity is generally higher for the spiked data compared with the patient data. This is again what is expected, since the spiked data is generally easier to score compared with the patient data.

One image set is false negative for the spiked data and this is from testing the SVM with a polynomial kernel. The image set can be seen in figure 6.12 and from the FITC image it looks like a cluster consisting of 2 or 3 CTCs, and the BLOB in the FITC image is thus relatively big and does not have a cell-like morphology compared with other CTC positive images. This could be the reason for this image set being scored negative.

For the patient data there are all in all 3 image sets which are scored false negative. The first one in figure 6.13 there are no apparent overlap between FITC and DAPI, which could lead to the wrongful scoring. This could be a result of either misalignment between the FITC and the DAPI images or the cytokeratin only being expressed on the side of the cell. In the second image set, figure 6.14, there are a very high signal from the CD45 image. The entire CD45 image is bright red, which could be the explanation for the wrongful classification. In the last false negative image set shown in figure 6.15 there are almost no signal in the FITC image and the object does not have a cell-like morphology.

Besides the above mentioned reasons for the misclassification, it is also possible that some of the images are wrongfully classified by the operator. The operators scores are qualitatively made and no ground truth exist, thus variability in the operators scoring is expected.

The algorithm that performed the best on the patient test data is the SVM with an rbf kernel trained on patient data. This is a bit surprising, since this was the algorithm that performed the worst in the cross validation. For the spiked data it was the SVM with an rbf kernel trained on spiked data that performed the best, this is on the other hand in correspondence with the cross validation.

The time elapsed for running the algorithms are given in table 6.9. From this it is clear that there are no significant time differences between the algorithms. The time elapsed for scoring the patient test set (652 hot spots) is around 10 minutes, which is a reasonable time considering the reduction in manual scoring time. The same goes for the spiked data where the time elapsed is around 5 minutes for 400 hot spots.

Since there are big variations in the data, it has not been possible to eliminate the cases of doubts, and a manual scoring is still necessary for these. Furthermore there are a significant amount of false positives, so the images scored positive should also be looked through and verified by the operator.

As mentioned earlier the cases of doubts contains more than one object. This is to avoid classifying the same image twice. Furthermore for some CTCs the cytokeratin is unevenly spread out through the cell and thus there can be several small green spots in the FITC image. For such cells the image could be classified as containing more than one object and thus classified as a case of doubt. By use of another way to handle images containing more than one object, less images might be classified as cases of doubts.

Another problem is the amount of false positive. When decreasing the amount of false negatives, the amount of false positives increases, and since it is really important to avoid false negatives in this case, the high amount of false positives have to be tolerated.

Optimizing this scoring algorithm are a difficult task. For such an optimization a large data set with a known ground truth should be used. This is not possible in this case, since no ground truth exists. Furthermore the data set used for this thesis is relatively small and just a few wrongful scorings in the training set can have a significant impact on the performance of the algorithm.

Chapter 8

Conclusion

The purpose of this thesis have been to make an automatic algorithm for scoring of cell images obtained using CytoTrack. For this it is crucial to avoid false negatives completely and the amount of false positives is less important.

The algorithm was developed based on 13 simple features and the importance of these was tested. For the patient data all features had similar importance, where on the other hand there were big variations in the importance of the features for the spiked data.

Both patient data and spiked data was cross validated and from this it is clear that lower error, and higher sensitivities and specificities, are achieved for the spiked data. These results corresponds with the expectations, since spiked data is generally easier to score compared with patient data.

In order to avoid false negatives the threshold has been lowered based on the ROC curves. The threshold was chosen so that the true positive rate was equal to 1. By changing this threshold high sensitivities were achieved and in several cases it was possible to completely avoid false negatives, but as a consequence of these high sensitivities, low specificities were computed.

Since there still were a lot of false positives and there were some image sets scored as cases of doubts, these should still be manually scored by a trained

operator. Around 15-20% of the true negatives are scored negative for the spiked data and around 25-30% for the patient data. This is a significant amount for large data sets and is in compliance with the objectives. Furthermore the time used for the scoring is around 10 minutes for a data set of 652 hot spots, which is a reasonable amount of time.

Eight different algorithms was tested on the patient data. Four of the algorithms were trained on spiked data and four were trained on patient data. The algorithms trained on the spiked data unfortunately gave a few false negatives, but higher specificities. The four trained on patient data gave a bit lower specificities and no image sets were scored false negative.

On the spiked data only four algorithms were tested and these were all trained on spiked data. For the spiked data three out of the four algorithms performed similar, but one algorithm scored one false negative.

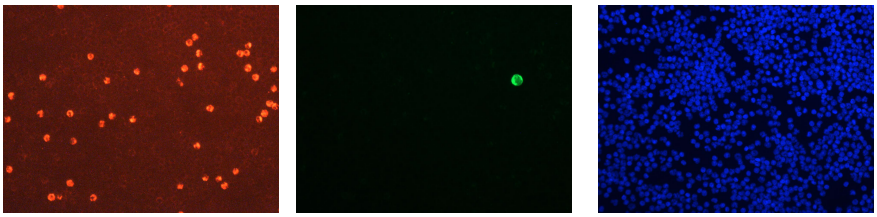
The algorithm that performed the best on the patient data was the SVM with an rbf kernel trained on patient data. Using this algorithm gave a sensitivity of 1 and a specificity of 0.4629 when testing on an unknown test set. Furthermore 30.99% of the true negatives was scored negative, which is in correspondence with the objectives.

For the spiked data, the algorithm that performed the best was SVM with an rbf kernel trained on spiked data. Using this algorithm gave a sensitivity of 1 and a specificity of 0.6724 when testing on an unknown test set. Furthermore 18.48% of the true negatives was scored negative, which is in correspondence with the objectives.

Appendix A

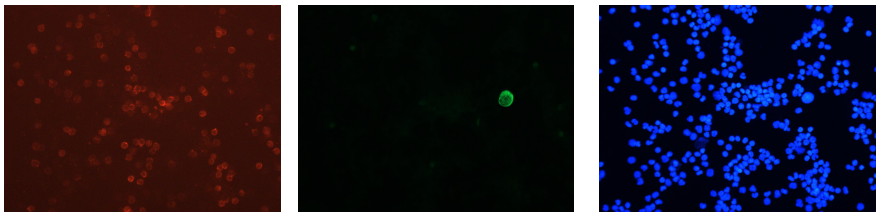
CytoTrack Images

In this section the uncropped versions of the cropped images shown in section 4.1, 4.2 and 6.4 are shown.



(a) CD45 **(b)** FITC **(c)** DAPI

Figure A.1: MCF7 cell imaged using CytoTrack (the original uncropped version of 4.1)



(a) CD45 (b) FITC (c) DAPI

Figure A.2: SkBr3 cell imaged using CytoTrack (the original uncropped version of 4.2)

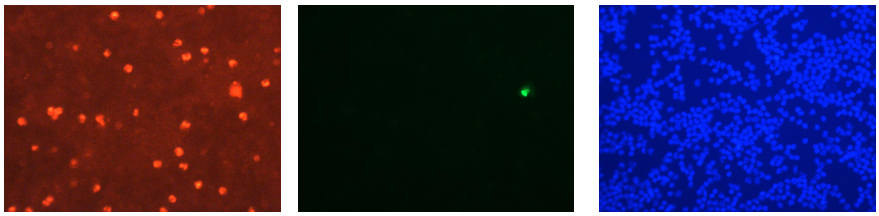


Figure A.3: Examples of false positive images from the spiked samples (the original uncropped version of 4.3)

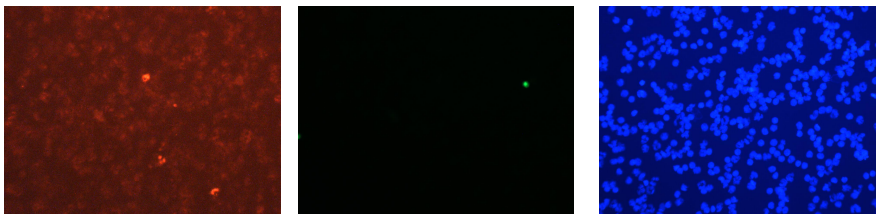


Figure A.4: Examples of false positive images from the spiked samples (the original uncropped version of 4.4)

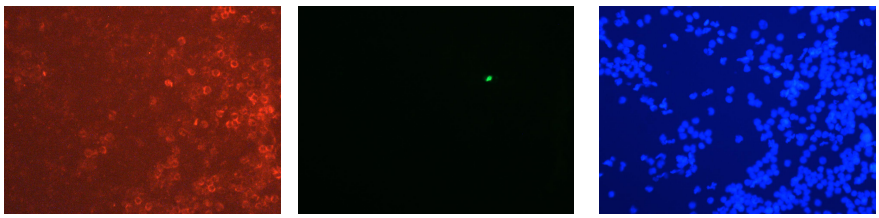


Figure A.5: Image examples of patient samples (the original uncropped version of 4.5)

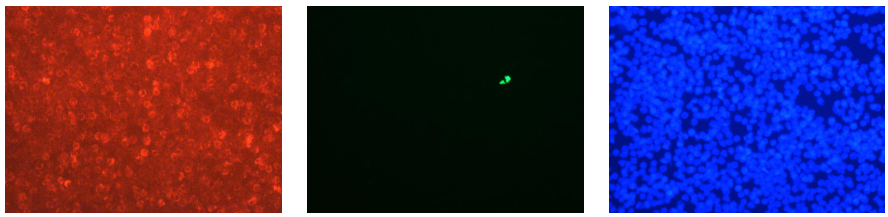


Figure A.6: Image examples of patient samples (the original uncropped version of 4.6)

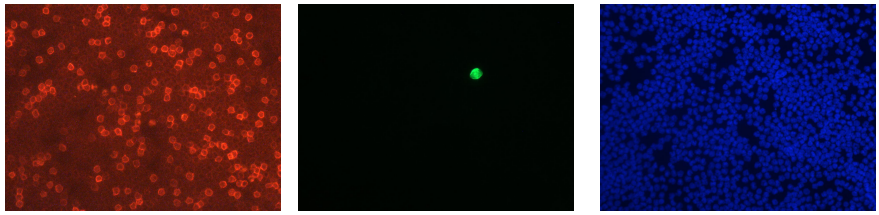


Figure A.7: Image example of a CTC in the B171 data set (the original uncropped version of 4.7)

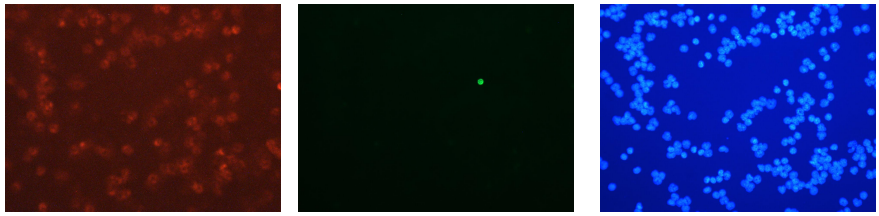


Figure A.8: Image example of a CTC in the B171 data set (the original uncropped version of 4.8)

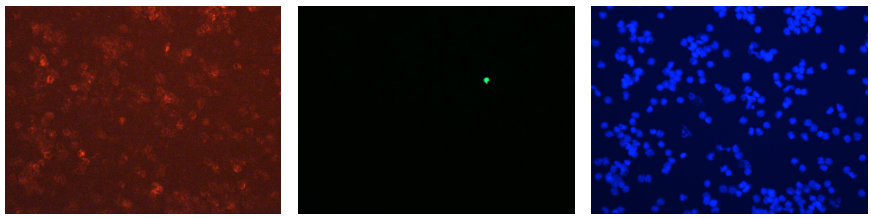


Figure A.9: Examples of false positive images from the patient samples (the original uncropped version of 4.9)

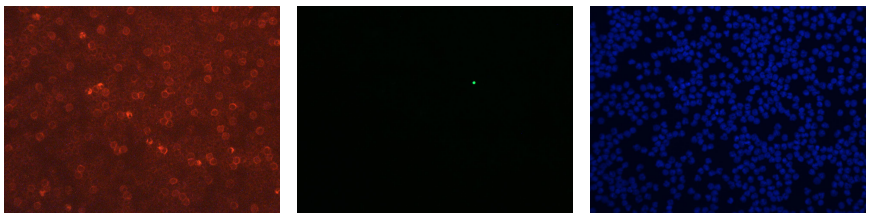


Figure A.10: Examples of false positive images from the patient samples (the original uncropped version of 4.10)

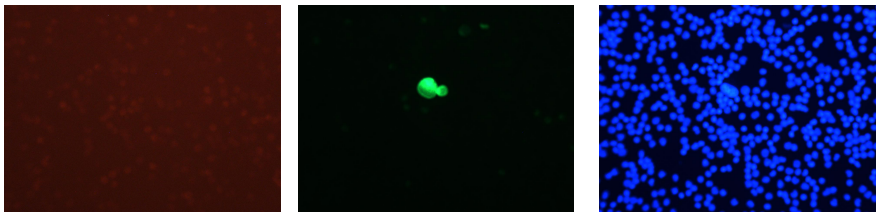


Figure A.11: Illustration of the false negative image from the spiked test data (the original uncropped version of 6.12)

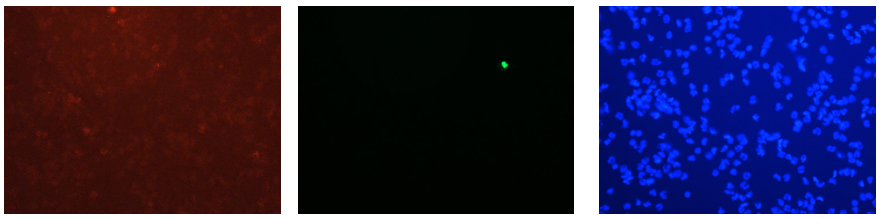


Figure A.12: Illustration of a false negative image from the patient test data (the original uncropped version of 6.13)

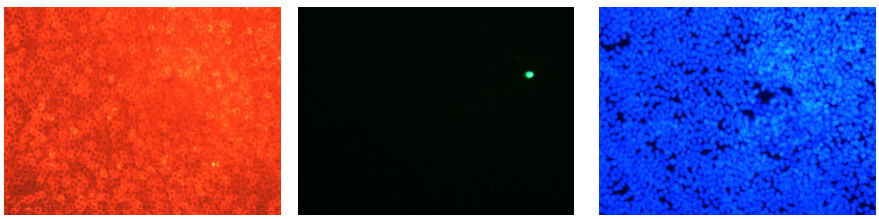


Figure A.13: Illustration of a false negative image from the patient test data (the original uncropped version of 6.14)

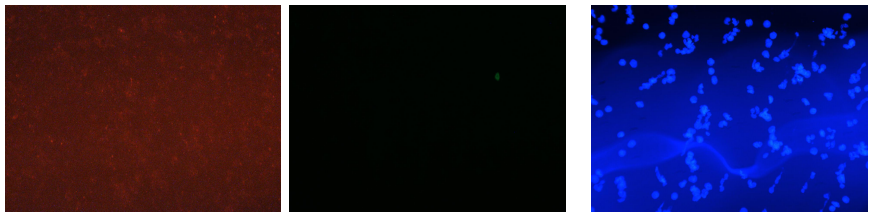


Figure A.14: Illustration of a false negative image from the patient test data (the original uncropped version of 6.15)

Appendix B

Feature Histograms

In this section the cumulative histograms for all the computed features are shown. In figure B.1, B.2 and B.3 the features for the patient samples are shown and in figure B.4, B.5 and B.6 the features for the spiked samples are shown.

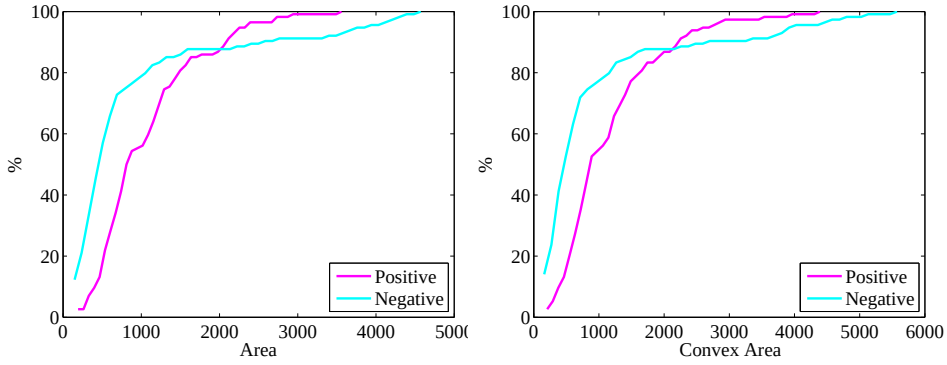


Figure B.1: Histograms of the area and the convex area for the patient samples.

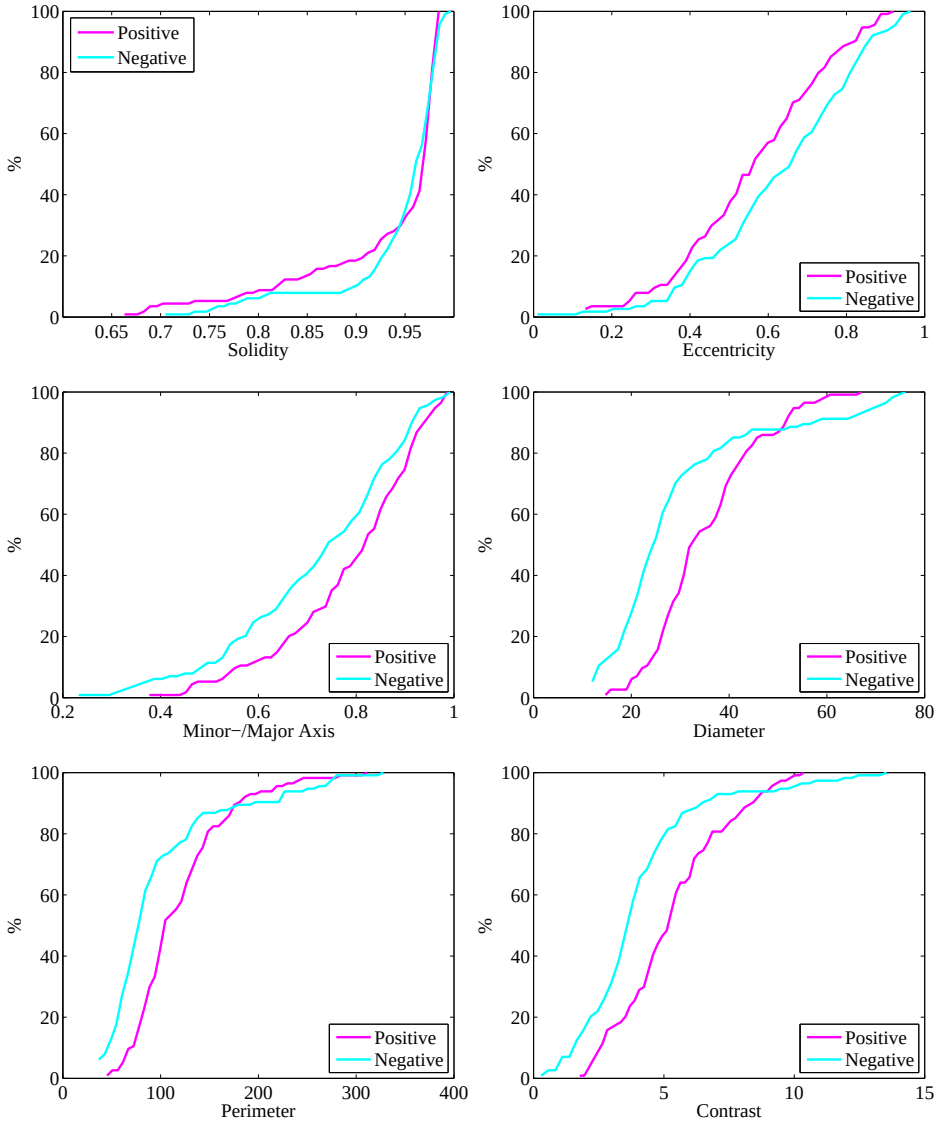


Figure B.2: Histograms of solidity, eccentricity, minor-/major axis, diameter, perimeter and contrast for the patient samples.

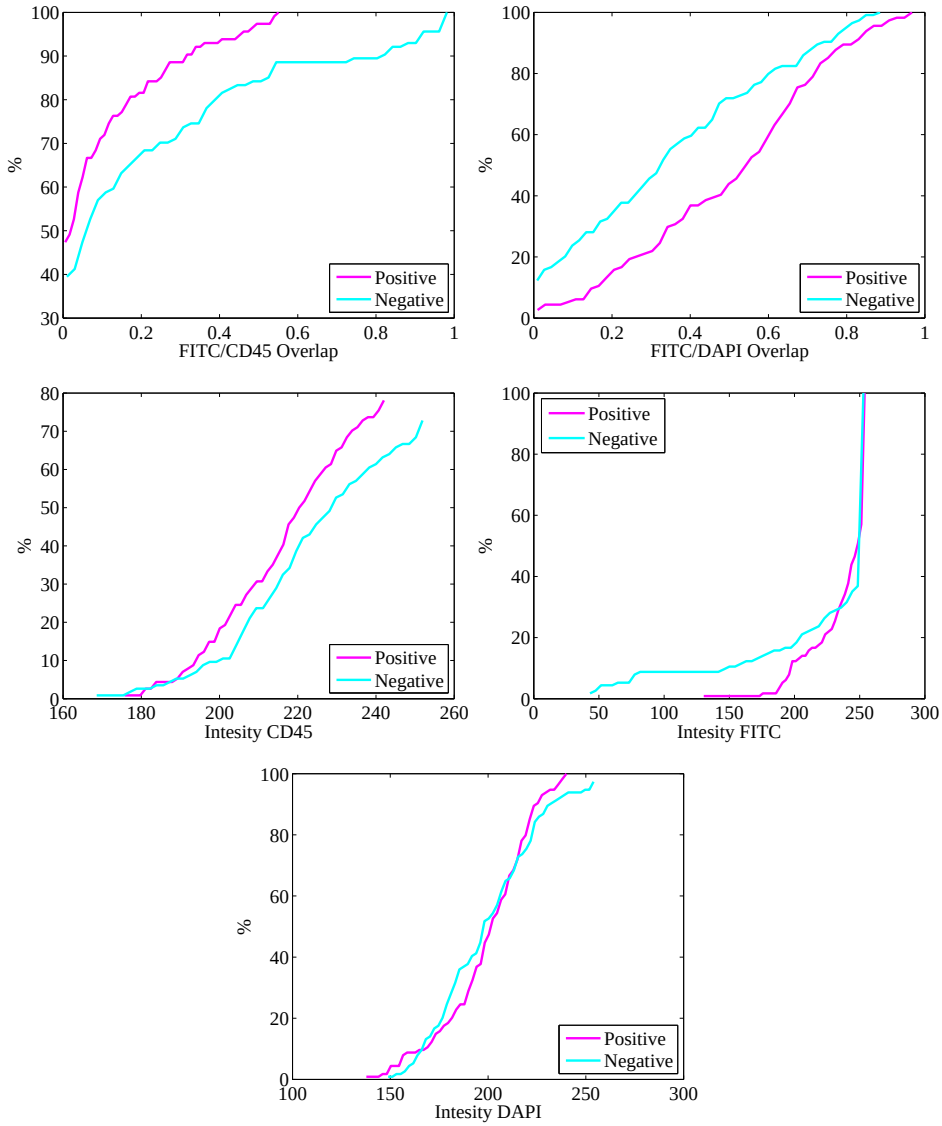


Figure B.3: Histograms of the FITC/CD45 and FITC/DAPI overlap, and of the intensities measured in CD45, FITC and DAPI for the patient samples.

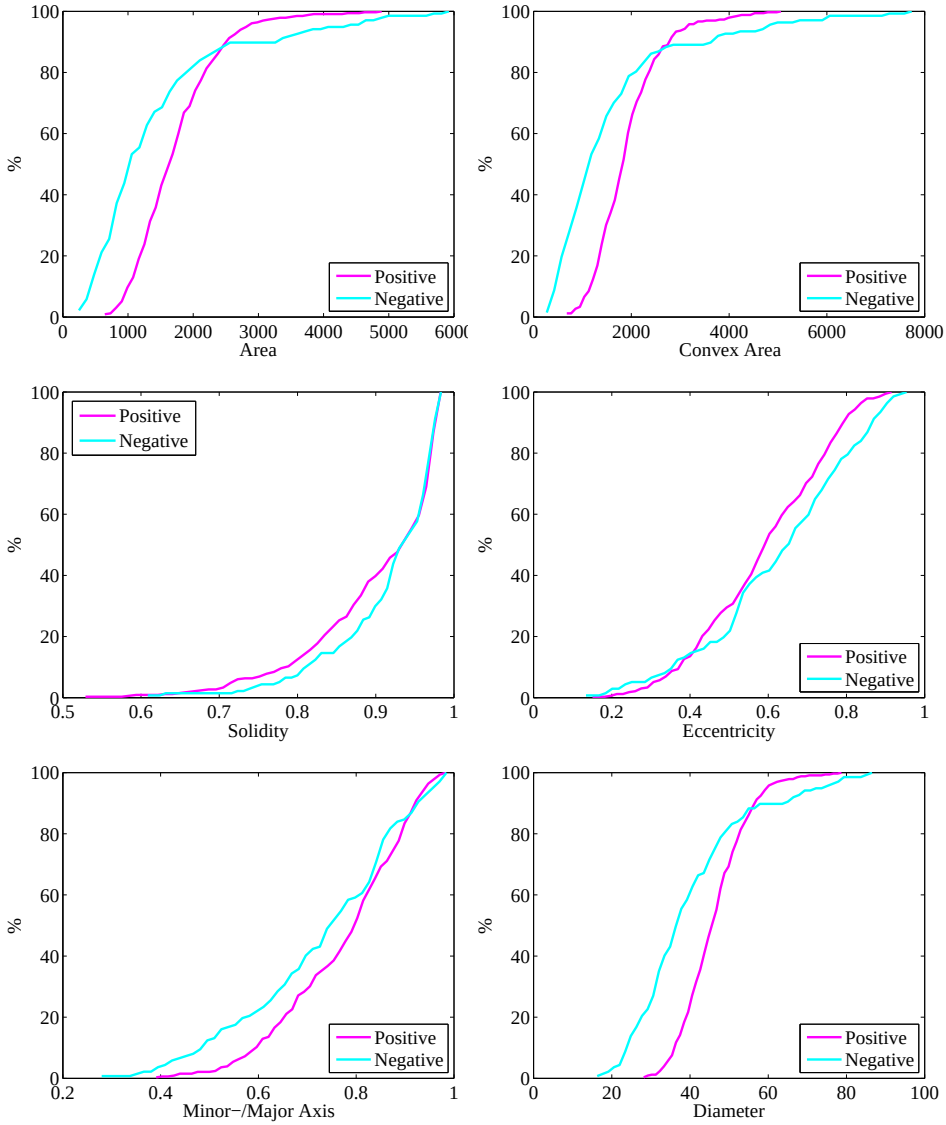


Figure B.4: Histograms of area, convex area, solidity, eccentricity, minor-/major axis and diameter for the spiked samples.

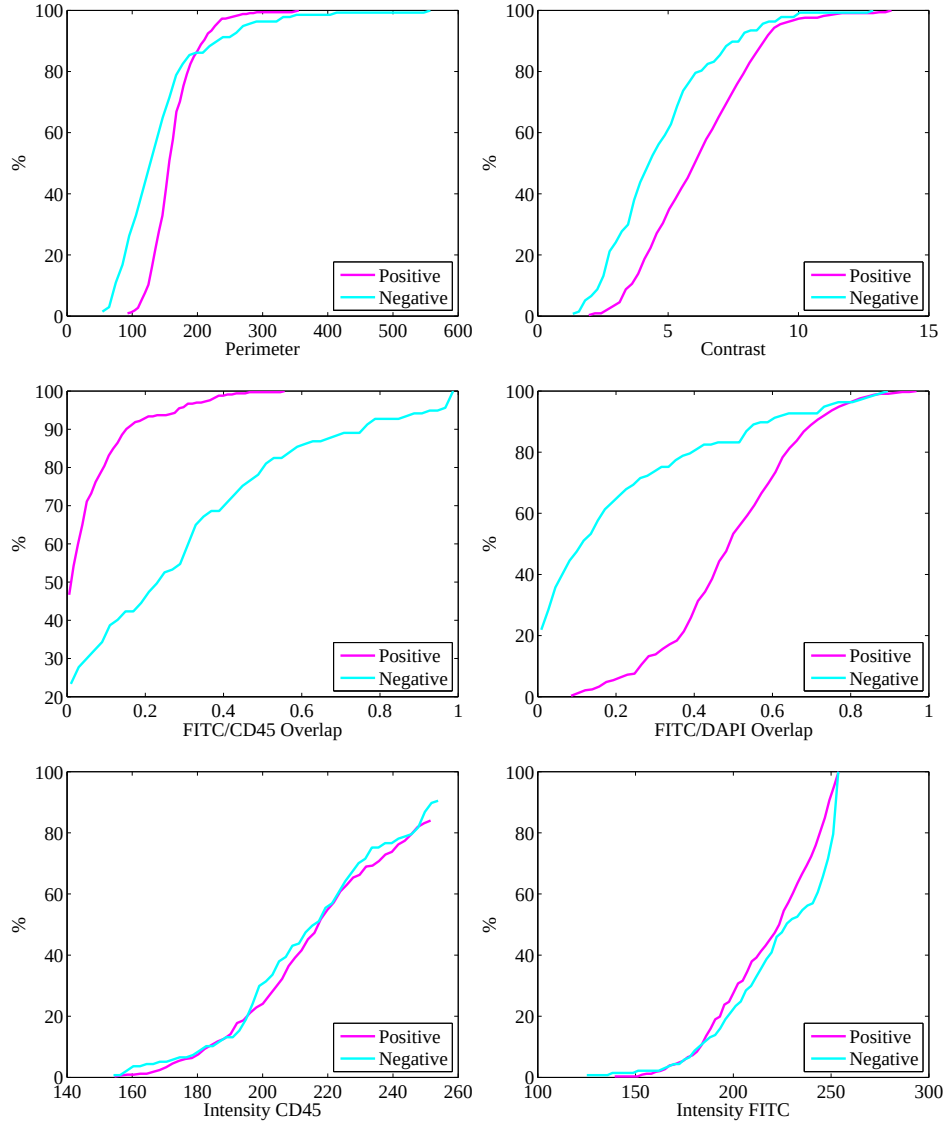


Figure B.5: Histograms of the perimeter, contrast, FITC/CD45 and FITC/DAPI overlap, and the intensities of CD45 and FITC for the spiked samples.

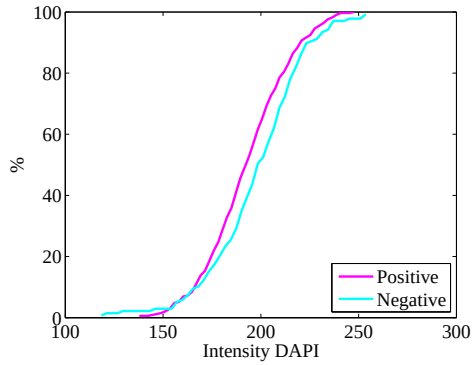


Figure B.6: Histograms of the intensities measured in DAPI for the spiked samples.

Appendix C

Specifications

Overview

In this section the specifications for the software developed as a part of the master project *Image based characterization of circulating tumor cells* are described. The purpose of the software is to allow users of the CytoTrack to make automatic scoring of cells. The images obtained from the CytoTrack can be used as input for the software, and the output will be a catalog of cells divided into three categories. The three categories should be: not a CTC, case of doubt, CTC. There will be a minimum of three input images; FITC, DAPI and CD45.

The scoring of the cells should be based on different parameters, e.g. eccentricity, diameter, intensity, contrast and size of nucleus.

Preprocessing: The input images are RGB images and these should be converted to gray scale and normalized, in order to make the images comparable. The normalized images should be segmented. By segmenting the images it is possible to extract objects from the images for further investigation. If necessary different filters and morphological operations may be applied.

Region of Interest: A region of interest (ROI) containing the hot spot may be cropped out of the binary FITC images. If the binary conversion of the DAPI

and CD45 images is not sufficient, another segmentation method may be applied on the DAPI and CD45 images in order to separate the individual nuclei/cells.

Categorization: Different methods for the categorization should be tested to find the optimal one. These methods should preferably include random forest and support vector machines.

Bibliography

[ATC14a] ATCC. Mcf7 (atcc®htb-22™). http://www.lgcstandards-atcc.org/Products/All/HTB-22.aspx, 2014.

[ATC14b] ATCC. Sk-br-3 [skbr3] (atcc®htb-30™. http://www.lgcstandards-atcc.org/Products/All/HTB-30.aspx, 2014.

[BACP07] Chris Bakal, John Aach, George Church, and Norbert Perrimon. Quantitative morphological signatures define local signaling networks regulating cell morphology. *SCIENCE*, 316(5832):1753–1756, 2007.

[BC] Leo Breiman and Adele Cutler. Random forests. http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm.

[BS09] Leo Breiman and E. Schapire. Random forests. 2009.

[C+04] M. Cristofanilli et al. Circulating tumor cells, disease progression, and survival in metastatic breast cancer. *The New England Journal of Medicine*, 351(8):781–791, August 2004.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995.

[Cyt14a] CytoTrack. *CTC Enumeration Test*, 2014.

[Cyt14b] CytoTrack. Cytotrack. http://www.cytotrack.dk, 2014.

[H+13] T. Hillig et al. *In vitro* validation of an ultra-sensitive scanning fluorescnce microscope for analysis of circulating tumor cells. *APMIS*, pages 1–6, August 2013.

[HEH+06] Michael Held, Holger Erfle, Nathalie Harder, Vassili Kovalev, Beate Neumann, Roland Eils, Urban Liebel, Jan Ellenberg, and Karl Rohr. Feature selection for evaluating fluorescence microscopy images in genome-wide cell screens. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:276–283, 2006.

[HGFO14] H.Hamilton, E. Gurak, L. Findlater, and W. Olive. Overview of decision trees. `http://dms.irb.hr/tutorial/tut_dtrees.php`, 2014.

[JBC+11] Ahmedin Jemal, Freddie Bray, Melissa M. Center, Jacques Ferlay, Elizabeth Ward, and David Forman. Global cancer statistics. *CA: A Cancer Journal for Clinicians*, 61(2):69–90, 2011.

[Lan13] Sven Langkjer. Phase ii trial with metronomic, capecitabine plus oral vinorelbine for metastatic breast cancer (xena). `http://clinicaltrials.gov/ct2/show/study/NCT01941771?term=XeNa&rank=1`, 2013.

[Lig12] MSc.ir. S.T. Ligthart. *Redefining circulating tumor cells by image processing*. PhD thesis, Enschede, May 2012.

[NH04] Thomas J. Nowak and A. Gordon Handford. *Pathophysiology*. McGraw Hill, third edition, 2004.

[Ots79] A threshold selection method from gray-level histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(1):62–66, Jan 1979.

[Pla99] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

[PSF+04] ZE Perlman, MD Slack, Y. Feng, TJ Mitchison, LF Wu, and SJ Altschuler. Multidimensional drug profiling by automated microscopy. *SCIENCE*, 306(5699):1194–1198, 2004.

[TM14] Inc The MathWorks. Support vector machines (svm). `http://www.mathworks.se/help/stats/support-vector-machines-svm.html`, 2014.

[WW07] L.V. Wang and H. Wu. *Biomedical Optics: Principles and Imaging*. Wiley, 2007.

[Y+11] M. Yu et al. Circulating tumor cells: approaches to isolation and characterization. *The Journal of Cell Biology*, 192(3):373–382, 2011.

[YFF08] H.D. Young, R.A. Freedman, and L. Ford. *Sears and Zemansky's University Physics*. Addison-Wesley, 2008.