1. Introduction

Cancer is the name given to a set of diseases characterized by uncontrolled growth and spread of abnormal cells. This continuous and unrestrained cell division can result in death of the patient [33]. In high-income countries, it is the second leading cause of death and the third in low and middle income countries. Cancer is responsible for more deaths than AIDS, tuberculosis and malaria combined. One in seven deaths worldwide is due to cancer [2]. Worldwide, lung, bronchus and trachea are the leading cause of cancer caused death amongst males, followed by liver. On the other hand, in females is breast then lung, bronchus and trachea cancer [2]. Besides the enormous impact cancer has on the number of people it affects, it also represents an immense economic burden. In 2010, the 13.3 million new cases of cancer estimated to cost the world US$290 billion, being approximately 53% medical costs, 24% of income losses and the remaining on non medical expenses. It is expected that in 2030 this value will rise to US$458 accounting for 21.5 million new cases of cancer [5, 7].

Cells from a primary tumour can detach and travel through the circulatory or lymphatic systems and are, therefore, called circulating tumour cells (CTCs). These cells can generate new colonies in sites far from where the first tumour was located, designated as metastasis [43]. In several tumours, this process has already occurred when the primary tumour is detected [18] leading to a high rate of compromised treatments. Approximately 90% of the deaths in cancer patients are due to metastasis [40, 41, 42]. The presence of CTCs in metastatic cancer patients is associated with poor survival prospects. Improvements in treatment and progress in early-stage diagnosis can reflect on higher survival rates. However the increasing number of treatment options (chemotherapy, radiation therapy, surgery, targeted therapy, immunotherapy, etc), raised the need for methods that determine if the intended therapy is being effective [34]. Several studies have disclosed that a change in the CTC count could be an indicator of treatment effectiveness [22]. The study of circulating tumor cells can provide game-changing methods to guide personalized therapies, increasing the survival rate of patients [14].

This project addresses the problem of automated identification of CTCs of Small-Cell Lung Cancer
proximately 5·1 CTC per mL of blood and it is surrounded by patients with metastatic cancer. There is approximately additionally CTCs are very rare cells in blood. In patients can be of 4% to 31% (median 14%) [22]. Ad-minimum. Inter-reviewer variability in CTC enumera-
tion: PCR-based analysis and cytometry based. The CellSearch System is the only FDA approved system for CTC enumeration, there is also the possibility of using different markers. The replacement of cytokeratin antibodies with other staining reagents, that target certain molecules, allows a better assessment of specific CTC, for example: staining reagent for Her-2 for breast cancer, Bel-2 for non-small lung cancer and non-Hodgkin’s lymphomas and/or AR for castration resistant prostate cancer [32].

In light of image processing and feature extraction, there are several focus points to be considered. First, the selection of the analysis area: when processing images from a cartridge, several of them have the edge of the cartridge that should be removed. Up to date, two solutions have been proposed for detection and removal of the sample border, via thresholding the FITC channel (the fourth channel that it is not used as a marker), which is a necessary step to get the true imaging area [22]. Second, if the images are retrieved with different machines, under different light conditions or present too much noise, there might be the need for image normalization. The Naka-Rushton filter was introduced in analysis of circulating tumor cells, by Svensson, et al. [34], for enhancement of foreground objects and suppression of background noise. The use of top-hat background subtraction algorithms can lead to both the presence of negative values and formation of extra contrast, thus the proper background subtraction method would be as follows: recording a black image with no objects present, followed by subtracting this black image from images with objects, however most of the times this image is not available [22]. Following edge removal, Svensson, et al. [34] also implement a Gaussian blurring filter for image smoothing. To locate objects and its outline, there is the need to implement segmentation techniques. These can be divided into two classes: contour-based and region-based. Contour-based techniques require edge enhancement steps to find the contour or edges of objects. Region-based can be of texture analysis, watersh edding or intensity thresholding (local or global). Svensson, et al. [34] applied the watershed algorithm to the DNA channel followed by the use of random forest to decide whether or not the ROI should be considered a candidate for further classification. Lighthart, et al. [22] implemented several algorithms, such as the Zack’s triangle threshold via channel image histogram, the Otsu’s threshold and isodata algorithms, for image segmentation in the study of CTCs. Finally, in order to analyse each cell, several different features have been studied, such as color histograms [34] and quantitative correlation [27], texture [27, 22, 21] and morphological features [27, 22, 21].
In order for the CTC enumeration system to be automatic, there is the need to have some kind of classification, which can be solved by the implementation of machine learning algorithms. Before stepping into the actual classification of each cell into CTC or non-CTC, Svensson, et al. [34] proposed the implementation of a Random Forest classifier to identify relevant ROI and, only after this, proceed to the classification itself. In this step, the features used were area and perimeter-to-area ratio. Up to date, several classification approaches have been presented for the automated classification of CTCs. The first classification method is not a machine learning implementation, it is based on numeric inclusion (example: if the size is within a certain range of values, peak intensity in the DAPI-DNA channel and standard deviation of CK-PE channel are bigger than specified thresholds and the peak intensity of the CD45-APC channel is smaller than a determined constant) [22, 21]. In recent years, more advanced techniques have been explored. Regarding generative models, both Naïve Bayes Classifiers [34] and Random Forests [27] have been successfully implemented. Support Vector Machines [34], a discriminative method, has also been studied for this problem and performed well. For performance evaluation, cross-validation is the algorithm used by Svensson, et al. [34]. When evaluating the performance of a classification algorithm applied to the identification of CTCs, it should be taken into consideration that the dataset is highly unbalanced due to the incredibly low number of CTCs when compared to the number of non-CTC in a sample. Therefore, accuracy might not be always the most informative measurement. Class imbalance addressed in this project that has not been addressed before.

1.2. Proposed Approach
Image data used in this thesis was provided by the Cancer ID project (http://www.cancer-id.eu/) team of University of Twente. This dataset consists of images from blood samples obtained from 59 patients with SCLC, the blood collection was done before, after one cycle and at the end of chemotherapy. Each blood sample corresponds to one cartridge, 175 four channel TIFFs, acquired with a fluorescent-based microscopy system CellTracks™ analyzer II, using a 10X NA 0.45 objective with filters for DAPI, PE, APC and FITC (biomarker not used for feature extraction). This dataset was previously described and analysed by Hiltermann, et. al [19]. In the present work, it is proposed, for CTC identification, a system that is composed by two main components: image processing and machine learning.

The image processing block contemplates a solution for edge removal, image normalization, ROI analysis and extraction of morphological features (area, eccentricity, perimeter, perimeter to area ratio), quantitative intensity related features (mean and maximum intensity, standard deviation of intensity signal, mass) and texture related features (local contrast, local entropy, histogram of oriented gradients). Before stepping into the classification part, outliers (example a ROI with an area too big to be considered a cell) were removed from the dataset. The machine learning block aims to compare the performance of four different classification algorithms: Support Vector Machine (SVM), k-Nearest Neighbor (k-NN), AdaBoost and RUSBoost. Parameter estimation was performed within a nested Cross Validation procedure. In this project we propose innovative methods for automated CTC enumeration in both of the main components (Image Processing and Classification) and also regarding the type of cancer. The automated identification of CTC present in previous works (using either CellSearch System or FSMW) was for breast [34, 21, 27], colorectal [27], non-small-cell lung cancer [34] and castration resistant prostate cancer [22]. Regarding feature extraction we present a new texture feature: histogram of oriented gradients. On classification, Support Vector Machines (along with Naïve Bayesian Classifiers) have been previously used on FSMW (breast and non-small cell lung cancer), with color histograms as features [34]. Using the CellSearch technology, the classifier studied was the Random Forests with morphological, texture, quantitative and correlation features, nevertheless the images were retrieved using a camera with improved resolution (Time Delay and Integration camera using a 40X 0.6 NA objective) [27]. Thus, the use of SVMs on CellSearch System and the use of k-NN, AdaBoost and RUSBoost algorithms introduce a new approach for the problem at hand. Additionally this project tries deal with the data imbalance, a problem that has not been address before in the context of CTC automated enumeration.

2. Methods
2.1. Image Processing
A) Image Normalization: An essential step, in order to quantitatively compare objects, is image normalization. This was performed the following way “all imported 8-bit multipage TIFF images were scaled from 0255 and had to be re-scaled to pseudo 12-bit using information stored in the TIFF-header” [22] (an offset and a maximum value related with IMMC/Veridex TIFF scaling) using the following equation: ImageToSegment = Offset + OriginalImage × MaximumValue − Offset max(OriginalImage) [22].

B) Edge Detection: Each dataset corresponds to one blood sample, therefore one cartridge (one scan), which corresponds 175 images. Part of these
images have present the cartridge border. For correct ROI segmentation it was necessary to detected the sample border and exclude the outside area from further analysis. This was accomplished via thresholding in the FITC channel. However cartridges have very irregular edges, especially at the corners, making necessary to compare the total selected area of the whole cartridge to a training set that was acquired manually [21].

C) Image Segmentation: Given the fact that every object present on the images is slightly visible above the background, a basic histogram-based thresholding algorithm is enough to segment the image. The algorithm chosen to perform this task was the Zack’s triangle threshold method, over the DNA channel. This geometric method assumes a maximum peak near one end of the histogram of pixel intensities and searches towards the other end. It was considered an object of interest a region of the image that has a higher intensity than the defined threshold. By adjusting the search threshold until the average brightness of the pixels contiguous to the segmented object was within a small fixed offset of the average background intensity, one can account for the variations in staining intensity [24]. In cases where the maximum is not near one of the histograms extremes, the algorithm searched for the threshold within the largest range.

D) Feature Extraction:

a) Morphological Features: area, perimeter, eccentricity, perimeter to area ratio.

b) Intensity Features: mean intensity, maximum intensity, standard deviation of intensity and mass of each channel.

c) Texture Features: The median was computed for the local contrast, local entropy and gradient amplitude due to the fact that each ROI had a different size and, for classification, each input vector was required to have the same size. The local contrast is the range value in a 3-by-3 neighborhood around the corresponding pixel in the input ROI. The local entropy measures the randomness of an image and it is computed as follows:

\[
e = - \sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i)
\]  

(1)

where \( z_i \) indicates the intensity, \( p(z) \) is the histogram of the intensity levels in a region and \( L \) is the number of possible intensity levels [17]. The gradient of an image represents a directional change in the intensity. The gradient amplitude encodes edges and local contrast of images. Using a sobel filter, first it is computed the directional gradients \( G_x \) and \( G_y \), with respect to each of the figure axis (x and y). The gradient magnitude and direction are then computed from their orthogonal components \( G_x \) and \( G_y \). Lastly, it was extracted for each ROI a 10-bins Histogram of Oriented Gradients (HOG). The idea behind HOG feature descriptor is that an object appearance and shape can be described by its distribution of intensity gradients or edge directions and it presents a certain degree of invariance to transformations or rotations. In order to compute the HOG an image is divided into small connected regions (cells), an histogram of gradient directions is then obtained for the pixels within each cell. The descriptor is the result of the concatenation of these histograms [13].

2.2. Classification and Performance Evaluation

In automated pattern recognition the system has to learn a model from the training instances and be capable of classifying future unseen data based on the previously formed model. Bellow it is presented the approaches implemented in this project:

A) \( k \)-NN: When a new point is presented to the \( k \)-NN, it attempts to find a predefined number of training samples closest to this new data point and predict its label by computing a majority voting. In this project, it was analysed the use of different number of neighbors, and, tackle the class imbalance problem, bootstrapping and the use of prior probabilities were tested. The theory behind NN is quite intuitive: consider an input space \( \mathbb{R}^d \), being \( d \) the dimension of the input space, patterns nearby that same input space most likely belong to the same class. The NN rule classifies a query pattern \( X \) to the class of its nearest neighbour in the training data \( D_n \), when given a set of examples \( D_n = (x_1, y_1), ..., (x_n, y_n) \), where \( x_i \in \mathbb{R}^d \) represents the input vectors \( x_i \in \mathbb{R}^d \) and \( y_i \) represent the class label. Without any prior knowledge, the Euclidean distance is typically used, which is defined as follows:

\[
d(X, x_q) = \left( \sum_{k=1}^{d} |X^k - x_q^k|^2 \right)^{\frac{1}{2}},
\]

(2)

where \( x_q \) is a new unclassified pattern. When prior probabilities are, added a weight is assigned to each class when computing the Euclidean distance. Bootstrap is a data-resampling strategy, this method has several applications [15]. In the light of this project, it was used as follows: the bootstrap resamples, with replacement, our dataset into smaller new datasets with a more balanced ratio of CTC vs. non-CTC. For each new dataset a \( k \)-\( NN \) model is generated which can be used to predict the class of a new pattern. In the end the class of a new pattern will be the result of mode of the predicted class given by each \( k \)-\( NN \) model.
B) SVM: Given a binary classification problem, the SVM algorithm searches the hyperplane that maximizes the distance to the support vectors (closest training vectors) of both classes [8, 12]. When no hyperplane is found on the input space, one can map the training set into a typically higher dimensional space (feature space) where data is linearly separable. Additionally, one can soften the constraints and allow for error, in which case the SVM will search for the hyperplane that minimizes the errors (soft margin). The last problem to take in consideration is the fact that we are dealing with an unbalanced dataset. To overcome this adversity, the use of different penalties has been proposed [37, 23]. Gathering the two extensions above (the use of kernels and soft-margins) along with this solution, we obtain the following SVM formulation for a binary classification problem:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} w^T w + C^+ \sum_{y_i=+1} \xi_k + C^- \sum_{y_i=-1} \xi_k \\
\text{subject to} & \quad y_k (w^T \phi(x_k) + b) \geq 1 - \xi_k \forall k \\
& \xi_k \geq 0 \forall k,
\end{align*}
\]

where \( w \) and \( b \) are the hyperplane coefficients, \( x_k \) and \( y_k \) are the feature vector and the class label associated with the \( k \)-th training sample, \( \phi(-) \) is the mapping function, \( \xi_k \) is the positive slack variable that accounts for the error committed in the classification of the \( k \)-th sample, \( C^+ = w_+ \times C \) and \( C^- = w_- \times C \), \( C \) is a tuning parameter that controls the cost of misclassification \( w_+ \) and \( w_- \) are the weights of the associated with the positive and negative classes, respectively. Two common kernels were tested in this study: the linear kernel, \( K(x_k, x_l) = x_k \cdot x_l \), and the RBF kernel, \( K(x_k, x_l) = \exp(-\gamma \|x_k - x_l\|^2) \). The SVM dual problem was solved numerically using LIBSVM, a publicly available software developed by Chang and Lin [10].

C) AdaBoost: The AdaBoost starts by fitting a classifier on the dataset, later it fits copies of the classifier on the same sample pattern, nevertheless the weights of incorrectly classified instances are adjusted so that in the following iterations the classifiers can focus in harder cases [26, 39]. Consider a training set \((x_1, y_1), \ldots, (x_t, y_t)\), where \( x_t \) represents a pattern in the input space \( X \) and \( y_t \) is the correspondent label. For the sake of simplicity, let us assume \( y_t \in Y = \{-1, +1\} \). The algorithm takes the training set as an example and calls a “weak” learner repeatedly in series of cycles \( t = 1, \ldots, T \). In the beginning all weights are set equally and at each round they are updated, in such way that the weak learner is forced to focus on the hard examples of the training set, i.e., incorrectly classified examples increase their weight.

The weight on the training pattern \( i \) on the round \( t \) is \( D_t(i) \). The weak learner has now to find a weak hypothesis \( h_t : X \rightarrow \{-1, +1\} \) adequate for the distribution \( D_t \). The quality of the hypothesis is measured by the error with respect to the distribution \( D_t \) on which the weak learner was trained, given by:

\[
\epsilon_t = Pr_{x \sim D_t} [h_t(x) \neq y_t] = \sum_{i: h_t(x_i) \neq y_i} D_t(i).
\]

After the weak hypothesis has been defined a parameter \( \alpha_t \) that measures the importance of \( h_t \). This step is followed by an update of the weight distribution so that the classifier focus on hard examples, meaning the weight of misclassified examples increases by \( \alpha_t \) and the weight of correctly classified examples decreases. In the end, a weighted majority vote of \( T \) weak hypothesis (being \( \alpha_t \) the weight assigned to hypothesis \( h_t \)), gives us the final hypothesis \( H \). In this project, the implementation of the AdaBoost.M1 used is the framework for Ensemble Learning of MATLAB R2014a, present in the Statistics and Machine Learning Toolbox.

D) RUSBoost: Proposed in 2010 by Seiffert, et al. [29], RUSBoost has its roots on the SMOTEBest algorithm. Both algorithms add a data sampling technique to the AdaBoost algorithm, the SMOTEBest uses an oversampling technique [11]. The RUSBoost algorithm uses an approach that has proved to be simple, fast and with good performance [36]. RUS (Random undersampling) simply randomly removes examples of the majority class until the desired distribution of classes is achieved. The full RUSBoost algorithm is explained in the form of pseudo-code in Figure 1. In this project, the implementation of the RUSBoost used is the framework for Ensemble Learning of MATLAB R2014a, present in the Statistics and Machine Learning Toolbox.

When developing an automated classification system, there is the need to assess the performance of the proposed classifiers. However, different performance metrics yield different meanings and trade-offs, one classifier can be optimal in one metric and suboptimal in another. The most commonly used metric is accuracy, however in this project metrics accuracy might be incredibly uninformative due to the fact that 99% of the dataset can be composed of non-CTCs. More adequate metrics and also widely used are balanced accuracy, sensitivity and specificity. Nonetheless, these require us to decide in which point of the Receiver-Operator Characteristic (ROC) we want to position ourselves to consider the classifier good. Therefore the main performance evaluation metrics used were the ROC curve and the Area Under the Curve (AUC).
Figure 1 RUSBoost [29, 28].

1: Given: Set $D$ of examples $(x_1, y_1), \ldots, (x_k, y_k)$ with minority class $y' \in Y$, $|Y| = 2$
2: Weak learner, WeakLearn
3: Number of iterations, $T$
4: Desired percentage of total instances to be represented by the minority class, $M$
5: Initialize $D_t (i) = \frac{1}{k}$ for all $i$.
6: for $t = 1, \ldots, T$ do
7: Create a temporary training dataset $S'_t$ with distribution $D'_t$ using random undersampling
8: Call WeakLearn, providing it with examples $S'_t$ and their weights $D'_t$
9: Get back a hypothesis $h_t : X \times Y \rightarrow \{0, 1\}$
10: Compute the pseudo-loss (for $S$ and $D_t$):
11: $\epsilon_t = \sum_{(x,y) \in S} D_t (i) (1 - h_t (x_i, y_i) + h_t (x_i, y'))$
12: Calculate the weight update parameter:
13: $\alpha_t = \frac{\epsilon_t}{\sum_{i} \epsilon_t}$
14: Update $D_t$:
15: $D_{t+1} (i) = D_t (i) \frac{\alpha_t}{\sum_{i} \alpha_t} (1 + h_t (x_i, y_i) - h_t (x_i, y' \neq y))$
16: Normalize $D_{t+1}$: Let $Z_t = \sum_{i} D_{t+1} (i)$
17: $D_{t+1} (i) = \frac{D_{t+1} (i)}{Z_t}$
18: Output the final hypothesis:
19: $H(x) = \text{argmax}_{y \in Y} \left( \sum_{t=1}^{T} h_t (x, y) \log \left( \frac{1}{Z_t} \right) \right)$.

A) Nested Cross-Validation: Performance evaluation of all classifiers was done using the nested CV procedure, originally proposed in [38]. Nested CV partitions the initial data into $k$ disjoint sets. Then, in each iteration, one set is left out (validation set), while the others (train set) enter in several CV procedures (inner CV, with $k'$ partitions), one for each parameter setting, from which the best parameters are chosen. Then, the samples present in the train set are used to build a model with the chosen parameters and tested with the validation set. For all partitions to be used for test, this procedure is repeated $k$ times. Measures of performance, such as sensitivity, specificity and ROC curves are then computed based on the true class labels and on the classification of each sample obtained when it was part of the test set. The nested CV provides not only unbiased estimates of performance metrics, but also allow for model parameters tuning.

B) ROC: In order to evaluate the results regarding the performance of the classifiers described above, Receiver Operator Characteristic graphs will be used. The area under the curve of a ROC has a baseline rate that is independent of the data, while in some other metrics it is data dependent [9]. Fawcett and Provost [16] did a thorough study on the use of ROC for evaluation of classifiers. The true positive rate ($TPR = p(Y | +) \approx \frac{positives correctly classified}{total positives}$) and the false positive rate ($FPR = p(Y | -) \approx \frac{negatives incorrectly classified}{total negatives}$), where $\text{+}$ and $\text{-}$ are the positive and negative instance classes, respectively. The $p(\text{+} | x_i)$ is the posteriori probability of the instance $x_i$ being positive. A ROC curve plots the TPR on the Y axis and FPR on the X axis, bringing the advantage of presenting the behaviour of a classifier regardless of class distribution or error cost. In order to choose the best classifier based on a ROC curve analysis one must maximize $(1 - FPR) \cdot TPR$, which corresponds to selecting the classifier with the higher area under the curve (AUC). This approach calculates the average performance of the classifier over the entire performance space [35, 6].

3. Results
3.1. Dataset - Cell Search Images
The fluorescent microscopy images used on the development of this thesis were provided by the Cancer-ID consortium. This was a multicenter study consisting of 59 patients with Small-Cell Lung cancer. From each patient it was retrieved three blood samples, one before (designated as baseline) and one after a cycle of chemotherapy and one at the end of chemotherapy. Each blood withdrawal corresponds to one cartridge and each cartridge corresponds to 175 4-channel TIFF images. The images were obtained using the CellSearch System and manually classified by expert reviewers. To the blood samples it was added ferrofluids with EpCAM (epithelial cell adhesion molecule) in order to select cells of epithelial origin, and they were stained by DAPI-DNA (4', 2-diamidino-2-phenylindole, dihydrochloride) for nuclear stain, PE-CK (cytokeratin 8, 18 Phycoerythrin and cytokeratin 19 Phycoerythrin) and CD45-APC (CD45-allophycocyan) to label leukocytes. The objective of the microscope was of 10x/0.45NA and it had filters for DAPI, CK, CD45 and FITC, respectively each of the 4 channels of the 4-page TIFF. The FITC channel was used only for removal of the edge of the cartridge. Each cartridge was classified by an expert reviewer. Regarding non-CTCs there is no information, in case of presence of CTCs, it is registered that within that area (a square) there is a CTC. Along with each cartridge set (175 images), there is an XML file with the position of each CTC relative to the whole carriage which is then transformed in relation to the image that is being analysed. Additionally, the TIFF header of each imaged has two values correspondent to an offset and a maximum value related to the condition in which the image was obtained, this information was used for image normalization. Due to limited computation capacities, three random datasets, from all the available, were used for testing. In total 525 images were processed and 141
634 ROIs were classified, 18 822 of which were Circulating Tumor Cells.

3.2. Experimental Design

The goal of this project is to access the viability of an automated classification system for identification of circulating tumor cells. With this purpose in mind, several classifiers were tested, namely k-NN, k-NN along with bootstrapping, k-NN using prior probabilities, SVM with a linear kernel, SVM with a RBF kernel, AdaBoost and RUSBoost. Along with testing several classifiers it was also analysed which set of features yielded better results. Thus each of the algorithms was tested for the set of features designated as All, Morphological, User (features related to the ones expert reviewers usually take into account when performing manual classification), Intensity, Texture, DNA (intensity and texture features of this channel), CK (intensity and texture features of this channel), CD45 (intensity and texture features of this channel), Table 1. All the algorithms were first tested with one dataset (one cartridge), followed by a second test using another cartridge and finally the concatenation of three datasets. The parameters $C$, both for linear kernel and RBF kernel, $\gamma$ for RBF kernel, $k$ (number of neighbors) for $k$-NN and the number of weak learners used in both of the ensemble methods were estimated using nested cross-validation. Other algorithm specifications are presented below:

- **k-NN** - trained assuming $k \in \{1, 3, 5, 7, 9\}$.
- **k-NN with bootstrapping** - $k = 3$ and bootstrapping was performed in such way that each set had there was the same amount of CTCs and non-CTCs.
- **k-NN with Prior Probabilities** - trained with the following sets of pairs of prior probabilities $\{(0.50, 0.50); (0.60, 0.40); (0.75, 0.25); (0.85, 0.15); (0.95, 0.05); (0.99, 0.01); (0.995, 0.005); (0.95, 0.05); (0.35, 0.65); (0.30, 0.70); (0.10, 0.90); (0.01, 0.99)\}$.
- **SVM Kernel** - Linear and RBF kernels were tested with weights $w_0 = 1$ and $w_1 = \frac{\#\text{non-CTCs}}{\#\text{CTCs}}$, corresponding to the weight of classes non-CTC and CTC respectively. The $C$ was assumed to have the following values $\{2^{-16}, 2^{-14}, 2^{-12}, 2^{-10}, 2^{-8}, 2^{-6}, 2^{-4}, 2^{-2}, 2^{0}, 2^{2}, 2^{4}\}$ and $\gamma \in \{2^{-18}, 2^{-14}, 2^{-10}, 2^{-6}, 2^{-2}, 2^{2}, 2^{6}, 2^{10}\}$.

- **Ensemble methods** - Both AdaBoost and RUSBoost were tested using using as weak classifier trees and the number of trees was $\{100, 200, 300, 400, 500, 600, 700, 800, 1000\}$.
- **Nested Cross-Validation** - The outer fold had 10 folds, and the inner 7.

3.3. Discussion on Image Processing

Each image was normalized. After this step the average background intensities of the images had a negligible difference between each other. Features are extracted by segmenting image by image, from a full cartridge (175 images). Take into consideration Figure 2, where an example of a CTC (Figure 2(a)) is presented side by side with a non-CTC example (Figure 2(b)), there is no obvious visual difference between a CTC and a non-CTC in either of the channels. If you observe the full dataset, ROI by ROI, it is hard to find, by visual inspection, a clear and obvious pattern that allows a non expert reviewer to clearly distinguish a CTC from a non-CTC. The segmentation of these two cells was correctly accomplished. However, there are some situations in which one can not be sure of the quality of the segmentation. If you consider Figure 3(a) it is not possible to know, at least for a non-expert reviewer, if we are dealing with a cluster of two cells, and, in that case the segmentation performs poorly, or if it just one big cell. Now consider the situation presented in Figure 3(b), we have an example of a situation where the segmentation algorithm does perform up to the expectation. On the left we have a cell, on the right an element which is not clear, it can be a smudge, debris or an apoptotic cell. Nonetheless the segmentation algorithm was not able to separate the two objects. Overall most of the objects were correctly segmented, however for the problems exemplified above no solution was im-

<table>
<thead>
<tr>
<th>Table 1: Set of features of each category used for classification.</th>
</tr>
</thead>
<tbody>
<tr>
<td>All-Measured Features</td>
</tr>
<tr>
<td>-----------------------</td>
</tr>
<tr>
<td>Area</td>
</tr>
<tr>
<td>Eccentricity</td>
</tr>
<tr>
<td>Perimeter</td>
</tr>
<tr>
<td>Perimeter to Area ratio</td>
</tr>
<tr>
<td>Mean Intensity CD45</td>
</tr>
<tr>
<td>Max. Intensity CD45</td>
</tr>
<tr>
<td>Standard Deviation Int. CD45</td>
</tr>
<tr>
<td>Mass,CD45</td>
</tr>
<tr>
<td>Median of Local Entropy DNA</td>
</tr>
<tr>
<td>Median of Local Entropy CK</td>
</tr>
<tr>
<td>Median of Local Entropy CD45</td>
</tr>
<tr>
<td>Median of Local Contrast DNA</td>
</tr>
<tr>
<td>Median of Local Contrast CK</td>
</tr>
<tr>
<td>Median of Local Contrast CD45</td>
</tr>
<tr>
<td>Standard Deviation Int. DNA</td>
</tr>
<tr>
<td>Standard Deviation Int. CK</td>
</tr>
<tr>
<td>Standard Deviation Int. CD45</td>
</tr>
<tr>
<td>HOG DNA</td>
</tr>
<tr>
<td>HOG CK</td>
</tr>
<tr>
<td>HOG CD45</td>
</tr>
</tbody>
</table>

Figure 2: Example of two cells present. Figure 2(a) is an example of a CTC, whereas Figure 2(b) is an example of a cell that is not a CTC. The red and green contours represent the countour resulting from segmentation.
implemented. By inspection of the histograms of each feature, by class, one can conclude that no feature clearly distinguishes one class from another. Only outliers related with abnormal size were removed based on inspection, all ROIs with an area $\leq 9\times 10^4$ and $\geq 0.3\times 10^4$ pixels were excluded from classification.

3.4. Classification Results

In this section it will be presented the performance of each implemented classification algorithm. Please note that no statistical hypothesis test was used for comparison purposes, the comparison was based solely on the ROC curves and AUC (Area Under the Curve). During implementation and evaluation, for smaller tests specificity ($\text{SPEC} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}$), sensitivity ($\text{SENS} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$) and balance accuracy ($\text{BAcc} = \frac{\text{SPEC} + \text{SENS}}{2}$) were used, however these results are not presented here. As stated before, first it was all tested with just one dataset, then another one and finally three datasets concatenated. Figure 4 presents the ROC curves of the three implementations of $k$-NN. Overall it is possible to observe that the classification performed by the $k$-NN, in either of the situations, is quite weak. All curves present an almost constant growth meaning that any increase in sensitivity will be accompanied by a linearly proportional decrease in specificity. Furthermore the curves are very close to the 45-degree diagonal and, as a result behaves nearly as random classifier. Between the three $k$-NN implementations, as expected, the $k$-NN coupled with the bootstrapping technique (Figure 4(a)) performed slightly better than the other two, due to the fact that it is implemented in such a way that tackles the problem of class imbalance. The best set of features was the DNA (Intensity+Texture). In all the cases the worst set of features was the CD45 (Intensity+Texture). The ROC curves for the Ensemble methods are depicted in Figure 6. Unexpectedly, on average the AdaBoost performed better than the RUSBoost. The best set for the AdaBoost (Figure 6(a)) was the CK (Intensity+Texture) and the worst was the CD45 (Intensity+Texture). In the case of the RUSBoost (Figure 6(b)) was the one that yielded better results Intensity and the worse the Morphology features set. Overall (considering all classifiers, the tests done with the two dataset separate and the test done with three datasets concatenated) the set of features that yielded worst classification results was the CD45 (Intensity+Texture), which is a bit odd given the fact that this is the exclusion marker. The set of features that generated better results was the CK (Intensity+Texture), followed by the Intensity features set. The best classifier was the AdaBoost.
followed by the SVM with RBF Kernel, however the results neither of the results meet the expectations. The results of all implemented classifiers are summarized in Table 2, in the form of AUC.

Table 2: Area Under the Curve (AUC) of each of the algorithms tested for the total of the 3 datasets.

<table>
<thead>
<tr>
<th>Area Under the Curve</th>
<th>All Morph.</th>
<th>User Int.</th>
<th>Tex.</th>
<th>DNA</th>
<th>CK</th>
<th>CD45</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-Nearest Neighbors w/ Prior P.</td>
<td>.6498</td>
<td>.5786</td>
<td>.6808</td>
<td>.6787</td>
<td>.6218</td>
<td>.5739</td>
</tr>
<tr>
<td>k-Nearest Neighbors</td>
<td>.6573</td>
<td>.5947</td>
<td>.6837</td>
<td>.6997</td>
<td>.6240</td>
<td>.5849</td>
</tr>
<tr>
<td>SVM Linear</td>
<td>.6267</td>
<td>.5702</td>
<td>.6316</td>
<td>.6056</td>
<td>.5702</td>
<td>.5188</td>
</tr>
<tr>
<td>SVM RBF</td>
<td>.6509</td>
<td>.5947</td>
<td>.6997</td>
<td>.6240</td>
<td>.5702</td>
<td>.5188</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>.7387</td>
<td>.6267</td>
<td>.7387</td>
<td>.6267</td>
<td>.7387</td>
<td>.6267</td>
</tr>
<tr>
<td>k-Nearest Neighbors</td>
<td>.7246</td>
<td>.6553</td>
<td>.7235</td>
<td>.6553</td>
<td>.7246</td>
<td>.6553</td>
</tr>
</tbody>
</table>

4. Conclusions

The main goal of this thesis was to study several approaches in order to build an automated classification system for Circulating Tumor Cells enumeration. Up to date, the interpretation of blood samples analysed by the CellSearch system, still depends on the expertise of a trained reviewer, and there has been a growing interest in developing automated systems that enumerate CTCs in a reliable fashion. This buzzing topic has also been growing due to the increasing amount of biomarkers, detection and physical isolation systems that are being studied and developed now, in order to perform real-time biopsies on cancer patients. This work presented a small summary of these systems, nevertheless it focused on the CellSearch System.

One of the objectives of this project was to study features that could be extracted from each cell and the impact they had on classification. The features analysed were related to the morphology, intensity and texture of each ROI. These were then grouped in sets in order to evaluate which were the ones that could be more informative and produced better classification results. It was concluded that the three best sets of features were the combination of all the extracted features, the set of features extracted from the CK channel (a combination of intensity and texture features from this channel) and the set of intensity features. The sets that generated worst results were the set of morphological features, and the texture and intensity features of both the DNA and the CD45 channels.

The second purpose of the current project was to evaluate and compare several pattern recognition systems. The large number of non-CTCs when compared with the scarce number of CTCs poses as problem that jeopardizes the classification systems, thus several approaches that tackle class imbalance were implemented and tested. The three machine learning algorithms that performed better were the two Support Vector Machines (that deal with class imbalance by associating weights to each of the classes) and the AdaBoost. The three worst were the three implementations of the k-Nearest Neighbor, and even in this case the one that performed best was the implementation with bootstrapping.

Overall all the implementations and results under-performed. It is not possible conclude, based on this thesis results, that it is possible to build an automated system for CTC enumeration of Small-Cell Lung Cancer.

To boost the results in CTC classification for SCLC several options can be studied and developed:

- Development of a more coherent and detailed ground-truth:
  - Stricter definition of what should be considered a CTC (different reviewers can assign the same object as different classes, even the same reviewer in different moments can classify the same object as a CTC one time and another time as a non-CTC);
  - Manual classification of CTCs after image segmentation (currently, in the manual classification, it is considered a CTC an area shaped like a rectangle that might contain one or more ROIs, and not necessarily all CTCs);
  - Manual classification in more than just CTC, for example also CTC debris and apoptotic CTC. These two can present a very different morphology and signal intensities compared to a normal CTC, and, currently, they are classified by an expert reviewer as a CTC;
  - Classification of non-CTC: the non-CTC class is everything else in the dataset, which creates a class (non-CTC) with very vague characteristics, it can be a white blood cell, debris, apoptotic cells, smudges, etc;

- Development of an automated classification system, that has more classes than just CTC and non-CTC;
- Using clustering and/or a learning algorithm for outlier removal;
- Studying the use of color histograms and other features;
- Using feature selection algorithms to better assess the informativeness of each feature, such as correlation and mutual information algorithms;
- Implementation of a noise reduction algorithm;
- Improvement of the segmentation algorithm.

The topic automated classification of Circulating Tumor Cells is still quite recent, has great potential and huge impact on the study of cancer and there is still a lot of room for development.

References

