Image Segmentation and Object Classification for Automatic Detection of Tuberculosis in Sputum Smears

By

Rethabile Khutlang

A thesis submitted to the

University of Cape Town in partial fulfilment of the

requirements for the degree of

MSc (Med)

in

BIOMEDICAL ENGINEERING

Faculty of Health Sciences

University of Cape Town

January 2009

Supervisor: Assoc Prof Tania Douglas

DECLARATION

I, …….., hereby declare that the work on which this dissertation is based is my original work (except where acknowledgements indicate otherwise) and that neither the whole work nor any part of it has been, is being, or is to be submitted for another degree in this or any other university.

I empower the university to reproduce for the purpose of research either the whole or any portion of the contents in any manner whatsoever.

Signature: ……

Date: ……..

ACKNOWLEDGEMENTS

I would like to thank Associate Professor Tania Douglas, my supervisor, for all her guidance and advice during my Masters project.

Thanks, also, to Sriram Krishnan, Andrew Whitelaw, Konstantinos Veropoulos and Genevieve Learmonth.

The financial assistance of the Department of Labour (DST), in the form of a scholarship, is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the DST.

ABSTRACT

An automated microscope is being developed in the MRC/UCT Medical Imaging Research Unit at the University of Cape Town in an effort to ease the workload of laboratory technicians screening sputum smears for tuberculosis (TB), in order to improve screening in countries with a heavy burden of TB.

As a step in the development of such a microscope, the project described here was concerned with the extraction and identification of TB bacilli in digital images of sputum smears obtained with a microscope. The investigations were carried out on Ziehl-Neelsen (ZN) stained sputum smears.

Different image segmentation methods were compared and object classification was implemented using various two-class classifiers, for images obtained using a microscope with 100x objective lens magnification. The bacillus identification route established for the 100x images, was applied to images obtained using a microscope with 20x objective lens magnification. In addition, one-class classification was applied the 100x images.

A combination of pixel classifiers performed best in image segmentation to extract objects of interest. For 100x images, the product of the Bayes', quadratic and logistic linear classifiers resulted in a percentage of correctly classified bacillus pixels of 89.38%; 39.52% of pixels were incorrectly classified. The segmentation method did not miss any bacillus objects with their length in the focal plane of an image. The biggest source of error for the segmentation method was staining inconsistencies. The pixel segmentation method performed poorly on images with 20x magnification.

Geometric change invariant features were extracted to describe segmented objects; Fourier coefficients, moment invariant features and colour features were used.

All two-class object classifiers had balanced performance for 100x images, with sensitivity and specificity above 95% for the detection of an individual bacillus after

Fisher mapping of the feature set. Object classification on images with 20x magnification performed similarly. One-class object classification using the mixture of Gaussians classifier, without Fisher mapping of features, produced sensitivity and specificity above 90% when applied to 100x images.

TABLE OF CONTENTS

Declaration……-2-

Acknowledgement.....-3-

Abstract……..-4-

Table of contents.....-6-

LIST OF FIGURES ..... - 10 -

LIST OF TABLES..... - 11 -

1 INTRODUCTION ..... - 12 -

1.1 Objectives..... - 13 -

1.2 Plan of Development..... - 14 -

2 LITERATURE REVIEW ..... - 16 -

2.1 Problem Description and Related Work by Others..... - 16 -

2.2 Image Segmentation..... - 17 -

2.2.1 Introduction..... - 17 -

2.2.2 Edge Detection..... - 18 -

2.2.3 Marker-Controlled Watershed Segmentation ..... - 18 -

2.2.4 Pixel Classifiers for Image Segmentation..... - 20 -

2.2.5 Filtering Segmented Objects..... - 21 -

2.2.6 Evaluating Segmentation Methods ..... - 21 -

2.3 Feature Extraction..... - 22 -

2.3.1 Introduction..... - 22 -

2.3.2 Extracted Features..... - 22 -

2.3.3 Types of Features..... - 26 -

2.3.4 Feature Normalisation..... - 26 -

2.3.5 Feature Representation ..... - 26 -
2.3.6 Object Labelling ..... - 29 -
2.4 Feature Subset Selection ..... - 30 -
2.4.1 Introduction ..... - 30 -
2.4.2 Feature Subset Selection Methods ..... - 30 -
2.5 Object Classification ..... - 33 -
2.5.1 Introduction ..... - 33 -
2.5.2 Receiver Operating Characteristics Analysis ..... - 34 -
2.5.3 Bayes' Classifier ..... - 35 -
2.5.4 Nearest Neighbour Classifier ..... - 35 -
2.5.5 Linear Classifier ..... - 36 -
2.5.6 Quadratic Classifier ..... - 37 -
2.5.7 Probabilistic Neural Networks ..... - 37 -
2.5.8 Support Vector Machines ..... - 38 -
2.5.9 Subspace Learning ..... - 40 -
2.5.10 One-class Classification ..... - 41 -
2.5.11 Conclusion ..... - 42 -
3 MATERIALS ..... - 43 -
3.1 100x Microscope ..... - 43 -
3.2 20x Microscope ..... - 44 -
3.3 Discarded Images ..... - 46 -
4 METHODS ..... - 47 -
4.1 Object Labelling ..... - 47 -
4.2 Segmentation Methods ..... - 48 -
4.2.1 Segmentation Evaluation ..... - 49 -
4.2.2 Marker-Controlled Watershed Segmentation ..... - 49 -
4.2.3 Combination of Multiple Pixel Classifiers ..... - 50 -

4.2.4 Contextual Pixel Classification using Canny Edge Detector ..... - 51 -
4.3 Feature Extraction Methods ..... - 52 -
4.3.1 Extracted Features ..... - 52 -
4.3.2 Data Normalisation ..... - 53 -
4.3.3 Linear Fisher Transform ..... - 53 -
4.4 Feature Subset Selection Methods ..... - 53 -
4.4.1 Population Based Incremental Learning ..... - 53 -
4.4.2 Correlation Based Feature Subset Selection ..... - 54 -
4.4.3 Sequential Floating Forward Selection ..... - 55 -
4.4.4 Branch and Bound Feature Subset Selection ..... - 55 -
4.5 Object Classification Methods ..... - 56 -
4.5.1 Bayes' Classifier ..... - 56 -
4.5.2 Nearest Neighbour Classifier ..... - 57 -
4.5.3 Logistic Linear Classifier ..... - 57 -
4.5.4 Quadratic Discriminant Classifier ..... - 57 -
4.5.5 Probabilistic neural networks ..... - 58 -
4.5.6 Support Vector Machines ..... - 58 -
4.5.7 Subspace Learning ..... - 58 -
5 RESULTS: TWO-CLASS CLASSIFICATION AT 100x MAGNIFICATION ..... - 60 -
5.1 Segmentation Results ..... - 60 -
5.1.1 Marker-Controlled Watershed Segmentation ..... - 60 -
5.1.2 Combination of Multiple Pixel Classifiers ..... - 62 -
5.1.3 Contextual Pixel Classification using Canny Edge Detector ..... - 64 -
5.1.4 Discussion on Segmentation ..... - 65 -
5.2 Feature Extraction and Feature Subset Selection ..... - 67 -
5.2.1 Filtering ..... - 67 -
5.2.2 Feature Subset Selection ..... - 67 -

5.3 Object Classification ..... - 69 -
6 RESULTS : TWO-CLASS CLASSIFICATION AT 20x MAGNIFICATION ..... - 76 -
6.1 Segmentation Results ..... - 76 -
6.2 Object Classification ..... - 78 -
6.3 Discussion ..... - 79 -
7 RESULTS: ONE-CLASS CLASSIFICATION AT 100x MAGNIFICATION ..... - 81 -
7.1 Segmentation ..... - 81 -
7.2 Object Classification ..... - 83 -
7.3 Discussion ..... - 85 -
8 DISCUSSION, CONCLUSIONS AND RECOMMENDATIONS ..... - 87 -
APPENDICES ..... - 90 -
REFERENCES ..... - 95 -

LIST OF FIGURES

Figure 2.1: An example ROC curve. 34
Figure 3.1: An image produced by the Nikon microscope at 100x magnification. Bacilli are red. 44
Figure 3.2: The setup for capturing a digital image with the Interscopic microscope. 45
Figure 3.3: An example image produced by the Interscopic microscope at 20x magnification. 45
Figure 3.4: An example of an image not used in the study. 46
Figure 4.1: A screen shot of the GUI used for labelling. 48
Figure 5.1: The original image (left), the results after marker-controlled watershed segmentation (middle) and the results after filtering based on object area (right). 61
Figure 5.2: Percentages of correctly classified pixels for increasing number of combined classifiers. 64
Figure 5.3: The outlines of segmented objects produced by the product of three classifiers overlaid on three sub-images of the test dataset. 64
Figure 5.4: The outlines of segmented objects produced by the contextual pixel classifier overlaid on three sub-images of the test dataset. 65
Figure 5.5: Results of the best combination of pixel classifiers on an image with red stains. 66
Figure 5.6: Example bacillus and non-bacillus objects. 69
Figure 5.7: Silhouettes plots of the bacillus (left) and non-bacillus (right) classes. 71
Figure 5.8: ROC curves for all classifiers. 72
Figure 5.9: The results of the established identification route (using logistic linear classifier) on an example image; bacillus objects are in red and non-bacillus objects in blue in the right image; touching bacilli are labelled as non-bacilli. 75
Figure 6.1: Percentages of correctly classified pixels for increasing number of combined classifiers. 77
Figure 6.2: Sub-images of the product of classifiers results overlaid on the first three images of the test dataset. 78
Figure 7.1: The results obtained using the mixture of Gaussians classifier overlaid on two images of the test dataset. 82
Figure 7.2: The decision boundary of the mixture of Gaussians classifier. 86

LIST OF TABLES

Table 5.1: The ratios of correctly and incorrectly classified pixels resulting from watershed segmentation applied to the blue channel of the RGB colour spaces. 61
Table 5.2: For each classifier in each colour space, the first entry is the ratio of correctly classified pixels; the second entry is the ratio of incorrectly classified pixels. 63
Table 5.3: Features selected using different selection methods. 68
Table 5.4: Evaluation of each classifier at its best point as found by cross-validation. 70
Table 5.5: Dissimilarity space reduction applied for the *k*NN classifier. 70
Table 5.6: Evaluation of classifiers after linear Fisher transformation and subspace learning. 71
Table 5.7: Evaluation of classifiers applied after Fisher mapping, by means of the area under the ROC curve. 72
Table 6.1: For each classifier in each colour space, the first entry is the ratio of correctly classified pixels; the second entry is the ratio of incorrectly classified pixels. 77
Table 6.2: Evaluation of classifiers on the Fisher mapped feature set. 79
Table 7.1: Performance evaluation of pixel classifiers. 82
Table 7.2: Evaluation of classifiers using different feature sets. 84
Table 7.3: Evaluation of classifiers using area under the ROC curve. 84
Table 7.4: Evaluation of classifiers using the set of all features. 85
Table 7.5: Evaluation of classifiers by means of the area under the ROC curve. 85

1 INTRODUCTION

The World Health Organisation (WHO) has declared tuberculosis (TB) a global emergency. TB is an airborne infectious disease that spreads easily in densely populated areas with poor sanitation. In low income countries the spread has been exacerbated by co-infection with HIV/AIDS. Consequently, the health services in these countries are severely strained. Currently TB kills about 2 million people annually; both the highest number of deaths and the highest mortality per capita are in the Sub-Saharan Africa region (WHO, 2007). A person with active TB infects 10-15 people annually. Early detection is vital for the monitoring and treatment, and hence control, of TB.

An automated microscope is being developed in the MRC/UCT Medical Imaging Unit Research at the University of Cape Town in an effort to ease the workload of laboratory technicians screening sputum smears for TB in countries with a heavy burden of TB.

TB is caused by *Mycobacterium tuberculosis*, which enters the body through the lungs. A person whose lungs are infected with the micro-organism is said to have pulmonary TB. *Mycobacterium tuberculosis* expectorated in sputum can be seen as clusters or individually in stained sputum smears under a microscope. There are two staining methods: auramine staining which requires a fluorescence microscope; and Ziehl-Neelsen (ZN) staining which requires a bright field microscope.

The microscope is at the heart of TB screening in low- and middle-income countries (Steingart et al., 2006), as it is a relatively cheap piece of equipment. The WHO calculates global incidence, prevalence and mortality from tuberculosis. Positive sputum smear microscope detection makes up the largest fraction of total TB detections (WHO, 2007).

A major shortcoming of conventional microscope TB screening is that sensitivity is low and variable - values in a range of 20-80% have been reported (Burdash et al., 1976; Lesarson et al., 2005; Steingart et al., 2006). Steingart et al. (2006) reviewed a number of

studies and concluded that fluorescence microscopy for TB screening has higher sensitivity than bright field microscopy. However, bright field microscopy of ZN-stained sputum smears is the method of choice in developing countries, due to the low cost and ease of equipment maintenance compared to fluorescence microscopy.

The aim of automation in the context of TB screening is to help the technician speed up the screening process and prevent errors due to boredom and fatigue. There is a shortage of senior pathologists to verify manual screening – as stipulated by the World Health Organisation – in developing countries. A technician normally examines between 30 and 40 smears in a day (Toman, 2004). Technicians look at a set number of fields on a slide and may diagnose a positive slide as smear negative because of sparseness of bacilli.

Automation might improve the low sensitivity of conventional TB screening, and might also reduce the variability of human operators in diagnosing a slide.

The project described here represents a step in the development of an automated microscope for TB detection and was concerned with the extraction and identification of TB bacilli in digital images of sputum smears. The investigations were carried out on ZN-stained sputum smears.

1.1 Objectives

The objectives of the study were to:

- Develop image segmentation algorithms to extract candidate bacillus objects in digital images of ZN-stained sputum smears obtained using a microscope.

- Extract descriptive features from segmented objects and perform subset selection on the features extracted, in order to select the most descriptive set of features.

- Apply pattern recognition algorithms for the classification of candidate objects as bacilli or non-bacilli, based on their features.

The purpose of segmentation is to limit the amount of information to be processed in the pattern recognition stage and to preserve objects with the shape and colour of bacilli. The feature extraction stage yields numerical values used as features of the bacillus population. The segmented objects can be clustered into groups based on the selected features. Classification is the pattern recognition stage where it is determined if an object is a bacillus or not.

The work was implemented in MATLAB (MathWorks, 2007).

1.2 Plan of Development

Chapter 2 comprises a literature review for the automatic detection of tuberculosis in sputum smears.

Chapter 3 describes the materials used in the study.

Chapter 4 details the methods that were used in the identification of TB bacilli.

Chapter 5 presents the results of segmentation and two-class classification for the automated detection of bacilli in sputum smear images obtained using a microscope with 100x magnification.

Chapter 6 presents the results of the identification route established in Chapter 5 applied to images from a 20x microscope.

Chapter 7 presents the results of classification with the identification problem defined as novelty detection or one-class classification.

Chapter 8 presents the overall conclusions drawn from the study. Recommendations for future work are also made.

2 LITERATURE REVIEW

This chapter reviews methods that may be suitable for the automatic detection of tuberculosis in sputum smears. The literature reviewed is specific to the automated identification of *Mycobacterium tuberculosis* in sputum smears.

2.1 Problem Description and Related Work by Others

Captured images cannot be fed straight to learning algorithms for object detection because they contain a lot of information, which would increase the computational requirements of the algorithms. Image segmentation decreases the amount of information in an image by extracting objects of interest.

Features are extracted from the segmented objects. A feature describes an aspect or a combination of aspects of an object in a way that would allow identification of the object. Objects of interest are classified as either bacilli or non-bacilli based on their features.

Classifier performance decreases with an increase in the number of features, because the classifier becomes more complicated and it is difficult to optimise a complicated function. As a result, features are selected according to their relevance and independence, in the feature selection stage.

Classification is achieved by a function that can learn a mapping of objects into respective classes from example objects and their classes. If classes of example objects are not given, classification is said to be unsupervised. The first step of unsupervised classification is to group or cluster example objects using the nearness of features' values; 'unseen' objects are classified into a group of example objects with similar features.

Veropoulos et al. (1999) explored the use of an automated method to detect tubercle bacilli in sputum specimens. They concluded that an automated method is feasible as they obtained high accuracy for the detection of individual bacilli in captured images of auramine stained sputum; artificial neural networks were used to classify detected objects as either bacilli or non bacilli. Forero et al. (2006) extended the automated detection process to unsupervised classification, using auramine-stained sputum.

Sadaphal et al. (2008) demonstrated proof of principle, without quoting accuracy, that colour-based Bayesian segmentation may be employed to extract TB bacilli in ZN-stained sputum smears. Santiago-Mozos et al. (2008) used support vector machines to identify bacilli in captured auramine-stained sputum smear images. They used one image to classify a subject as TB infectious or non infectious; they also defined an uncertainty region for which the system requested another image for classification if an image fell within it.

2.2 Image Segmentation

2.2.1 Introduction

Veropoulos et al. (1999) used the Canny edge detector to find edges of objects in images of auramine-stained sputum. In a preliminary study on microscope images of ZN-stained sputum smears, Veropoulos (2001) converted captured images from the Red-Green-Blue (RGB) colour model to the Hue-Saturation-Intensity (HSI) colour model. The saturation band was thresholded and the Canny edge detector applied to find the contours of potential TB bacilli. Forero et al. (2006) used Canny edge detection and morphological operations to retain objects having morphology similar to that of TB bacilli in auramine stained sputum smear images; they used the RGB colour model.

Russell (2006) experimented with grey scale and colour image segmentation methods. Grey scale segmentation yielded results that were not useful for bacillus classification. Russell (2006) used colour compensation, proposed by Castleman (1998) to segment

captured sputum smear images. Colour compensation is the correction of the spread of an RGB colour channel into the other two channels. The green channel showed bacilli the most clearly. It was colour compensated, then thresholded to segment bacilli. The threshold was determined empirically.

Different segmentation methods implemented in the study reported on here are described in more detail below.

2.2.2 Edge Detection

Veropoulos et al. (1999) and Forero et al. (2006) suggested Canny edge detection for the segmentation of bacilli in images of auramine stained sputum smears.

Canny edge detection first requires convolution of an image with a two-dimensional Gaussian, and then with the derivative of the Gaussian, in order to smooth the image. The local gradient and gradient direction are then computed at each point. The threshold of the gradient strength is determined by finding the point in the histogram of the image, after the convolutions, at which the cumulative sum exceeds the estimated fraction of pixels not belonging to edges. A second threshold is used to include weak edges if they are connected to strong edges. It is found empirically as a fraction of the main threshold.

2.2.3 Marker-Controlled Watershed Segmentation

Digabel and Lantuejoul (1977) were the first to apply the watershed transformation as a morphological operation in image processing. It has developed into one of the major alternatives to detection of discontinuities, thresholding, or region processing segmentation methods. The watershed segmentation transformation is based on the concept of a watershed in the field of topography. The pixels of the image represent elevation at their respective locations. Points at which a water droplet, if placed, would

be at a minimum, or would fall to a minimum, form a catchment basin or watershed. On the other hand, points at which a water droplet, if placed, would fall equally likely to more than one minimum form divide lines, which are the desired segmentation lines.

There are different ways of implementing watershed segmentation, most of which are based on multiple complete scanning of the image being processed (Vincent and Soille, 1991). Segmentation methods that perform multiple scanning of an image are very slow. Vincent and Soille (1991) proposed a different algorithm that is based on sorting image pixels in an increasing order of their gray values, then performing progressive image flooding. The method is explained below.

Image gray levels are sorted to determine their frequency distribution. The cumulative frequency distribution of the image is then computed and is used to assign each image pixel to a unique cell in the sorted array. The image is then progressively flooded. With each flooding, catchment basins whose minimum is less than the flood level *h* are assigned unique labels. Since the image was sorted, pixels at height *h +1* can be accessed and are named the mask. Pixels in the mask that have neighbours as already labelled pixels are assigned to a queue. A queue is a large array of pointers to pixels. The queue enables extension of labelled catchment basins (inside the mask) by computing geodesic influence zones (Appendix A). The process is repeated until the flood level is at the maximum of the sorted array of pixels.

The gradient image is often used as the input to the watershed algorithm. However, due mainly to noise, watershed usually results in over-segmentation. The results can be improved by introducing markers. Internal markers are connected components inside each of the objects of interest, while external markers engulf each object and are contained within the background. The watershed algorithm then only finds the divide lines contained between internal and external markers. Gonzalez et al. (2004) used the extended minima transform for the internal markers. The transform yields the set of minima in an image that are deeper than a height *h*, determined empirically. For the external markers, they used watershed results of the distance transform of the internal

marker image. The distance transform is the matrix the size of an input image containing distances from every pixel to the closest nonzero-valued pixel.

2.2.4 Pixel Classifiers for Image Segmentation

A classifier uses the features of an object to determine its class. Section 2.5 provides background to classifiers. Features describe certain characteristics of objects to be classified. Pixel classifiers can be used for segmenting an image where objects are image pixels. For pixel classifiers, the intensity value of the pixel is usually used as the feature. The major drawback of pixel classifiers is that they do not take spatial information into account (Sollie, 2003). There are numerous ways of addressing this problem.

One way of addressing the spatial information problem is to classify a pixel based on its class and that of its neighbours. The pixel could be assigned a class if the majority of its neighbours share that class; for objects with a specific shape the neighbours can be chosen in accordance with that shape. Twellmann et al. (2001) used 15 x 15 pixel patches to classify fluorescent lymphocytes in micrographs of tissue sections. Lenseigne et al. (2007) described each pixel by a 9-dimensional feature vector, corresponding to its 3 x 3 neighbourhood, to segment confocal microscope images to quantify the amount of *Mycobacterium tuberculosis* for drug-discovery. Long et al. (2004) used principle components analysis to reduce the number of features in a micrograph patch for detection of unstained viable cells in bright field microscope images.

The second way of addressing the spatial information problem is to combine pixel classification and conventional segmentation techniques (Sollie, 2003). Examples of conventional segmentation techniques are edge detection and histogram thresholding methods. Sonka (2008) listed various ways to use contextual information to improve the performance of pixel classifiers. Firstly, a post-processing filter can be applied to a segmented image; small regions then disappear as they are assigned to the most probable object in their neighbourhood. Secondly, pixels can be merged into homogeneous

regions and such regions classified instead of individual pixels. Lastly, pixel labels in a neighbourhood of a pixel can form a feature vector which, together with neighbourhood pixels' intensity values, goes through a second classifier to assign the label of the corresponding pixel. Any contextual pixel classification scheme performs better for multispectral images, because any two channels with low correlation are more descriptive than a single channel.

Meuric et al. (2005) used a combination of pixel classifiers to implement contextual pixel classification for segmentation of microscope images for lung cancer diagnosis. They ranked different classifiers according to their performance, and then combine top performing classifiers to improve pixel classification. There are various ways to combine pixel classifier outputs; the label of a pixel can be determined by voting among outputs of chosen classifiers, or by averaging classifier outputs. Besides choosing the combination scheme, the number of pixel classifiers to use in combination is another parameter that must be specified.

2.2.5 Filtering Segmented Objects

Segmentation is applied to decrease the amount of image information input to a classifier. Forero et al. (2006) filtered segmented images to decrease the number of objects and hence further decrease the amount of information in segmented images. Their filter was based on area and eccentricity. Objects whose area was too small or too large or whose eccentricity was too small were rejected. Different features can be explored to optimise the filtering process.

2.2.6 Evaluating Segmentation Methods

Meurie et al. (2003) performed segmentation of images of bronchial cells to aid in the diagnosis of lung cancer. They proposed a segmentation evaluation method that uses a

manually segmented reference image to provide true or false classification rates. To compute these rates, they found the common rate and the difference rate. The common rate is the number of pixels belonging to objects that are correctly classified; the difference rate is the number of pixels that belong to objects in the reference image but are not identified as the same class in the segmented image and pixels that belong to background in the reference image identified as object pixels in the segmented image. For each class, the common rate is averaged by the reference image object pixels; while the difference rate is averaged by the union of the reference image object pixels and the segmented image object pixels. According to Egmont-Petersen et al. (2002), finding common and difference rates does not give within-region homogeneity and between-region heterogeneity, which are crucial in quantifying segmentation quality. However, common and difference rates can be used to objectively rank different segmentation methods according to performance.

2.3 Feature Extraction

2.3.1 Introduction

Features are extracted from the segmented objects. Bacilli in colour sputum smear images have shape and colour as important features or descriptors. Forero et al. (2006) used Hu's moments, compactness and eccentricity – all shape features – for auramine stained sputum smear images. Veropoulos (2001) used average RGB values inside and around an object, and the standard deviation of RGB values inside and around an object as colour features for objects in images of ZN-stained sputum; Fourier features were used for shape description.

2.3.2 Extracted Features

Objects in a microscope view field are not ordered in any manner, but are scattered in the background. If the segmentation stage is thorough, only objects with features similar

to those of TB bacilli will be preserved in an image. However, segmentation does not change the positioning of preserved objects. So the features to be extracted should not be affected by the relative position of the object in an image, if they are to be descriptive. They should be geometric change invariant; common geometric changes are scaling, translation and rotation. Furthermore, if the images are captured under differing lighting conditions, their colour information will differ. So colour features should be invariant to such changes - photometric changes.

Boundary Descriptors

The length of the object boundary is the simplest boundary descriptor. The diameter of the boundary can be a descriptor if it is unique. The major axis is a descriptor and so is the minor axis, a line perpendicular to the major axis; their ratio is called eccentricity and it is also a descriptor (Gonzalez et al., 2004). Eccentricity is a relevant descriptor because of the long and thin shape of the bacilli.

Fourier Descriptors

The 2-dimensional coordinates of the boundary pixels of an object form a closed shape. Starting at an arbitrary point, the boundary can be represented as a complex sequence of coordinates. The second value of each coordinate is made imaginary,

$$s(k) = x(k) + jy(k)$$

for k = 0,1,2,…K-1. K is the number of boundary pixels. The discrete Fourier transform of $s(k)$ is

$$a(u) = \sum_{1}^{K-1} s(k) e^{\frac{-j\pi 2ku}{K}}$$

The complex coefficients $a(u)$, u = 0, 1, 2…K-1, can be used as Fourier features (Gonzalez et al., 2004). The sequence $s(k)$ can be formed using the complex coefficients

and the inverse Fourier transform. The boundary will have the same number of pixels even if few coefficients $a(u)$ are used to reconstruct each point of the sequence $s(k)$. High frequency coefficients account for fine details, while low frequency components account for the global shape.

Fourier features can be made invariant to translation and rotation using the transform: $r(u) = \sqrt{|a_x(u)|^2 + |a_y(u)|^2}$ (Sonka, 2008); where $a_x(u)$ and $a_y(u)$ are the real and imaginary parts of the coefficients or descriptors. Scale invariance is achieved by the transform: $w(u) = r(u)/r(1)$. Veropoulos (1999) used 15 Fourier coefficients.

Statistical Moments

Mean, variance, and higher-order moments can be used to describe the shape of the 1-D object boundary representation (Gonzalez et al., 2004). They are rotation, translation and scale (geometric transformation) independent (Forero et al., 2006). 2-D moments describe object regions.

Forero et al. (2006) used Hu's moments to describe bacilli. The function used in the evaluation of moments considers a binary image; hence Hu's moments do not consider colour information. On the other hand, Mindru et al. (2004) presented moment invariants that combine shape and colour information.

Moments presented by Mindru et al. (2004) use functions representing colour space bands and characterise the shape and colour distribution of a pattern uniformly. Such moments are referred to as generalised colour moments. Generalised colour moments yield a set of features (moment invariants) larger than that obtained with Hu's moments for a minimum order used. Hu's moments provide a less general form of invariance to geometric changes because moment invariants' stability decreases with an increase in order (Mindru et al., 2004). Therefore generalised colour moments, order confined to one, are an improvement on Hu's moments that use orders up to three.

Moment invariants derived from generalised colour moments are not only geometric change invariant, they are also photometric transformation invariant. They are therefore considered relevant to the description of bacilli. Appendix B contains equations for the derivation of generalised colour moments and moment invariants used to describe bacilli in this study.

Forero et al. (2006) state that moment invariants and Fourier features are rarely used together, because they both describe shape features. The moments they considered were Hu's. But the moment invariants presented by Mindru et al. (2004) describe how shape and colour information combine. Therefore they can be used in conjunction with Fourier features which only describe shape information.

Compactness

Compactness provides a measure of how closely the shape of the object approaches a circle; it is the ratio of the perimeter and area of the object (Forero et al., 2006). Bacilli are long and thin, so compactness may be a descriptive feature. Trattner et al. (2004) used a similar measure – the shape index – to characterise bacterial types. High values of the shape index represent objects with high circularity.

Colour

The bacilli that cause TB are acid fast bacilli (AFB). They stain red and the background blue when stained with the ZN method. Therefore colour features can be used to distinguish TB bacilli in images of ZN-stained sputum smears as Veropoulos (2001) suggested.

2.3.3 Types of Features

Features can either be categorical or continuous. A categorical feature takes any of a number of categories. On the other hand, continuous features can be plotted on a number line. All the features considered for the description of TB bacilli are continuous. A proximity-based classifier uses nearness of query objects' features to features of known objects for classification and is suitable for continuous features such as those used for TB bacillus detection.

2.3.4 Feature Normalisation

Most classifiers use distance to training data points to label query points. The query point takes the label of a training point whose features are closest to its own. Therefore features should preferably have the same units of measurement. With the Euclidean distance used, some features might have higher values and dominate the labelling decision, as the distance metric does not take into account the dispersion of the descriptors. To guarantee fairness to all features, data may be normalised. Normalisation has a further advantage of minimising the effect of outliers (Theodoridis and Koutroumbas, 2003). The data is normalised by subtracting the mean of each feature from each feature element, then dividing each feature element by the standard deviation of that feature. This ensures that all features have zero mean and standard deviation of one.

2.3.5 Feature Representation

As an alternative to using the extracted features directly in the classifier, the features could be mapped to a feature space that captures maximal variation between objects. The following feature representation schemes are commonly used.

Principal Components Analysis

Images having the same dimensions can be stacked together. If there are *n* such images, the vector x_i contains the n pixel values for each pair of coordinates at location *i* (Gonzalez et al., 2004).

$$x_i = [x_1, x_2 ... x_n]'$$

If the images have the size $M * N$, there are MN such vectors. Their $n * n$ covariance matrix, real and symmetric, can be used to find a set of n orthonormal eigenvectors. The principal components are given by

$$y = A(x - m_x)$$

where the rows of the matrix A are the normalised eigenvectors (Gonzalez et al., 2004). The transform uses the largest eigenvalues of the covariance matrix, hence the name principal components, to reconstruct the vector x.

Singular Value Decomposition

Principal components analysis can be applied only to square matrices, for instance the covariance matrix. Conversely, singular value decomposition (SVD) can be performed on any matrix. SVD is a factorisation of a matrix *X* into the diagonal matrix *S* of the same dimensions of *X* and unitary matrices *U* and *V* such that

$$X = U * S * V'$$

The columns of matrix *U* are in the direction of the principal axes and the elements of *S* yield their lengths (MathWorks, 2007). A unitary matrix has its columns orthogonal to each other.

Principal components analysis (PCA) yields true eigenvectors of a square input matrix. SVD uses the feature vectors to find singular values of the feature space. SVD translates true features to a new feature space; it returns whitened data – the input data with features orthogonal to each other. Features corresponding to the largest eigenvalues or singular values capture maximum variance across the set of features.

Non-linear Fisher Mapping

Scatter matrices are class separability criteria based on information related to the way feature vectors are scattered in the feature space (Theodoridis and Koutroumbas, 2003). They are not based on the Gaussian assumption of the distribution of a dataset. The within class scatter matrix S_w defines the variance of features for each class.

$$S_w = \sum_{i=1}^{m} P_i S_i$$

where S_i is the covariance matrix for each of the *m* classes; P_i is the prior probability of each class. The between-class scatter matrix S_b defines the between class distances for all classes

$$S_b = \sum_{i=1}^{m} P_i (u_i - u_o)(u_i - u_o)^T$$

where u_o is the global mean vector and u_i is the mean of each class *i*. Fisher mapping reduces dimensionality of the feature space based on the optimisation of the between class scatter matrix with respect to the within class scatter matrix (Franco et al., 2006). PCA accumulates the variance of the feature space in the first few features, while the Fisher transform maps the feature space to a low dimensional space that aims to better

discriminate between different classes. Franco et al. (2006) applied Fisher transforms to improve the classification task of face recognition.

The non-linear Fisher transform is an improvement on the Fisher transform for multi-class problems and generally for problems with low separability between the classes. It applies PCA to the dataset. For standard PCA, large between-class distances dominate the first eigenvectors of the S_b matrix. Their influence is minimised by introducing a weight factor, based on the error function, in the definition of S_b (Franco et al., 2006).

$$S_b = \sum_{i=1}^{m-1} \sum_{i-1}^{m} P_i P_j w(d_{ij})(u_i - u_o)(u_i - u_o)^T$$

$$w(d) = (2d^2)^{-1} erf(\frac{d}{2\sqrt{2}})$$

$$erf\ (a) = 2(\sqrt{\pi})^{-1} \int_{t=1}^{a} \exp(-t^2) dt$$

where d_{ij} is the square root of the Mahalanobis distance. With this modification to the S_b matrix, small between-class distances are accentuated according to their error contribution. The mapping is performed using the eigenvectors corresponding to the largest eigenvalues of S_b.

2.3.6 Object Labelling

Some segmented objects are bacilli, while others are not. Objects are labelled as bacilli or non-bacilli so that a classifier can use them for training. Veropoulos et al. (1999) had objects on the images of the sputum smear view fields labelled by an expert. Forero et al. (2006) had an expert label objects and edited the data to remove outliers.

2.4 Feature Subset Selection

2.4.1 Introduction

Feature subset selection involves selecting the best or most descriptive subset of features for classification. It uses an evaluation function which can be monotonic or non-monotonic. In the case of monotonic evaluation, a feature subset cannot have a larger evaluation than a larger subset that contains it. There are different ways of selecting the best subset, some of which are optimal and some suboptimal. An optimal method cannot miss the best subset from all possible subsets, whereas a suboptimal method can.

Feature subset selection methods that use the classifier to evaluate subsets are called wrapper methods, while those that use an independent evaluation function are called filter methods and are faster than wrapper methods (Hall, 2000). Forero et al. (2006) do not use feature selection methods. They discard features that are functions of others; for example eccentricity, which can be derived from Hu's moments. Feature subset search methods are given below, together with evaluation functions usually used with them.

2.4.2 Feature Subset Selection Methods

Thornton's Separability Index

Accuracy of classification can be measured by splitting data into train and test sets and observing how many test objects are classified correctly. The process has to be done many times, with different sets; accuracy is the average result. Greene (2001) proposes Thornton's separability index (SI) as a measure of feature subset merit that yields results similar to accuracy of a nearest neighbour classifier obtained with multiple data splits, but is fast and easy to calculate.

SI can be used to perform an exhaustive search of the feature space and come up with the best feature subset. However, if the number of features is large, an exhaustive search can be computationally expensive. For N features 2^N subsets will be evaluated. Population-based incremental learning (PBIL) may be used to evaluate the merit of a feature subset based on SI where it is impractical to evaluate every possible subset. Feature subsets are selected probabilistically; progressively the search is prejudiced toward subsets that yield a higher SI evaluation by using weights (Baluja, 1994).

Correlation Based Feature Selection

The correlation based feature selection (CFS) algorithm evaluates features as a subset, not individually (Hall, 2000). It is a filter feature selection algorithm hence it does not evaluate a feature subset using a classifier, as wrapper feature selection algorithms do.

The best first search strategy is used by the CFS algorithm to search the feature space (Hall, 2000). It begins with an empty subset and adds a feature with the highest evaluation. Then the subset is expanded by adding a single feature that improves the evaluation. The process is repeated until a point where expanding the subset does not improve the evaluation. The procedure stops after five consecutive non-improving subsets. The evaluation used is

$$Merit = \frac{kr_{cf}}{\sqrt{k + k(k-1)r_{ff}}}$$

where r_{cf} is the average feature-class correlation and r_{ff} is the average feature-feature correlation for a subset with *k* features (Hall, 2000). The dot product of two vectors can be interpreted as their correlation and reveals the directional relationships of the two vectors. It is usually normalised by the magnitudes of the two vectors to the range -1 to +1. -1 means the two vectors are parallel and in opposite directions, while zero means

they are orthogonal. The evaluation function selects features to a subset that are uncorrelated to current features in the subset, yet highly correlated to the class vector.

Oscillating and Floating Search Methods for Feature Selection

Sequential forward or backward selection (SFS or SBS) algorithms are fast. They consist of a set number *F* of steps forward or backward to find the next best feature. This is done until a desired number of features are obtained. Their flaw is that they perform a single-track search (Pudil et al., 1994). Once the feature is removed or added it remains so even if it could become significant or insignificant in subsequent subsets.

Floating search methods are an improvement of SFS and SBS algorithms. Sequential floating forward or backward selection (SFFS or SBFS) make use of backtracking. Backtracking is applied as long as the resulting subsets have better evaluations than the previous evaluation at that level. Pudil et al. (1994) set the stopping criterion as having obtained the number of features desired. For non-monotonic evaluation functions, the stopping criterion could be set as a particular number of consecutive non-improving feature inclusions or exclusions. Therefore non-monotonic floating search methods would have no parameter setting.

Oscillating search methods refine a subset of *D* features from *M* available features. They operate by removing *d* worst features from the current subset then adding *d* best features from *M* total features to that subset (Somol and Pudil, 2000). The process is repeated with varying *d* values until a set criterion is met. Oscillating search methods can be used to refine the results of floating search methods. A subset of *D* features to be refined can be the best subset output by floating search methods. It is generally difficult to select a subset to be refined if the floating search method uses a monotonic evaluation criterion (Somol and Pudil, 2000).

Branch and Bound Feature Selection

Veropoulos (2001) used branch and bound (B&B) and SFS feature selection algorithms for bacillus identification. The SFS algorithm is sub-optimal, while B&B is optimal (Theodoridis and Koutroumbas, 2003). B&B uses a monotonic evaluation function to select the best subset of *d* features out of *M* features. It models a tree where the root is the set of all features and the leaves are subsets with *d* features. The evaluation function is used to follow a path with the highest evaluation; that path leads to the best subset. The disadvantage of B&B is that it is slow; Somol et al. (2004) present a B&B algorithm that does not compute all evaluations – it predicts some if they are not at the leaves – to gain speed.

2.5 Object Classification

2.5.1 Introduction

A classifier that uses example objects that are labelled to learn a mapping of objects to labels is a supervised classifier. An unsupervised classifier learns from example objects that do not have labels. Forero et al. (2006) used unsupervised classification; a clustering algorithm grouped the example objects (training data points), and a probabilistic classifier assigned 'unseen' examples (test data points) to a group with a dominant distribution at their location.

The generalisation performance of a classifier is the discrimination of its decision boundary between different classes (Duda et al., 2001). If a classifier is less simple than the optimum required, the boundary is susceptible to noise and over-fits training data; and vice versa. Hence good generalisation performance corresponds to optimum complexity of a classifier.

At times it is important to know the ratios of misclassified points in each class. Knowing such ratios can help develop a classifier with balanced performance (similar ratios of misclassified points in each class).

Sensitivity is the ratio of true positive predictions to the total positive predictions of a classifier. Specificity is the ratio of true negative predictions to the total negative predictions made. A method of combining sensitivity and specificity to get a better picture of the classifier performance is given below; then the different classifiers are introduced.

2.5.2 Receiver Operating Characteristics Analysis

Receiver operating characteristics (ROC) analysis is used to evaluate the performance of a classifier (Duda et al., 2001). The ROC space is two dimensional, thus classifiers can be compared or evaluated by a point in the plane. Usually sensitivity is plotted on the *y*-axis, and (1 – specificity) on the *x*-axis. The ROC curve is obtained by varying the classifier threshold between its extremes. Points on the curve closest to the upper left corner of the ROC space correspond to classifier parameters that yield good performance. An example ROC graph is shown in Figure 2.1.

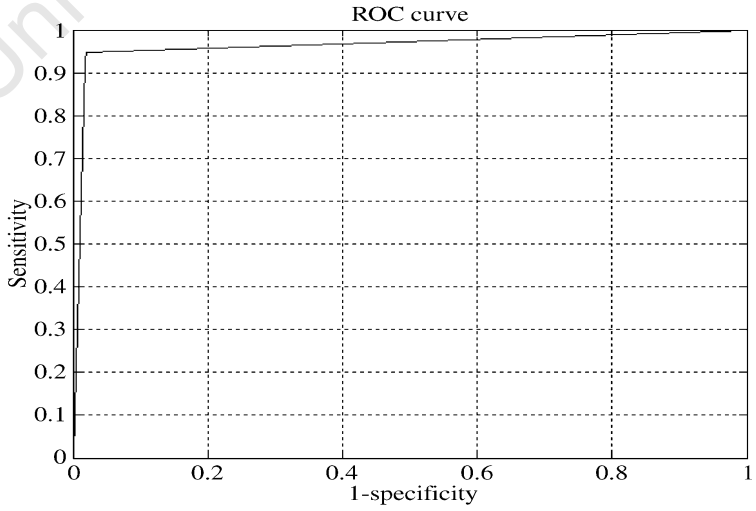

Figure 2.1: An example ROC curve.

2.5.3 Bayes' Classifier

Forero et al. (2006) used Bayesian classification (Duda et al., 2001) to identify bacillus and non-bacillus objects in captured microscope images of auramine stained sputum. The Bayes classifier is optimal in that it minimises the probability of error in assigning a class to an object. It is based on Bayes' rule,

$$P(A \mid x) = P(x \mid A) * \frac{p(A)}{p(x)}$$

which states that the probability that some instance x is from class A equals the probability of x given that the instance belongs to class A factored by the probability of class A, divided by the probability of instance x being observed at all. $P(x \mid A)$ represents a probability density function (PDF). There are many different PDFs, however the normal distribution is the most common; it is described by the mean and covariance of a dataset. An instance is assigned to a class that has the highest evaluation of $P(A \mid x)$ in a multiclass problem.

The distribution of bacilli in the captured images can be modelled quantitatively as a normal distribution. This assumption is due to the central limit theorem (the sum of identically distributed objects is approximately normally distributed if they have a finite variance), and becomes more accurate for an increasing number of objects. Therefore Bayes' classifier for given normal densities may perform well in classifying segmented objects as either bacilli or non-bacilli. Training of a Bayes' classifier constitutes estimating means and covariances of classes (from a training dataset) to form normal distributions.

2.5.4 Nearest Neighbour Classifier

The nearest neighbour (NN) classifier predicts the labels of query objects by comparing them to stored objects whose labels are known (Fukunaga, 1990). The comparison is the

Euclidean distance between each query point and the stored points. A query point is assigned the label of the stored point closest to it. If *k* points are used instead, the NN extends to the *k*NN classifier, and the query point is assigned the label of the majority of its neighbours. According to Greene (2001), the Thornton separability index (SI) can sufficiently determine the generalisation performance of the *k*NN classifier, and hence avoid evaluation by multiple data splits into train-test sets. SI can be used to determine the value of *k* to be used in the *k*NN classifier.

The *k*NN is known to be asymptotically optimal in the Bayes sense; however when the data points of the training dataset have variable characteristics over the space, the performance may become sub-optimal (Pekalska et al., 2006). To improve performance of proximity-based classifiers, the training points can be mapped to a dissimilarity space. The dissimilarity measure will be small for points in the same class and large for points from different classes, with objects that are similar in the new space lying close to each other; the resulting decision boundary can become overly complex. Different procedures can be used to reduce the number of objects in the dissimilarity space, for example random selection or an editing and condensing (EdiCon) algorithm. Pakalska et al. (2006) observed that random selection of objects had performance comparable to using feature selection algorithms to reduce the size of the dissimilarity space. The EdiCon algorithm removes the noisy objects, and then randomly chooses objects that have good performance on the nearest neighbour rule (Pekalska et al., 2006).

2.5.5 Linear Classifier

Classification using linear regression is faster than using the *k*NN classifier because the *k*NN classifier calculates the Euclidean distances to each stored point for each query point presented. On the other hand, the linear regression classifier finds a linear mapping between stored points and their labels, and uses the mapping to predict labels of query points. The mapping minimises the errors between classes in the least square sense, using the Euclidean distance. The logistic linear classifier is obtained by iteratively

reweighting the least squares solution to the plot of a line separating the two classes (Fukunaga, 1990).

2.5.6 Quadratic Classifier

The quadratic discriminant classifier is a density based classifier; the classes of the dataset are assumed to have normal density distributions. The classes are distinguished by their covariance matrices. For each class, the covariance matrix and the mean vector are used to estimate the density distribution. The decision function of the quadratic discriminant classifier is derived from the Bayes likelihood ratio, which is a quadratic function (Fukunaga, 1990):

$$\frac{1}{2}((X-M_1)^T\Sigma_1^{-1}(X-M_1))-\frac{1}{2}((X-M_2)^T\Sigma_2^{-1}(X-M_2))+\frac{1}{2}\ln\frac{|\Sigma_1|}{|\Sigma_2|} \gtrless \ln\frac{P_1}{P_2}$$

where X is an object feature vector, M is the mean vector, $\sum$ is the covariance matrix, and P_1 and P_2 are prior probabilities of the classes. The quadratic classifier assigns labels to objects based on the inequality; mean and covariance matrices are estimated from the training objects (Fukunaga, 1990).

2.5.7 Probabilistic Neural Networks

Artificial neural networks (ANN) are versatile learning algorithms for medical image segmentation and object detection (Nattkemper et al., 2003). Back-propagation (BP), a gradient descent algorithm, is a learning algorithm used by nonlinear multilayered perceptron (MLP) neural networks. It was used by Veropoulos (2001) for training ANN for classification of objects in sputum smear images. He also used its variation, the scaled conjugate gradient, based on the conjugate gradient optimisation technique.

However, single-layered radial basis function (RBF) networks are more inviting to use because they are easier to train than MLP neural networks.

RBF networks do not have the training complication of deciding on the number of hidden layers – they have a single layer (Duda et al., 2001). The single hidden layer is composed of radial basis functions, and Gaussians are frequently used. The output layer of a RBF network is a linear transformation of the outputs of radial basis functions; it can be optimised using linear techniques which are faster than MLP training techniques. Probabilistic neural networks (PNN) are RBF networks used for classification (Duda et al., 2001). PNN replaces each data point by a radial basis function (kernel) and uses these kernels to estimate the probability density functions of the training data. Kernels are centred on each of the training objects; they are added to estimate the distribution of each class. A query object assumes the label of the class whose PDF dominates at its position. For test objects far from the training objects, PNN assume the mean output level, while MLP extrapolates. Extrapolation is more likely to be wrong the further the test data is from the training objects.

PNN requires more neurons than BP networks, but can be designed in the fraction of the time it takes to train BP networks. The PNN classifier has one tuning parameter, the kernel width parameter, and leave-one-out (LOO) validation can be used to search for its optimum. LOO validation passes through data with *t* objects *t* times and omits one point each time. The remaining points are used to predict the label of the point left out; accuracy is the fraction of correct labels (Joachims, 1999). The PNN parameter is the spread of the Gaussians.

2.5.8 Support Vector Machines

Artificial neural networks are based on the empirical risk minimisation principle; empirical risk is described as the measured error rate on a training set with a fixed number of objects (Vapnik, 1998). According to Long et al. (2004) this makes ANN

vulnerable to sub-optimal solutions due to local minima in the optimisation function. Furthermore it makes their training complex as they are susceptible to training problems such as 'overfitting' or 'underfitting'. On the other hand, Support Vector Machines (SVM) are based on the structural risk minimisation principle, which seeks to minimise the empirical risk and additionally reduce capacity to a minimum suitable to describe the training data (Vapnik, 1998).

For a two-class linearly separable classification task, SVM aims to establish a hyperplane that correctly classifies all of the training vectors x_i, for i = 1, 2… N

$$g(x) = w^T x + w_o = 0$$

The hyperplane has to have the same distance to the nearest points in each of the two classes. For the non-separable case, the aim is to establish a hyperplane that produces a minimum number of misclassified objects from each class. This is a constrained optimisation problem; it can be solved using Lagrange multipliers.

In the dual Langrangian formulation, the training objects appear as pairs when computing the inner product of the optimisation (Theodoridis and Koutroumbas, 2003). The inner product of the two vectors in a higher dimensional space can be expressed as a function of the inner product of the corresponding vectors in the original space (Theodoridis and Koutroumbas, 2003). This equivalence is called Mercer's theorem; it states that the inner product can be equated to a kernel function. Common kernel functions are polynomial, hyperbolic tangent and radial basis functions.

The limitation of SVM is that solving the quadratic optimisation problem for large training data requires huge computer memory. Sequential Minimal Optimisation (SMO) is a common method used to solve the quadratic programming (QP) problem for large training sets (Platt, 1998). SMO solves the problem by breaking it into a series of smaller QP problems.

Training of SVM involves the search for the parameter C – a positive constant introduced to control the cost of misclassified objects – and the parameter of the kernel function using cross-validation. Leave-one-out cross-validation error can be estimated by finding the fraction of support vectors that are correctly classified (Joachims, 1999). Support vectors are training objects for which Langrange multipliers are positive.

2.5.9 Subspace Learning

Subspace learning is used when classification involves classes that have large intra-class variability. Unsupervised clustering is the first step of subspace learning. Clustering algorithms are used to cluster objects in each class to capture the intra-class variability. The number and the centres of clusters in each class are usually determined empirically. Each cluster or subspace is characterised by its cluster centre. Franco et al. (2006) use the method in face recognition to represent each individual by several lower dimensional clusters obtained by an unsupervised clustering of different poses; the input to the classification method is the dataset of all extracted features with dimensionality reduced by non-linear Fisher transforms. The approach may capture the intra-class variability of TB bacilli.

The clustering algorithm usually used to produce subspaces is the k-means clustering algorithm (Franco et al., 2006), which partitions the points in the data matrix into *k* clusters. Partitioning minimises the total, over all clusters, with-in cluster sums of point to cluster-centroid Euclidean distances.

The number of clusters in each class is determined empirically. K-means clustering is usually run several times with different initial positions for cluster centres to search for more compact clusters. Furthermore, initial positions should be relatively far apart so that most of the data points are inside the area obtained by joining cluster centres. This reduces the bias of the choice of initial cluster centres.

The silhouette value of an object is a measure of the similarity between the object and others in its own cluster, compared to objects in other clusters. Its value ranges from -1 to +1; -1 indicates that an object has probably been assigned to a wrong cluster and +1 indicates that an object has been assigned to the cluster that suits it the most. The average of the silhouette values can be used as a quantitative measure of cluster merit; a value between 0.7 and 1 indicates good clustering (Kaufman and Rousseeuw, 1990). Multiclass classifiers may have improved bacillus identification when they have as their classes the clusters derived from the training data.

2.5.10 One-class Classification

The detection of TB bacilli using one-class classifiers, introduced by Forero et al. (2006), is extended in this work. One-class classification, or novelty detection, is applied to cases where the class of objects that are not of interest – outliers – cannot be sufficiently modelled (Bishop, 1994) and is applied here because it is difficult to make an accurate ontology of the debris that may be present in a captured sputum smear image. Ypma et al. (1999) applied novelty detection to find faults in rotating mechanical machines; it is easy to obtain measurements for normal working conditions but difficult to measure readings from all possible failure conditions.

The Gaussian one-class classifier models the target objects as a Gaussian distribution, and objects with features falling below a threshold are labelled as outliers. The mixture of Gaussians (MoG) classifier uses a number of Gaussians to create a more robust description of the target class (Bishop, 1995). The principle components analysis (PCA) one-class classifier allows a choice of the eigenvectors of the target data covariance matrix to be used in describing the target data; removing high variance eigenvectors usually improves performance for data with low dimensionalities (Tax and Muller, 2003). MoG and Gaussian classifiers are density based; the PCA classifier is a reconstruction classifier – it reconstructs object parameters using low variance

eigenvectors, while the one-class *k*NN classifier describes the boundary of the target class (Tax, 2001).

Usually, the outliers of a one-class classifier cannot be sufficiently sampled, therefore their training involves setting a percentage error a classifier is allowed to make on the training target objects; the percentage error defines the number of target objects that may be misclassified as outlier objects. One-class classifiers form a closed decision boundary around the target data points. There is a trade-off between rejected target objects and accepted outliers. To compare performance of classifiers on a test set, the same allowable percentage of error is specified on the training target set for the different classifiers. The function derived using the set percentage of error on target objects is used to classify the test dataset.

2.5.11 Conclusion

Classification is the final step of the automated identification process. Supervised classification is usually used for identification, with unsupervised classification used to investigate the distribution of populations in defined classes. The classifiers introduced in this section are generally considered easy to train.

3 MATERIALS

The data used were images of Ziehl-Neelsen (ZN) stained sputum smear slides. The slides were prepared by the South African National Health Laboratory Services (NHLS) at Groote Schuur Hospital in Cape Town. The NHLS routinely prepares auramine stained sputum slides. The residual sputum was stained using the ZN method to accommodate this investigation. All slides used were positive for tuberculosis as confirmed by the NHLS. Where not enough sputum was left to prepare ZN-stained slides, the auramine stained slides were re-stained for the ZN investigation. The slides where not examined through cover slips. Sputum had been liquefied (digestion) and decontaminated.

A desktop computer was used to develop image processing and pattern recognition algorithms (3.20 GHz Intel Celeron processor; 0.99 GB of RAM). MATLAB 7.4.0 was used to develop all the algorithms. Prtools (Duin et al., 2004) and MATLAB Support Vector Machines (Cawley, 2000) toolboxes were used along with MATLAB. The data description toolbox (Tax, 2008) was used in one-class classification.

3.1 100x Microscope

A Nikon Microphot-FX microscope with a Kodak DC290 Zoom digital camera attached was used to obtain a dataset of images. The microscope was used at 100x magnification with an oil objective lens and numerical aperture 0.17. The microscope had a 10x ocular lens and a built-in set of ancillary magnification lenses - ‘Optivars’; a 1.25x Optivar was used. The microscope was manually focused to capture images, and white balance was done on an empty slide. Figure 3.1 shows an example 480 x 720 pixel image produced by the microscope. Ten to 100 images were typically taken of different fields in one slide.

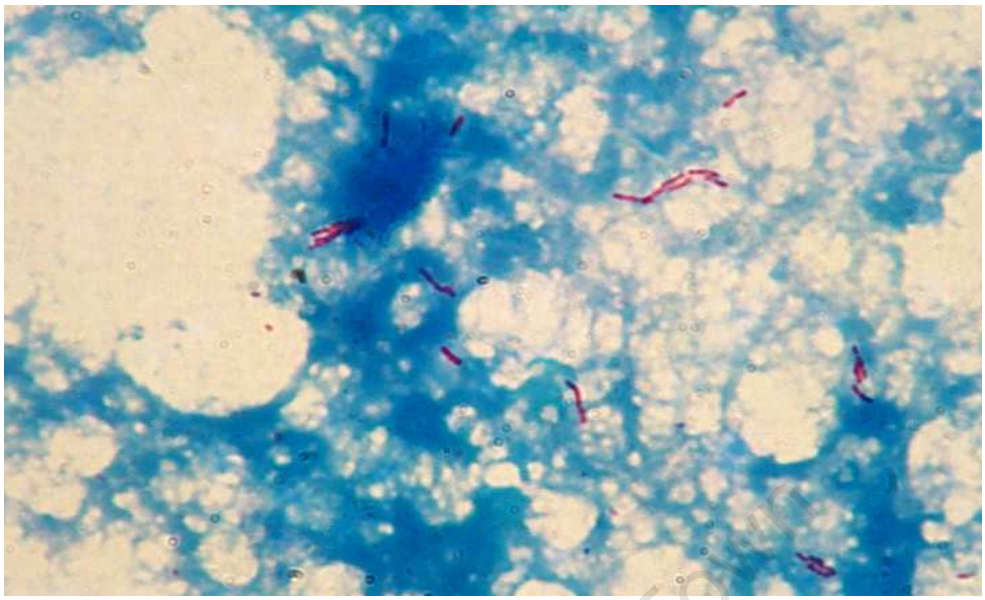

Figure 3.1: An image produced by the Nikon microscope at 100x magnification. Bacilli are red.

3.2 20x Microscope

A prototype microscope developed by Interscopic Analysis LLC (Cleveland, Ohio, USA) was used to capture images at 20x objective lens magnification. The microscope was manually focused. It had a numerical aperture of 0.45. The microscope uses white light from a light emitting diode to illuminate slides. The microscope is designed for image viewing on a computer monitor after image capture using an integrated digital camera; it has a 10x telan lens between the objective and the camera. The camera is connected to a computer by a USB connector. Figure 3.2 shows the setup for capturing digital images from a slide.

The digital camera of the microscope was connected to a Dell Latitude D610 notebook computer hence the monitor was the only display of the slide view field. Ten to 100 images were typically taken of different fields in one slide. Figure 3.3 shows an example image, 1024 x 1280 pixels, produced by the microscope.

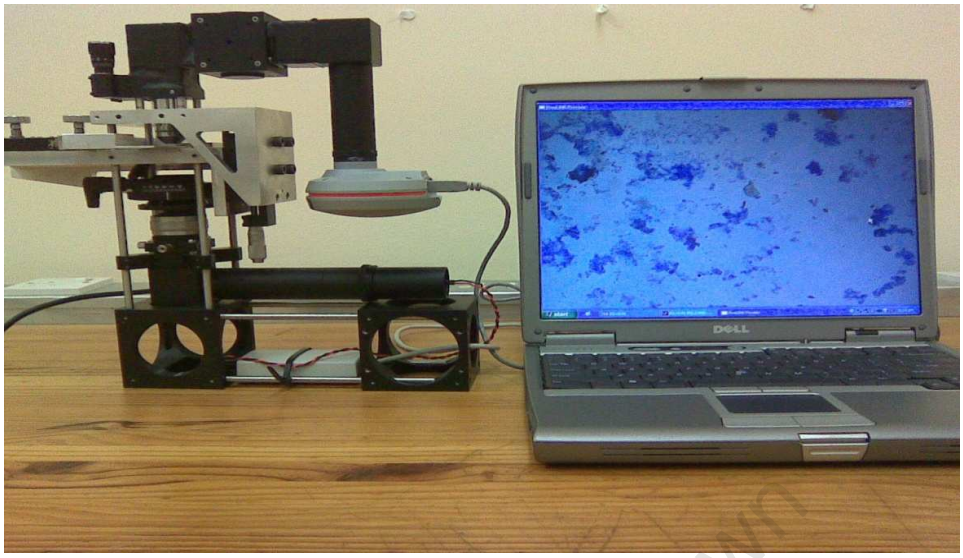

Figure 3.2: The setup for capturing a digital image with the Interscopic microscope.

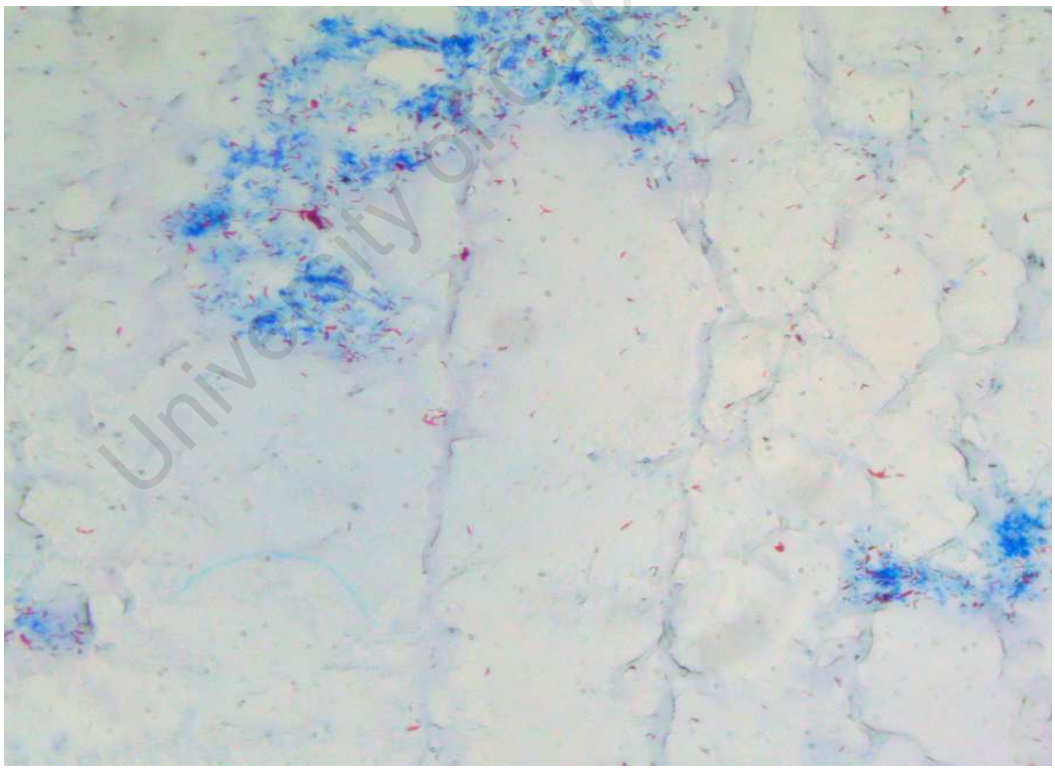

Figure 3.3: An example image produced by the Interscopic microscope at 20x magnification.

3.3 Discarded Images

Images with bacilli clumped together were discarded. Discarding such images did not limit the amount of bacilli available for the study because slides with clustered bacilli will also have bacilli spread across many view fields as they come from patients with a high bacillus load.

The redness of the tubercle bacillus corresponded to how much Ziehl-Neelsen carbol fuchsin stain the mycobacterium had absorbed. The slides were decolorized so that only the bacilli, with waxy coating, retained the colour red. Sometimes the acid alcohol used to decolorize the slide was not applied thoroughly, resulting in dark patches on the slide when the methylene blue counter-stain, which stains the background blue, was washed over the slide. Images from such slides were discarded. Figure 3.4 shows an example image not used for either training or testing classifiers.

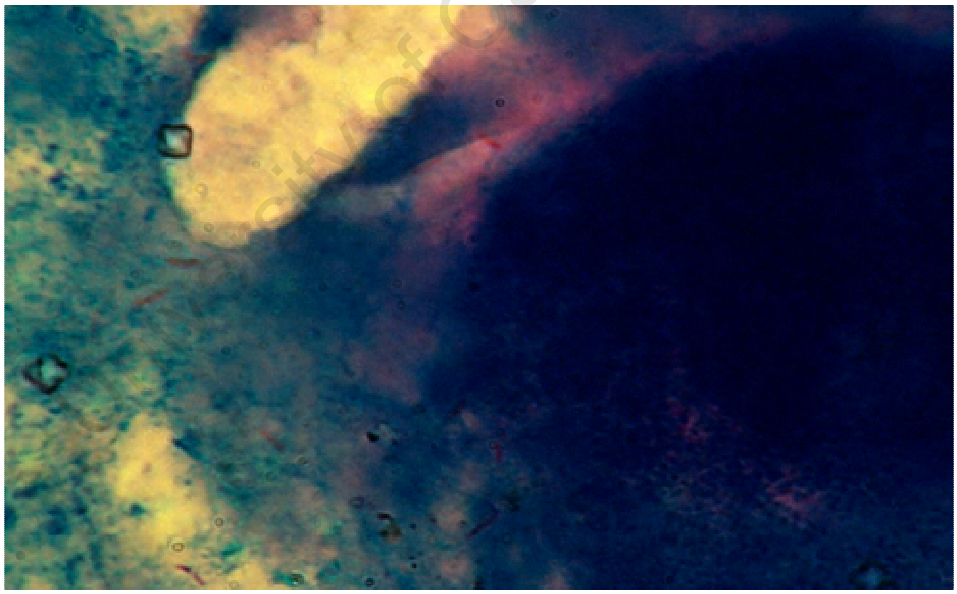

Figure 3.4: An example of an image not used in the study.

4 METHODS

This chapter details the methods that were used in the identification of bacilli. The image segmentation methods are described, followed by feature extraction and feature subset selection. The chapter ends with the procedures followed to classify image objects as either bacillus or non-bacillus objects. Below is the procedure for labelling image objects with the help of an expert pathologist.

4.1 Object Labelling

A machine vision algorithm is trained by exposing it to typical cases of the condition it is to learn. In the case of sputum smear images, typical cases are captured images of bacilli. Classifiers need to be trained with images containing objects labelled with the different classes. Objects in Ziehl-Neelsen stained sputum smear microscope images were labelled as bacilli or non-bacilli with the help of researchers at the Automated Tuberculosis Microscopy Facility of the University of Cape Town and an expert pathologist from the South African National Health Laboratory Services at Groote Schuur hospital.

These object labels were used for classifier training and served as the gold standard in assessing the performance of classifiers in identifying objects in test images. For pixel classifiers, individual pixels in an image were labelled, by manual outlining of the objects and automatic labelling of the pixels enclosed by the outlines.

To assist researchers in labelling objects in the microscope images, a graphical user interface (GUI) was developed and compiled. An image is loaded into the GUI. Perimeter pixels of a bacillus in an image are saved after an outline has been drawn around it. The GUI gives an option to record locations of bacilli by saving coordinates of mouse clicks on the image. A bacillus is regarded as a positive object. Debris is marked as negative objects. Debris is remnants of cells, bacilli destroyed by macrophages, or food particles. To simplify the learning process, only bacilli with their length in the focal

plane were considered as positive objects during the labelling. Bacilli that were clustered together into a clump were not considered for training. Figure 4.1 is a screen shot of the tool used for labelling.

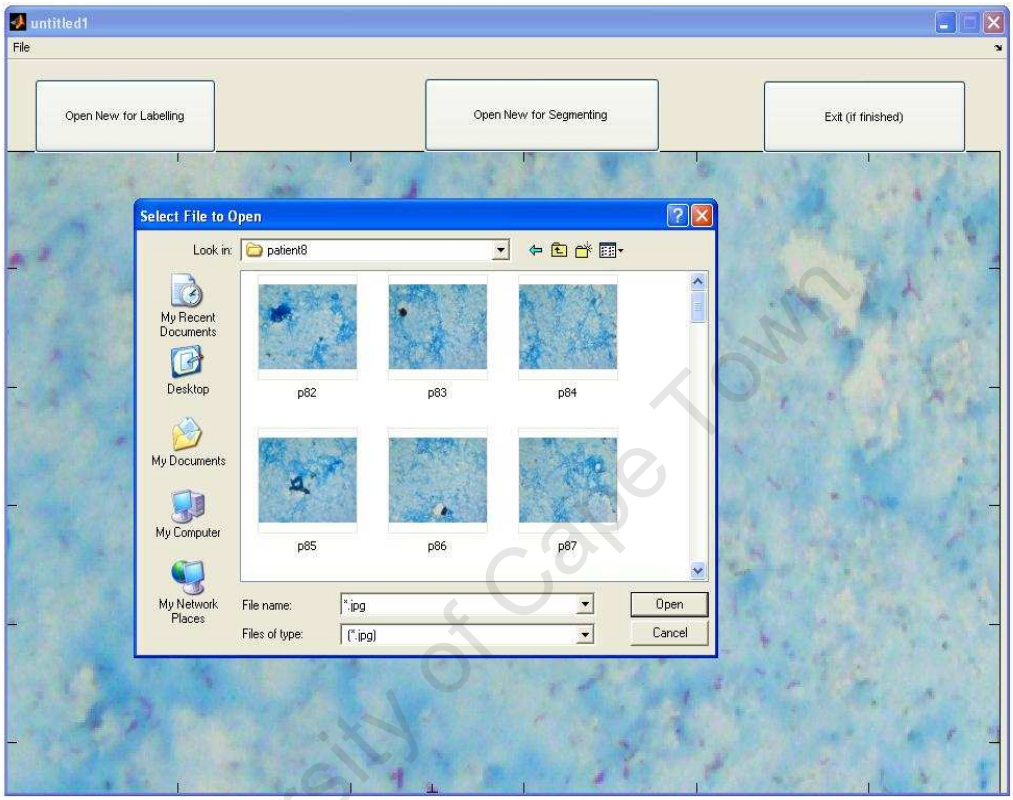

Figure 4.1: A screen shot of the GUI used for labelling.

4.2 Segmentation Methods

The fundamental image segmentation techniques include thresholding, boundary-based and region-based techniques, which are often combined. Speed and accuracy are the major factors determining which combinations are used. However, the most decisive factor in selecting segmentation methods is the nature of the images to be processed. For the segmentation of images of Ziehl-Neelsen stained sputum smears, the colour red

against a blue background and the size of the TB bacillus relative to the image size were important factors in considering the suitability of segmentation methods.

The image segmentation methods investigated were generally expected to withstand the variability of the shape of TB bacilli. The performance of the segmentation methods was not to change depending on the amount of *Mycobacterium tuberculosis* in an image. Ideally, all bacilli in the focal plane of the image should be picked up by the proposed segmentation technique. Marker-controlled watershed segmentation, a combination of pixel classifiers, and contextual pixel classification were explored, and are outlined below after a brief outline of the segmentation evaluation method.

4.2.1 Segmentation Evaluation

The accuracy of different segmentation methods was evaluated using a procedure proposed by Meurie et al. (2003) and described in Section 2.2.6. Images were manually segmented to provide a reference using the GUI described above. The GUI allowed the drawing of an outline around each bacillus which had its length in the focal plane.

4.2.2 Marker-Controlled Watershed Segmentation

The Ziehl-Neelson staining procedure stains bacilli red and background blue. On the electromagnetic spectrum, red and blue light occupy distinct positions that do not intersect. This prompted the application of the watershed method to segment bacilli in images. For a complemented image, bacilli form topological hills against the backdrop of valleys that correspond to the blue background. Marker-controlled watershed segmentation was performed using the following procedure:

1. The input image channel that showed bacilli the most clearly was filtered using a homomorphic filter (Sonka, 2008), which pronounces object outlines by

removing multiplicative noise; an image is filtered by a high-pass filter, which is damped so as not to suppress low frequency details.

2. The filtered image regions that were on average one pixel deeper than their surroundings were used as watershed internal markers.
3. The external markers of the watershed were found by finding the watershed lines of the distance transform (Gonzalez et al., 2004) of the internal markers.

4. The Sobel edge detector (Gonzalez et al., 2004) was used to find the gradient image of the channel showing bacilli the most clearly. The watershed lines were found on the gradient image with the imposed markers; the lines were used as outlines of segmented objects in an image.

i. Combination of Multiple Pixel Classifiers

Each pixel was classified by considering its values in the three channels of the colour space used. Logistic linear, Bayes', quadratic and Euclidean distance linear classifiers were used. The procedure for classifying a pixel by the combination of classifiers was as follows:

1. The classifiers were first trained. Bayes' classifier was trained by estimating the mean and covariance matrices of the two classes in the training dataset. The logistic and Euclidean linear classifiers are parameter-less; their training involved establishing the mapping that classifiers used to label objects of the training dataset. Lastly the quadratic classifier used the training dataset to find the covariance matrices of the two classes.

2. Classifiers were combined using different combination schemes and the segmentation performance of each combination was evaluated. The combination schemes used were the mean, median, minimum, maximum and the product of

classifiers' output posterior probabilities. All classifiers assigned two posterior probabilities to a pixel, one for each class. The mean combination scheme averaged posterior probabilities in each class; the median scheme found the median of the probabilities; the minimum and maximum schemes respectively found the minimum and maximum posterior probabilities; the product combination scheme substituted a vector of probabilities in each class by the product of its elements. A pixel assumed the label of the class with the highest probability.

3. Evaluation of the segmentation performance of a combination scheme was carried out using the procedure outlined in Section 4.2.1.

4.2.4 Contextual Pixel Classification using Canny Edge Detector

Segmentation using pixel classifiers can be improved by incorporating spatial information. Spatial information in the form of image object gradients was included in pixel classifier segmentation. The following procedure was used in an attempt to improve the segmentation results of the pixel classifier with the top performance as evaluated by the procedure outlined in Section 4.2.1:

1. Canny edge detection was used to find edges in the channel of the colour space used that showed bacilli the most clearly.

2. Canny edge detection results were overlaid on the pixel classifier results. Only objects detected by both methods formed segmented objects.

3. Segmentation results were obtained by combining the object outlines returned by the edge detector and the object interior pixels returned by the pixel classifier. At times the segmented objects had holes. Morphological closing and opening

(Gonzalez, 2004) were used to fill holes and remove spurs from objects, respectively.

4.3 Feature Extraction Methods

This section details the methods used for the extraction of features. The same number of features was extracted for all objects in the set of segmented images.

4.3.1 Extracted Features

Eccentricity and Compactness

Eccentricity and compactness of objects preserved by segmentation were extracted.

Fourier Features

Fourier features were extracted from the boundary of each object in a segmented image. The classification accuracy of the nearest neighbour classifier was used to determine the number of coefficients to use.

Moment Features

Moment invariants described by Mindru et al. (2004) were extracted for each object, using both the boundary and the interior pixels. For each object, moment invariants were calculated for each colour channel.

Colour Features

Colour features for each segmented object were extracted from the three colour channels. For each colour channel the value of the central pixel was considered as a

feature, and the means of the total and perimeter pixel values were two features. The last two colour features were the standard deviations of the total and perimeter pixels.

4.3.2 Data Normalisation

A dataset of objects was normalised. For each object, the mean value of each feature was subtracted from the feature value and the resulting feature value was divided by the standard deviation of that feature.

4.3.3 Linear Fisher Transform

Linear and non-linear Fisher transforms yielded the same mappings. Non-linear Fisher mapping addresses the problem of emphasizing large between-class distances. Since TB identification is a two-class problem, linear Fisher transforms were used.

4.4 Feature Subset Selection Methods

A feature subset selection method takes as input a dataset of all objects that were segmented, each with the same number of features extracted. The features of the dataset can either be original, normalised or mapped to a feature space that captures their maximum variability. Feature subset selection methods output the indices to features that constitute the best subset, to produce a dataset with a reduced number of features.

4.4.1 Population Based Incremental Learning

The following procedure was followed for feature subset selection using population based incremental learning (PBIL):

1. The number of generations during which the evaluations would run was set to 20.

2. The second input parameter was the number of trial masks to form for each generation; it was set to 50. Each trial mask was a vector of length equal to the number of features.

3. For each generation, trial masks consisting of 0's (corresponding to excluded features) and 1's (corresponding to included features) were randomly created. For each mask, the separability index (SI) of the data – features selected by the mask – was calculated. The mask corresponding to the best SI was kept. The mask that produced the highest SI in the last generation was used as the vector of indices to the best subset of features.

4. To calculate Thornton's separability index, the matrix of squared Euclidean distances from each point to all other points in the data set was found. The fraction of objects having another object with the same class number at the shortest distance to them was used as the separability index.

4.4.2 Correlation Based Feature Subset Selection

"Best first search" is the search strategy employed by the correlation based feature selection (CFS) method to search the feature space of a dataset. Each subset of features was evaluated by the CFS figure of merit. The procedure for selecting the best subset was as follows:

1. The correlation of each feature vector – feature values for all objects – with the vector containing class labels corresponding to each training object was calculated (the feature-class correlation).

2. For each feature, the correlations of the feature with the rest of the features were calculated (the feature-feature correlation). Both the feature-class and the feature-

feature correlations are required for the calculation of the evaluation figure of merit, as shown in Section 2.4.2.

3. The first element of the best subset of features was the feature with the highest correlation with the class labels vector. A feature that improved the evaluation figure of merit – using best first search – from the rest of the features was added to the subset. The additions of features stopped when the highest evaluation figure of merit of a feature from features not yet included in the subset, equalled that of the last addition of a feature to the subset of best features.

4.4.3 Sequential Floating Forward Selection

The nearest neighbour (NN) classifier was used to evaluate a subset of features for sequential floating forward feature selection. The procedure for selecting the best subset was as follows:

Beginning at an empty subset, features were added if they improved the evaluation– the NN accuracy. After each addition of a feature, the subset was assessed to see if removing any of the present features could improve the evaluation. The best subset of features selected was the subset corresponding to the highest evaluation value.

4.4.4 Branch and Bound Feature Subset Selection

The branch and bound feature selection method does not yield the number of features to select from a dataset; it yields the best ordering of features when given a desired number of features. The monotonic evaluation function used to follow the path with the highest evaluation was the nearest neighbour (NN) accuracy. Leave-one-out cross-validation was used to determine the NN accuracy for the selection of each of the features being ordered.

4.5 Object Classification Methods

Classifiers use training datasets to establish a mapping that classifies test objects into the classes represented in the training data. All classifiers were trained using datasets that were constructed to capture the variability of segmented objects from microscope images of sputum smears.

The generalisation performance of a classifier depends on the training procedure carried out to find the classifier's parameters. Most of the classifiers used in this study were trained using cross-validation, a training procedure that is suitable when limited training sets are available (Duda et al., 2001). If the classifier is presented with numerous training-test datasets, its accuracy is the average of the accuracies of all training-test datasets. Cross-validation uses one training-test dataset to evaluate the average accuracy of a classifier. For a dataset with 100 objects, the first 10 can be used for testing and the rest for training. Then the next 10 used for testing and the rest for training, and so on. This is called 10-fold cross-validation.

The classifiers were tested on data that was not used in the training process, to determine their accuracy. The classification method for each classifier is outlined below.

4.5.1 Bayes' Classifier

To classify test objects using the Bayes classifier, the classifier was trained and classification was done using the trained classifier following the procedure below:

1. The Bayes classifier was trained by finding the mean and covariance matrices, which describe a normal distribution.

2. The mapping determined in the training process was used to classify each test object into one of the classes of the training dataset.

4.5.2 Nearest Neighbour Classifier

The *k*NN classifier is the quickest to train but it is slow in classification because, to classify each object, it compares it to each of the training dataset objects.

1. Training a *k*NN classifier constitutes normalising the training dataset and finding the number of nearest neighbours from the training set to use when classifying test objects. The separability index is used to find the number of nearest neighbours to use.

2. For classification, the objects of the test dataset took the label of the training object closest to it, using the Euclidean distance measure.

4.5.3 Logistic Linear Classifier

The logistic linear classifier used the training dataset to learn a mapping that mapped training objects to their labels. The learnt mapping was used to classify test objects.

4.5.4 Quadratic Discriminant Classifier

Quadratic discriminant classifier used the covariance matrix and the mean vector to estimate the density distribution of each class. The density distributions were used to form the decision boundary as described in Section 2.5.6.

1. The training dataset was used to estimate the covariance matrices for the two classes.

2. The decision boundary formed from the two distributions was used to classify objects of a test dataset.

4.5.5 Probabilistic neural networks

Probabilistic neural networks (PNN) are radial basis function networks used for classification. The procedure for classifying a dataset of objects, each with the same number of features selected, was as follows:

1. First the PNN was trained using leave-one-out cross-validation to find sigma, the spread of the radial basis functions of the PNN classifier.

2. The sigma, together with the labelled dataset, was used to create a PNN network. A test dataset was classified by applying the PNN using the test objects as inputs.

4.5.6 Support Vector Machines

Support vector machines (SVM) are classification algorithms that have two parameters to be established during the training process – the parameter of the kernel function γ and the upper bound to the Langrangian multipliers C.

1. To find the C- γ pair to use for constructing the SVM classifier, the leave-one-out error for different C- γ pairs was obtained. The C- γ pair that corresponded to the lowest error was used. The leave-one-out error was estimated as the fraction of support vectors incorrectly classified.

2. The SVM classifier was used to classify objects of the test dataset.

4.5.7 Subspace Learning

The shape of *Mycobacterium tuberculosis* varies from curved to straight rods of length 1 – 10 µm (Forero et al., 2006). This prompted investigation of subspace learning and

classification to incorporate intra-class variability of bacilli. Subspace classification was carried out using the following procedure:

1. A linear Fisher transformation was performed on the training dataset.

2. The kmeans clustering algorithm was used to find clusters within each of the two classes using the Euclidean distance measure. The silhouette routine was used to inspect how good the clusters are; values range from -1 to +1. The composition was considered strong if the silhouette average was 0.7 – 1.

3. All classifiers were trained using the clustered datasets. All classifiers were evaluated by considering the fraction of test bacillus objects assigned to clusters of the non-bacilli class.

5 RESULTS: TWO-CLASS CLASSIFICATION AT 100x MAGNIFICATION

This chapter presents experimental results obtained using images from the microscope with the 100x objective. The results of segmentation, feature subset selection and classification are presented.

5.1 Segmentation Results

One of the main factors affecting classifier performance is how accurately training objects are defined. This calls for thorough evaluation of the segmentation methods used, since the segmentation stage isolates objects from which descriptive features are extracted for use in the classification stage. The procedure for the evaluation of image segmentation results proposed by Meurie et al. (2003) was used to rank segmentation methods objectively in order of performance. Marker-controlled watershed segmentation, a combination of multiple pixel classifiers, and contextual pixel classification using the Canny edge detector in conjunction with a pixel classifier were compared. The same test set of five images from different subjects was used to evaluate each segmentation method.

Each of the five images that constituted the test dataset had a manually segmented version. The manually segmented version was used to derive ratios of correctly and incorrectly segmented pixels. An image was manually segmented using the GUI described in Section 4.1.

5.1.1 Marker-Controlled Watershed Segmentation

The watershed algorithm was applied to the blue channel of the RGB colour space, as this channel showed the bacilli most clearly, by visual inspection. Table 5.1 shows the

ratios of correctly segmented pixels and incorrectly segmented pixels; the ratios were calculated as described in Section 2.2.6. The ratios of the five images were averaged.

Table 5.1: The ratios of correctly and incorrectly classified pixels resulting from watershed segmentation applied to the blue channel of the RGB colour spaces.

RGB(blue channel)			
Ratio of correctly classified pixels	0.3282	Ratio of incorrectly classified pixels	0.9926

Some of the objects extracted were too large to be bacilli; the high percentage of incorrectly classified pixels suggests that big patches of background were classified as bacillus pixels for all colour spaces. This prompted the use of a filter based on the area of an object. If an object was larger than a threshold, it was filtered out. The threshold was set above the typical area of a bacillus object, which was empirically found to be 50 pixels. Figure 5.1 shows a sub-image of the segmentation results using the watershed method, for the RGB colour space. The last sub-image shows the effect of using a filter with a threshold set at 400 pixels. In spite of the filtering, some inappropriately segmented background areas persist.

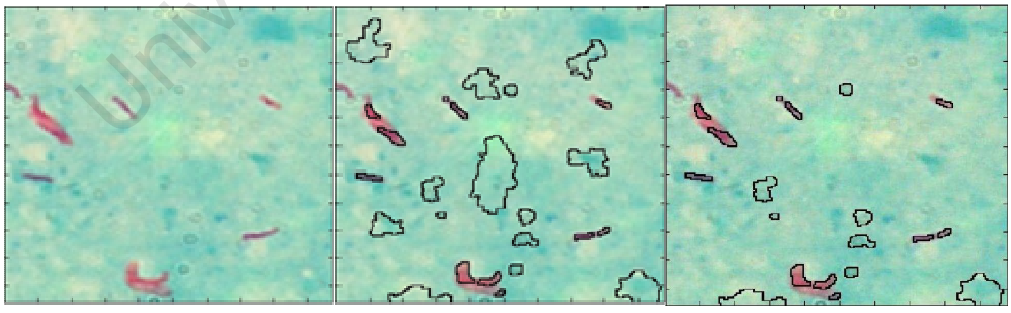

Figure 5.1: The original image (left), the results after marker-controlled watershed segmentation (middle) and the results after filtering based on object area (right).

5.1.2 Combination of Multiple Pixel Classifiers

The dataset of images used to train pixel classifiers was derived from nine subjects. Image pixels were used as objects. The dataset was composed of 28 images, from which pixels of bacilli in the focal plane were labelled as +1. A subset of background pixels was labelled as -1. The objects of the training dataset had three features – the pixel values of the three channels of a colour space. The dataset contained 40666 objects, of which 20637 were bacillus objects.

The training dataset was used to find the mean and covariance matrices used to estimate the distributions of the two classes of the Bayes' classifier. For the Euclidean distance linear classifier, the training dataset was used to find the minimum error in the least square sense. The logistic linear classifier used the sigmoid function to maximise the likelihood measure for assigning objects of the training dataset to their classes. Lastly the quadratic classifier used the training dataset to find the covariance matrices of the two classes.

As pixel classification relies on pixel colour, the colour space used in classification may influence the accuracy of the results. The performance of the classifiers on different colour spaces was examined, so as to select the colour space that produced the best results.

The classifiers were assessed using a dataset of five images from different subjects. The images each had a manually segmented version used to calculate the ratios of correctly and incorrectly classified pixels. Table 5.2 shows these ratios, which were used to rank pixel classifiers in order of segmentation accuracy, and to investigate which colour space is best suited for the pixel classifier segmentation method.

The RGB colour space was chosen as the input image colour space, as it had the best performance across different classifiers.

Table 5.2: For each classifier in each colour space, the first entry is the ratio of correctly classified pixels; the second entry is the ratio of incorrectly classified pixels.

Classifier	RGB	HSV	YCbCr	CIE-Lab
Logistic linear	0.8770 0.4259	0.8595 0.6886	0.8775 0.4255	0.8526 0.4657
Bayes	0.8839 0.3808	0.9017 0.5871	0.8835 0.3816	0.8779 0.4397
Quadratic	0.8839 0.3808	0.8879 0.5343	0.8462 0.3770	0.8653 0.4178
Euclidean distance linear	0.8573 0.3733	0.8549 0.6758	0.8581 0.3732	0.8332 0.4082
Average	0.8755 0.3902	0.8760 0.6215	0.8663 0.3893	0.8573 0.4329

Figure 5.2 shows the percentages of correctly classified pixels for different classifier combinations with increasing numbers of classifiers. Classifiers were added in decreasing order of performance, based on the results in Table 5.2. The combination schemes used were the product, mean, median, minimum and maximum of the pixel classifiers.

The product combination scheme performed best, using Bayes', quadratic and logistic linear classifiers, with the percentage of correctly classified pixels at 89.38%. For the first two classifiers, all combination schemes had the percentage of correctly classified pixels as 88.38%, and had the lowest percentage of incorrectly classified pixels, namely 38.08%; the Bayes' and quadratic classifiers each produced similar results individually, so combining them delivered no added benefit. However, the combination of three classifiers was chosen despite the small percentage improvement over a two-classifier combination or a single classifier, because segmentation is an intermediate stage in the identification process, and should be as accurate as possible, as the final identification is judged only on the classifier results. Figure 5.3 shows the segmentation results of the product of classifiers on three sub-images of the test dataset.

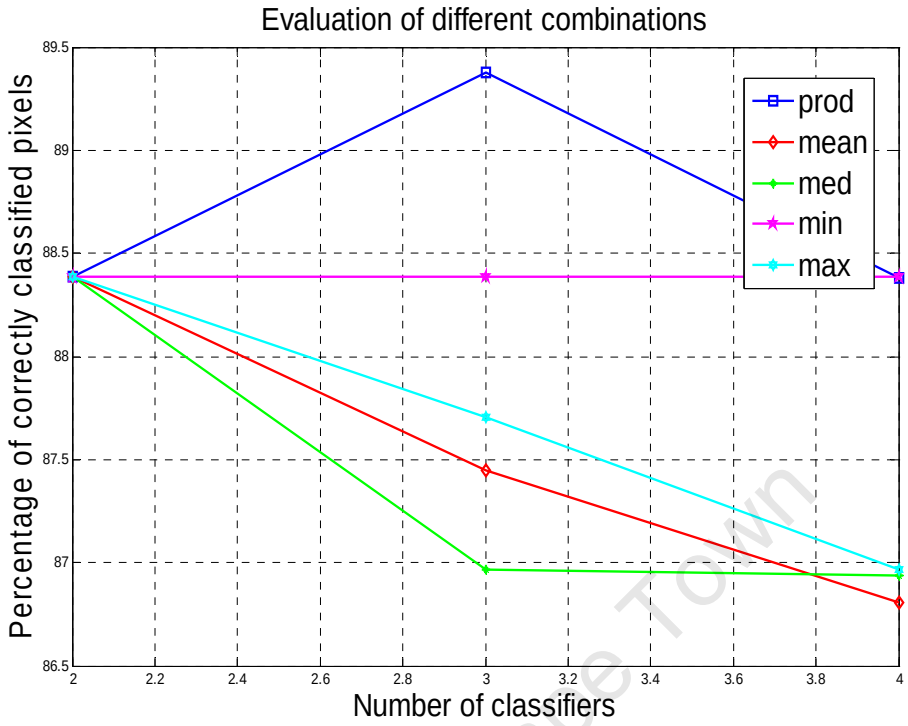

Figure 5.2: Percentages of correctly classified pixels for increasing number of combined classifiers.

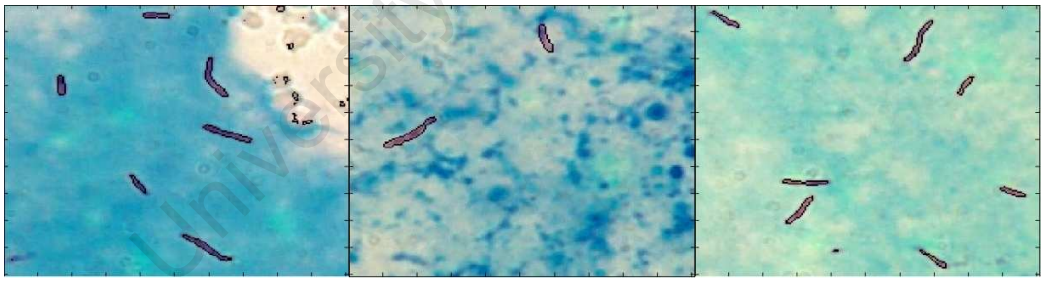

Figure 5.3: The outlines of segmented objects produced by the product of three classifiers overlaid on three sub-images of the test dataset.

5.1.3 Contextual Pixel Classification using Canny Edge Detector

Since the Bayes classifier had the highest percentage of correctly classified pixels, it was used in conjunction with the Canny edge detector to segment test images. The Canny

edge detector was applied to the blue channel of the RGB colour space, as the blue channel showed bacilli the most clearly, by visual inspection. The ratio of correctly classified pixels obtained from the dataset of five test images was 93.20%, and that of incorrectly classified pixels was 50.51%. Figure 5.4 shows some results of contextual pixel classification.

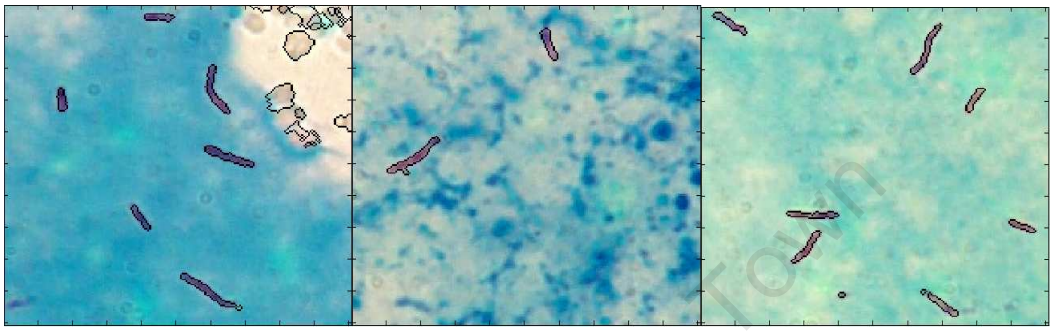

Figure 5.4: The outlines of segmented objects produced by the contextual pixel classifier overlaid on three sub-images of the test dataset.

5.1.4 Discussion on Segmentation

The product of the Bayes', quadratic and logistic linear pixel classifier was chosen as the segmentation method to be used in further steps, due to its superior performance. Contextual pixel classification using the Canny edge detection method had the highest percentage of correctly classified pixels. However, it also had a high percentage of incorrectly classified pixels, and this method was more likely to segment non-bacillus objects than the combination methods as can be observed when comparing the first sub-image of Figure 5.4 to that of Figure 5.3. The contextual pixel classification method using Canny edge detection uses the object gradient to include spatial information. The reason it produces more non-bacillus objects than the combination methods is that the gradient is an attribute of both bacillus and non-bacillus objects On the other hand, the combination methods use only the pixel colour information, which distinguishes bacilli

from other objects in the image, such as remnants of cells, bacilli destroyed by macrophages, or food particles. The Canny edge detector will pick up such objects.

No evaluation of sputum smear image segmentation was found in the literature for comparison with these results. The combination of pixel classifiers did not miss a bacillus object in the focal plane of an image. The biggest source of error for the segmentation method was that some slides had red stain deposits that had crystallised, possibly due to the delay in washing off the Ziehl-Neelsen carbol fuchsin with the acid alcohol used to decolorize the slide, as shown in Figure 5.5.

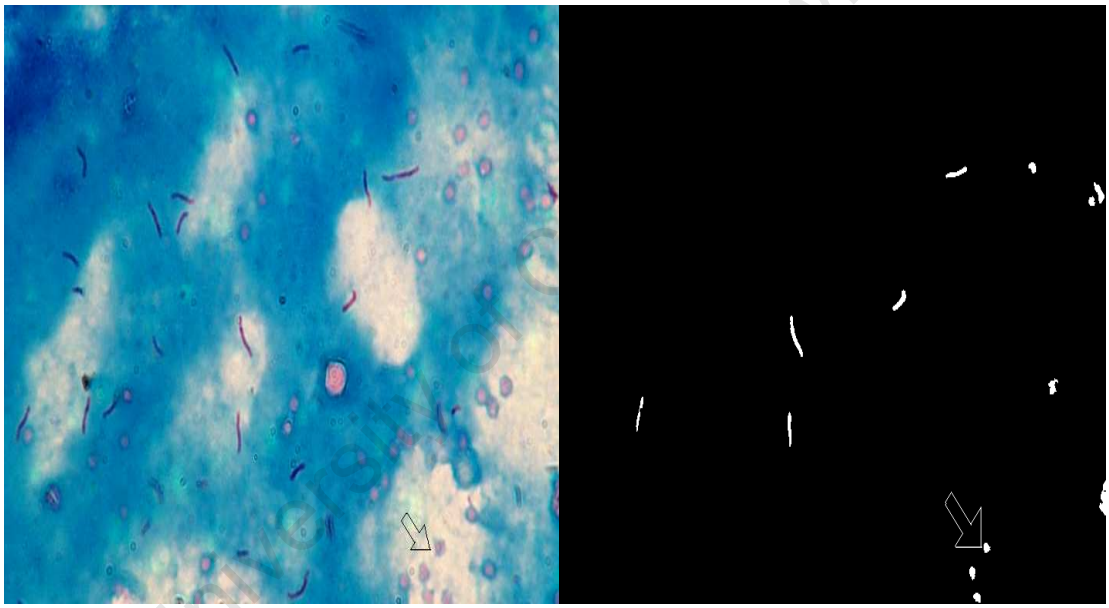

Figure 5.5: Results of the best combination of pixel classifiers on an image with red stains.

One way in which to eliminate this error could be automation of the staining procedure, which would further speed up TB screening. When assessing the performance of the pixel classifier, the merit criterion should incorporate an indication of how many bacillus objects in an image are picked up. This figure would influence the classification accuracy when determining performance of the overall identification system.

5.2 Feature Extraction and Feature Subset Selection

5.2.1 Filtering

The segmentation results of the best combination of pixel classifiers were filtered. The filter was based on the area and eccentricity of objects. If an object had an area less that 15 pixels or eccentricity less than 0.650, it was discarded. The remaining objects in the image were sent to the object classifier.

5.2.2 Feature Subset Selection

The feature extraction stage outputs a dataset made up of features associated with objects that resulted from the segmentation phase. All objects have equal numbers of features. The features of a dataset are normalised so that none biases classification results. Furthermore, features can be mapped to a feature space that de-correlates individual features. This section documents the results of selecting the best features to use for the object classification stage. The input to all feature selection methods is a dataset produced in the feature extraction stage.

The results of the feature subset selection methods performed on the normalised features are summarised in Table 5.3 below. The dataset used to investigate feature subset selection methods was drawn from three subjects, producing 185 images. The dataset constituted 1629 bacillus objects and 1697 non-bacillus objects.

The feature subset selection methods use different criteria for selecting the best features. This provides an avenue for a classifier to be tested using different feature sets. An evaluation figure of merit cannot be used to compare different selection techniques, but is used to select the best feature subset for a specific technique. The evaluation figures of merit used for the different selection technique are described in Section 2.4.2.

Table 5.3: Features selected using different selection methods.

Algorithm		PBIL	CFS	SFFS	B&B	
Evaluation figure of merit (%)		98.65	89.86	97.99	98.32	Key: PBIL = population based incremental learning CFS = correlation based feature selection SFFS = sequential floating forward selection B&B = branch and bound Ec = eccentricity Com = compactness Avg = average Std = standard deviation Ce = centre pixel A = inside object P = around object ● = selected ○ = not selected
Ec Com	1	●	●	●	●	
	2	●	●	○	●	
Fourier Features	3	○	○	○	●	
		●	○	○	●	
		●	○	○	●	
		○	○	●	○	
		●	●	●	●	
		●	○	○	○	
		●	●	●	●	
		●	○	●	○	
		●	●	●	●	
		●	○	○	○	
		●	●	●	●	
		○	○	●	○	
		○	○	○	●	
	16	●	○	●	●	
Moments	17	○	○	○	○	
		○	○	○	●	
		○	○	●	●	
	20	●	○	○	●	
Blue	Ce	○	●	○	○	
	Std P	○	○	○	○	
	A	●	●	●	●	
	Avg P	○	○	○	○	
	A	●	●	●	●	
Green	Ce	●	●	●	●	
	Std P	○	●	○	○	
	A	●	●	●	●	
	Avg P	○	○	○	○	
	A	●	●	●	●	
Red	Ce	●	●	●	●	
	Std P	○	○	○	○	
	A	●	○	●	○	
	Avg P	○	○	●	○	
	A	●	○	○	●	

5.3 Object Classification

The object classifiers used shape and colour information to classify segmented objects. The training dataset consisted of 6901 objects from eleven subjects. 4999 objects were labelled as bacilli and 1902 were labelled as non-bacilli. The fraction of objects in each class reflects the ratio of the number of objects in each class to the total number of objects yielded by the segmentation method. Therefore classifiers used the number of objects in each class as priors.

The logistic linear classifier used the sigmoid function to maximise the likelihood measure for objects of the training dataset to their classes. The Bayes classifier was trained by finding the mean and covariance matrices of the two classes. The quadratic classifier was trained by finding the covariance matrices of the two classes. Leave-one-out cross-validation was used to train the rest of the classifiers.

All classifiers were tested using a dataset with 1838 objects labelled as bacilli and 2520 objects labelled as non-bacilli, from a set of eight subjects separate from the training set. Figure 5.6 shows example bacillus and non-bacillus objects. The performance of different classifiers on feature subsets obtained using different feature selection procedures, all evaluated at their best operating point as found by cross-validation, is shown in Table 5.4.

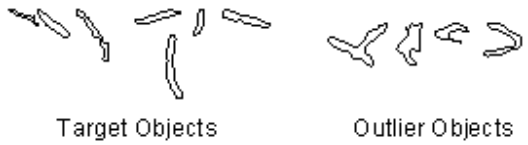

Figure 5.6: Example bacillus and non-bacillus objects.

Table 5.4: Evaluation of each classifier at its best point as found by cross-validation.

Selection of Features	Classifier / Accuracy (%) / Sensitivity (%) / Specificity (%)					
	*k*NN	Bayes'	Linear	Quadratic	PNN	SVM
PBIL	88.9628 99.1295 81.5476	68.2423 99.6736 45.3175	83.2263 99.8912 71.0714	76.9619 100 60.1587	85.0161 99.5103 74.4444	79.3254 99.1295 64.8810
Branch and Bound	84.4654 99.6736 73.3730	66.8196 99.4559 43.0159	82.3084 99.7280 69.6032	76.0670 99.9456 58.6508	82.3543 99.6192 69.7619	77.5585 99.2927 61.7063
Correlation based Selection	84.3736 99.5647 73.2937	87.9302 99.6736 79.3651	83.2263 100 70.9921	86.8518 99.5103 77.6190	82.4231 99.4015 70.0397	80.5645 99.8368 66.5079
Sequential floating forward	87.1960 99.6192 78.1349	89.0087 99.6192 81.2698	83.2492 99.9456 71.0714	75.6540 100 57.8968	84.8325 99.2927 74.2857	80.4497 99.5103 66.5476

Dissimilarity space reduction using two different methods (random and editing and condensing - EdiCon) was performed for the *k*NN classifier on all features. The *k*NN classifier uses nearness of a test object to training set objects to classify test objects. The results are shown in Table 5.5.

Table 5.5: Dissimilarity space reduction applied for the *k*NN classifier.

*k*NN classifier	Classifier / Accuracy (%) / Sensitivity (%) / Specificity (%)	
	Random	EdiCon
	91.5558 99.8368 85.5159	89.1005 99.9456 81.1905

Classifier training and testing were performed on Fisher mapped feature sets – Fisher mapping was applied on the set of all features. Subspace learning was performed on Fisher transformed data points and these results were also classified. K-means clustering

was used to cluster objects of each class. Starting at two clusters, the silhouette average was used to find the number of clusters to use, based on the training set. Clustering results for different cluster numbers were compared and the number of clusters that corresponded to the highest silhouette average was chosen. The bacillus class had an average of silhouette values of 0.6857 and the non-bacillus class had 0.7251; Figure 5.7 shows the clusters found in each class; each object is put in its cluster. Table 5.6 compares the performance of classifiers on the Fisher mapped feature set only to that on the clustered classes, for the test set.

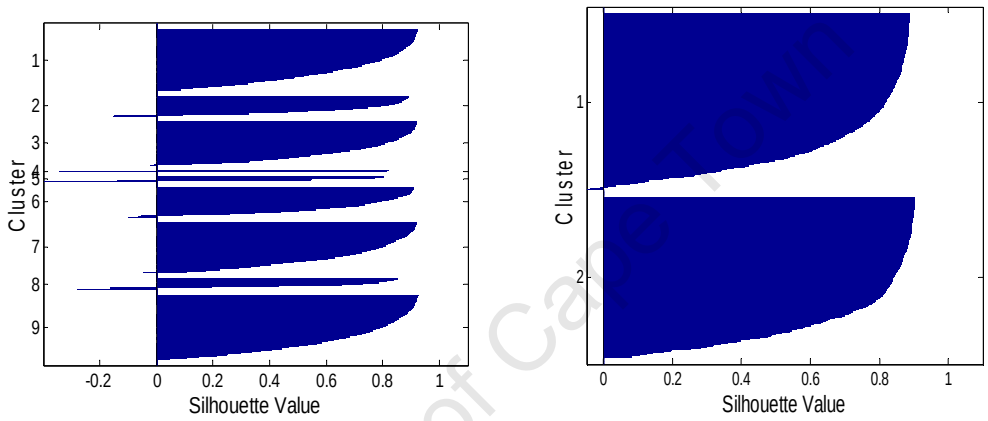

Figure 5.7: Silhouette plots of the bacillus (left) and non-bacillus (right) classes derived from the training set.

Table 5.6: Evaluation of classifiers after linear Fisher transformation and subspace learning.

	Classifier / Accuracy (%) / Sensitivity (%) / Specificity (%)					
	*k*NN	Bayes'	Linear	Quadratic	PNN	SVM
Linear Fisher mapping	98.5544 97.7693 99.1270	97.6595 95.3210 99.3651	98.5085 97.6605 99.1270	97.6595 95.3210 99.3651	98.5314 97.7149 99.1270	98.5544 97.7693 99.1270
9 bacillus and 2 non-bacillus classes with linear Fisher mapping	98.5314 97.7149 99.1270	74.1166 98.4222 56.3889	96.9481 93.5256 99.4444	98.7609 98.3678 99.0476	98.3249 97.2252 99.1270	98.5314 97.7149 99.1270

ROC curves were plotted for classifiers with linear Fisher transformed feature sets as inputs, as this method produced the best results. Figure 5.8 shows the ROC curves, the area under the ROC curves was measured for all classifiers – shown in Table 5.7.

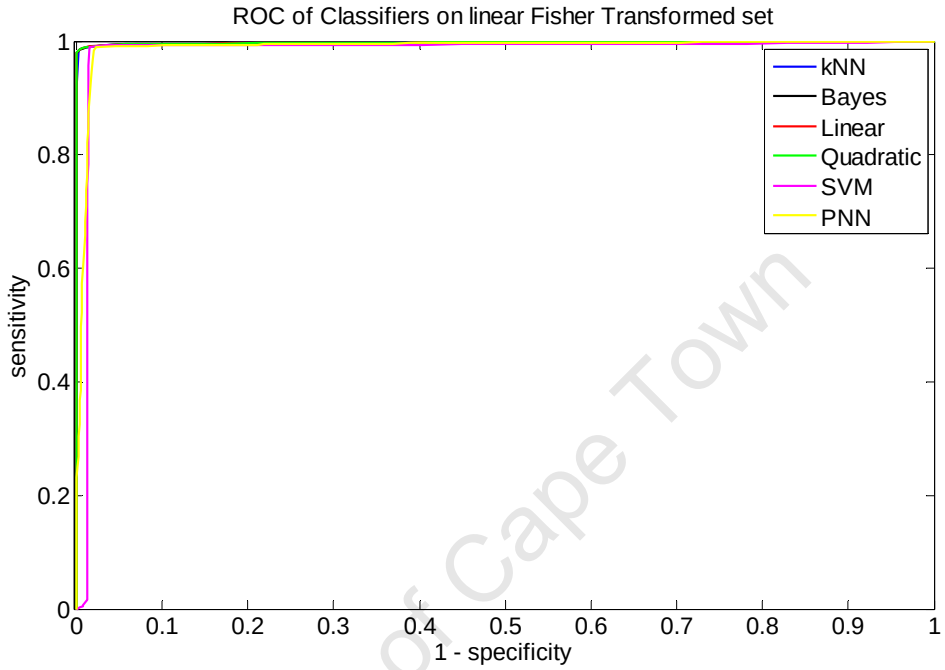

Figure 5.8: ROC curves for all classifiers.

Table 5.7: Evaluation of classifiers applied after Fisher mapping, by means of the area under the ROC curve.

	Classifier					
	*k*NN	Bayes'	Linear	Quadratic	PNN	SVM
Area under the ROC curve	0.9975	0.9975	0.9981	0.9981	0.9814	0.9895

Discussion

The performance of all classifiers was highest when Fisher mapping was applied to the feature set. All classifiers had balanced performance on the Fisher-mapped dataset, with sensitivity and specificity above 95% for the detection of an individual object. The Bayes and quadratic classifiers had the lowest accuracy of 97.67%. Veropoulos et al. (1999) achieved sensitivity of 94.1% and specificity of 97.4% in the classification of objects in auramine-stained sputum smear images (63x magnification) using a feed-forward neural network with four hidden units. Forero et al. (2006) obtained sensitivity of 97.89% and specificity of 94.67% in the classification of images of auramine stained sputum smears (25x magnification). A direct comparison of bacillus detection on auramine- and ZN-stained smears is not possible, but our comparable accuracies are encouraging.

The identification route that gives the best performance uses the product of three pixel classifiers as the segmentation method. Extracted features are transformed using linear Fisher mapping. The logistic linear and the quadratic classifiers performed best judging by the area under the ROC curve for different classifiers – Table 5.7. The *k*NN, linear, PNN and SVM classifiers, however, had higher sensitivity, an important consideration in TB screening. The good performance of the linear classifier may indicate linear separability of the two classes. The product of pixel classifiers followed by the linear classifier took 53.39 seconds to label objects in the full version of the first sub-image in Figure 5.4.

In Table 5.4, the *k*NN classifier performed better in terms of the overall accuracy compared to the other classifiers, since all feature selection algorithms use a nearest-neighbour based evaluation function - except the CFS algorithm which doesn't follow the same evaluation. Dissimilarity space reduction improved the *k*NN classification results.

The use of pixel classifiers dictated what priors to use as inputs to object classification. Pixel classifiers introduced high-level operations into the image processing step; consequently segmented images had fewer non-bacillus objects than would have been produced by conventional low-level image processing techniques. This was the motivation for using as priors the number of objects in each of the two classes defined (bacillus and non-bacillus). Objects presented to the classifiers for the final identification step were more likely to be bacilli than non-bacilli. Images containing lots of debris or objects out of focus or which were poorly stained were likely to produce more non-bacillus objects than bacillus-objects, but such images were not used for training and testing the classifiers.

An improvement in the feature-based classifiers may be achieved by implementing an object filter that operates on features. Objects may be rejected based on the distribution of each feature. Objects with feature values above a threshold would be declared bacillus objects right away and only objects with feature values falling within a band sent to the final classification stage.

Future work should include the search for the most descriptive feature, as simple classifiers may be expected to have good performance with it. Furthermore, there is a need for a feature that will describe touching bacilli, which usually form a *T* shape. Figure 5.9 illustrates the touching bacilli that may be captured with such a feature. The current scheme labels touching bacilli as non-bacilli.

In the automated TB screening application, an image will be captured by a digital camera. Then the best identification route will be taken to detect bacilli in that image. The WHO recommends that a subject be declared TB-negative if there are no bacilli seen in 100 high-power microscope view fields (WHO, 2003). Therefore, the microscope will be directed to capture the next image if no bacilli are detected in the present image.

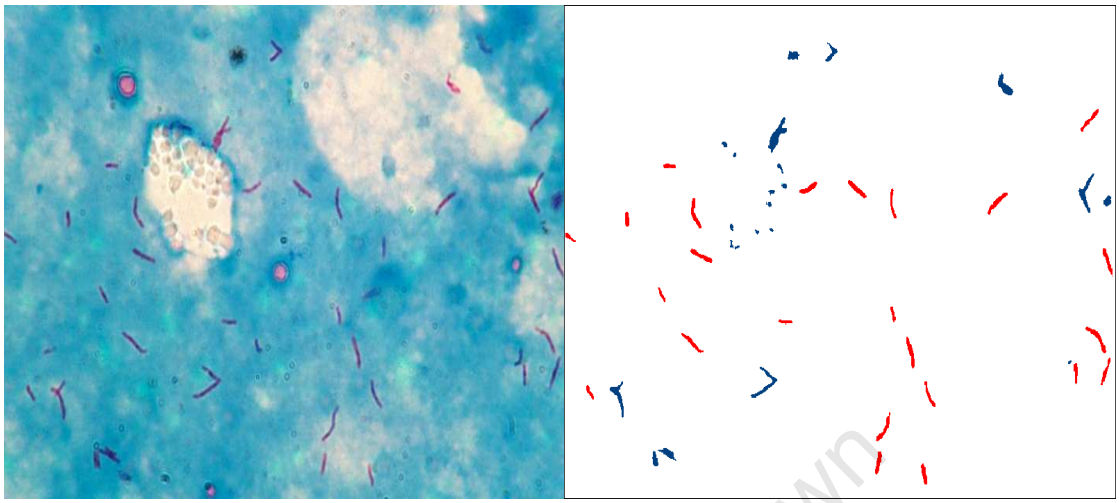

Figure 5.9: The results of the established identification route (using logistic linear classifier) on an example image; bacillus objects are in red and non-bacillus objects in blue in the right image; touching bacilli are labelled as non-bacilli.

6 RESULTS : TWO-CLASS CLASSIFICATION AT 20x MAGNIFICATION

The identification route established in Chapter 5 (the product of three pixel classifiers for segmentation, followed by the logistic linear object classifier) was applied to detect TB bacilli from images obtained using a 20x microscope; the results are presented in this chapter.

6.1 Segmentation Results

The dataset of images used to train pixel classifiers was obtained from four subjects. The dataset was composed of 26 images. The dataset contained 17907 objects; each object was represented by the three pixel values of a colour space. 8672 objects of the dataset were bacillus objects, the rest were non-bacillus objects.

The classifiers were assessed using a dataset of five images. Table 6.1 shows the ratios of correctly and incorrectly classified pixels. The ratios were used to rank pixel classifiers in order of segmentation accuracy, and to investigate which colour space is best suited for the pixel classifier segmentation method.

Figure 6.1 shows the percentage of correctly classified pixels for different combinations of pixel classifiers with increasing numbers of classifiers. The RGB colour space was used. The classifiers were added as ranked in Table 6.1. The combination schemes used were the product, mean, median, minimum and maximum of the pixel classifiers.

Table 6.1: For each classifier in each colour space, the first entry is the ratio of correctly classified pixels; the second entry is the ratio of incorrectly classified pixels.

Classifier	RGB	HSV	YCbCr	CIE-Lab
Logistic linear	0.5823 0.8410	0.5652 0.9473	0.5823 0.8411	0.5009 0.9443
Bayes	0.5150 0.7408	0.5342 0.9464	0.5143 0.7397	0.4081 0.6865
Quadratic	0.5150 0.7408	0.5126 0.7875	0.5143 0.7397	0.4081 0.6865
Euclidean distance linear	0.4132 0.7826	0.6362 0.9708	0.4098 0.7863	0.3470 0.7597
Average	0.5064 0.7763	0.5621 0.9130	0.5052 0.7767	0.4160 0.7693

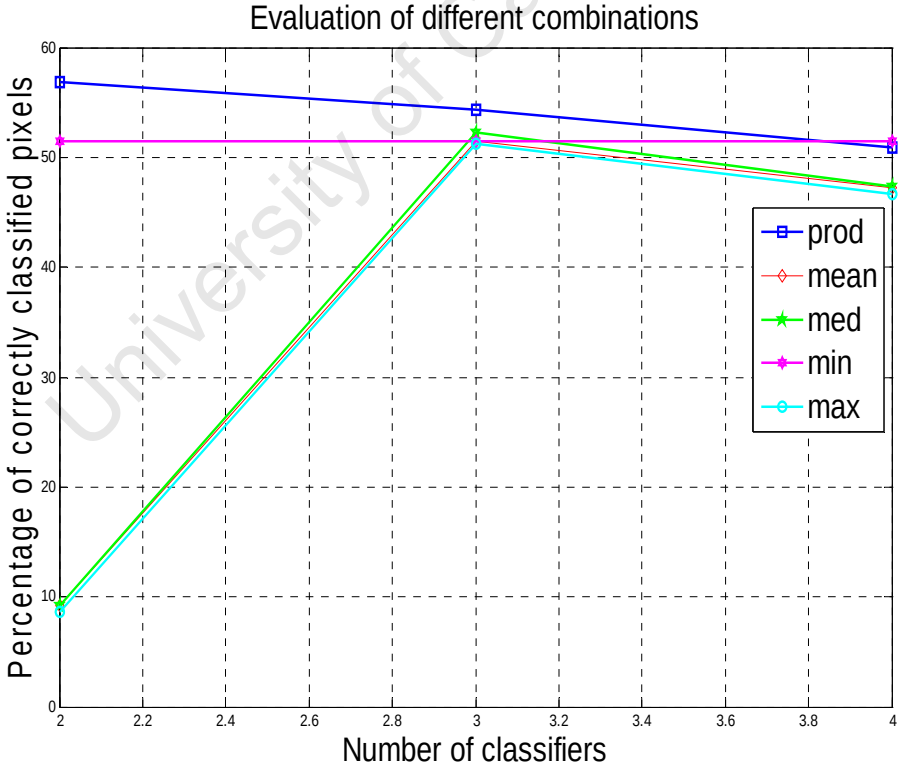

Figure 6.1: Percentages of correctly classified pixels for increasing number of combined classifiers.

The product combination scheme performed best, with logistic linear and Bayes' classifiers; the percentage of correctly classified pixels was 56.89% and that of incorrectly classified pixels 78.84%. Figure 6.2 shows the results of the product of classifiers on three sub-images of the test dataset.

Figure 6.2 shows typical results of the combination of pixel classifiers chosen as the main segmentation method. The method produced many small objects that corresponded mainly to bacilli that were out of focus. All segmented images were filtered based on object area. The filter threshold was set to 5 pixels; larger objects were retained.

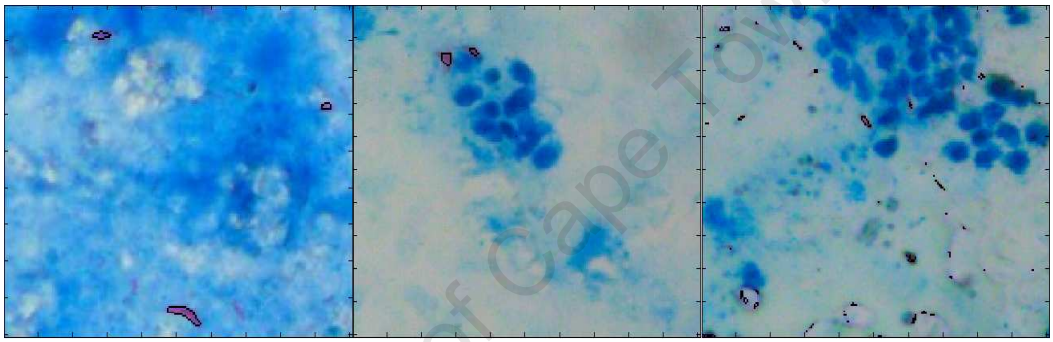

Figure 6.2: Sub-images of the product of classifiers results overlaid on the first three images of the test dataset.

6.2 Object Classification

The training dataset consisted of 1371 objects from seven subjects. 1000 objects were labelled as bacilli and 371 were labelled as non-bacilli. For all classifiers, the numbers in each class were used as the priors of such classes. All classifiers were tested using a dataset that had 578 objects labelled as bacilli and 153 objects labelled as non-bacilli. Classification was performed on a Fisher transformed feature set; the mapping was applied on all extracted features. Table 6.2 shows the performance of classifiers on the Fisher mapped feature set.

Table 6.2: Evaluation of classifiers on the Fisher mapped feature set.

	Classifier / Accuracy (%) / Sensitivity (%) / Specificity (%)					
	*k*NN	Bayes'	Linear	Quadratic	PNN	SVM
Linear	97.67	97.40	97.53	97.40	97.40	97.67
fisher	99.13	99.48	99.30	99.48	98.96	99.48
mapping	92.15	89.54	90.84	89.54	91.50	90.84

6.3 Discussion

Images captured at 20x microscope magnification have a larger view field than those captured at 100x magnification. A big view field translates to a short time to scan the whole sputum slide and a larger number of bacilli per view field. However, the poor segmentation is a severe limitation.

It is difficult for a technician to label individual bacillus objects at 20x magnification and thus to provide a gold standard against which to compare the segmentation algorithm.. At low magnification, the relative distance between objects is small. This makes the distinction of individual objects difficult. Furthermore, smaller bacilli are not picked up by the microscope.

The ability of the segmentation algorithm to distinguish the red colour of bacillus objects is affected by the focus of the microscope. Focus is difficult to attain in a large view field. The loss of focus leads to loss of the contrast of the colour red against blue background. The images lose sharpness and resolution deteriorates. The percentage of correctly classified pixels of 56.89% reflects not only the loss of object outlines but the total loss of some objects. The segmentation method using the product of the logistic linear and Bayes' pixel classifiers had a percentage of incorrectly classified pixels as 78.84%, which shows that the distinction between bacillus and non-bacillus pixels is difficult.

Object classification was performed on the objects extracted by the segmentation method. The classification results are good; the classifiers were able to differentiate bacillus objects from the specks picked up by the segmentation method. Linear Fisher mapping increased the difference between the two classes.

7 RESULTS: ONE-CLASS CLASSIFICATION AT 100x MAGNIFICATION

This chapter presents the results of density based, boundary based and reconstruction classifiers (Tax, 2001) for one-class classification. Images from the 100x microscope were used.

In ZN-stained sputum smear images, acid-fast bacilli are red against a blue background. Bacilli have a waxy coating which absorbs the red of the Ziehl-Neelsen carbol fuchsin; the background is stained blue by the methylene blue counter-stain. This prompted the formulation of the detection of TB as a novelty detection problem, as the colour red may be associated only with bacilli. Remnants of cells, bacilli destroyed by macrophages, or food particles are outlier objects. The objects in the images are identified using two stages of classification. The aim is to identify individual bacillus – target – objects with their length in the focal plane of the image. Two stages of classification are used, as in the two-class classification applications described in Chapters 5 and 6. The first stage (segmentation) uses colour information and pixel values are used as features by classifiers. The second stage of classification (object classification) uses shape information.

7.1 Segmentation

The dataset of images used to train pixel classifiers was derived from nine subjects and was composed of 28 images, as it was for two-class classification; pixels of bacilli in the focal plane were labelled as target objects. A subset of background pixels was labelled as outliers. The objects of the training dataset had three features – pixel values of the three channels of an RGB colour space. The dataset contained 40666 objects, of which 20637 were bacillus objects.

The classifiers were assessed using the dataset of five images from different subjects used for two-class pixel classifiers. The mixture of Gaussians (MoG), Gaussian and PCA

one-class classifiers were used; the false rejection they were allowed to make on target training objects was set at 20%. The images each had a manually segmented version used to calculate the correctly and incorrectly classified pixel ratios, which are shown in Table 7.1. The ratios were used to rank pixel classifiers in order of performance. Figure 7.1 shows example segmentation results.

Table 7.1: Performance evaluation of pixel classifiers.

	Gaussian	Mixture of Gaussians	PCA
Ratio of correctly classified pixels	0.5056	0.7574	0.6444
Ratio of incorrectly classified pixels	0.4337	0.3454	0.6882

Figure 7.1: The results obtained using the mixture of Gaussians classifier overlaid on two sub-images of the test dataset.

The mixture of Gaussians performed best in pixel classification. It showed the lowest ratio of incorrectly classified pixels, which translates into few outlier pixels classified as bacilli. It picks up most of the bacilli with their length in the focal plane of an image; the relatively low percentage of correctly classified pixels – 75.74% – was mainly due to inaccuracies in detecting object outlines.

The mixture of Gaussians is a density based classifier. The dataset had three features; therefore density estimation was less complicated due to low dimensionality.

7.2 Object Classification

Objects output by the first stage of identification were filtered based on their area; the threshold was set at a minimum of 50 pixels and a maximum of 400 pixels. The classification accuracy of the nearest neighbour classifier was used to determine the number of Fourier coefficients to use – 14. Moment invariants, and eccentricity and compactness were also used to describe the shape of objects. Bacillus objects with their length in the focal plane of the image were the target objects, and the rest were outlier objects. The training dataset consisted of 4376 objects from eleven subjects; the same images were used as in two-class classification, but different objects were extracted by the different segmentation procedure. 2728 objects were labelled as target and 1648 were labelled outlier.

MoG, Gaussian, PCA and kNN classifiers were used; the false rejection they were allowed to make on the set of target training objects was 5%. All classifiers were tested using the same dataset from eight subjects as used in two-class classification, with 1064 objects labelled as target objects and 1157 objects labelled as outliers. Table 7.2 shows the performance of different classifiers on different features. The area under the ROC curve for classifiers is shown in Table 7.3; this is a robust error measure for one-class classification (Metz, 1978).

Table 7.2: Evaluation of classifiers using different feature sets.

Classifier	Classifier Accuracy (%) Sensitivity (%) Specificity (%)			
	Gaussian	MoG	PCA	*k*NN
Fourier features	73.75 90.32 58.51	84.33 90.51 78.65	71.68 92.29 52.72	71.41 92.20 52.29
Moment invariants	51.91 94.55 12.71	50.88 94.36 10.89	50.47 95.49 9.08	50.74 94.83 10.20
Eccentricity and compactness	85.59 94.36 77.53	90.18 98.21 82.80	90.81 98.59 83.66	87.12 93.99 80.81

Table 7.3: Evaluation of classifiers using area under the ROC curve.

Classifier	Area under the ROC curve			
	Gaussian	Mixture of Gaussians	PCA	*k*NN
Fourier features	0.8152	0.9232	0.7583	0.7933
Moment invariants	0.5944	0.5766	0.5931	0.5437
Eccentricity compactness	0.9044	0.9509	0.9154	0.9065

Table 7.4 shows the performance of classifiers on the full set of extracted shape features and on the linear Fisher mapped feature set; Table 7.5 shows classifiers evaluated using the area under the ROC curve.

Table 7.4: Evaluation of classifiers using the set of all features.

	Classifier Accuracy (%) Sensitivity (%) Specificity (%)			
	Gaussian	Mixture of Gaussians	PCA	*k*NN
Set of all extracted features	85.59 91.07 80.55	93.47 90.88 95.85	81.00 90.88 71.91	78.12 94.08 63.44
Linear Fisher mapping	90.63 84.40 96.37	89.78 81.49 97.41	47.91 100 0	90.68 84.40 96.46

Table 7.5: Evaluation of classifiers by means of the area under the ROC curve.

Classifier	Area under the ROC curve			
	Gaussian	Mixture of Gaussians	PCA	*k*NN
Set of all extracted features	0.9424	0.9810	0.9069	0.8934
Linear Fisher mapping	0.9759	0.9801	0.5000	0.9748

7.3 Discussion

The mixture of Gaussians classifier performed best in the second stage of classification, using all features. Among the different feature sets, using eccentricity and compactness alone produced the highest accuracy for all classifiers (Table 7.2); the addition of Fourier features and moments increased specificity and reduced sensitivity for the Gaussian and mixture of Gaussian classifiers, and reduced overall performance for the PCA and *k*NN classifiers. The PCA classifier performed poorly on the linear Fisher mapped test set because it requires variance of features, which is removed by Fisher

mapping. Fisher mapping improved specificity but reduced sensitivity for the other classifiers.

An improvement in identification may be attained by implementing an object filter based on features. Object rejection can be increased by studying the distribution of each feature and setting thresholds on each. Objects with feature values above a threshold can be declared bacillus objects right away, and only objects within a band sent to the final classification stage.

Classifying only objects in a band might require optimisation of the decision boundary, for instance by careful selection of the training set, as the decision boundary of a classifier is learnt from training objects. Figure 7.2 shows the decision boundary of the mixture of Gaussians classifier using eccentricity and compactness.

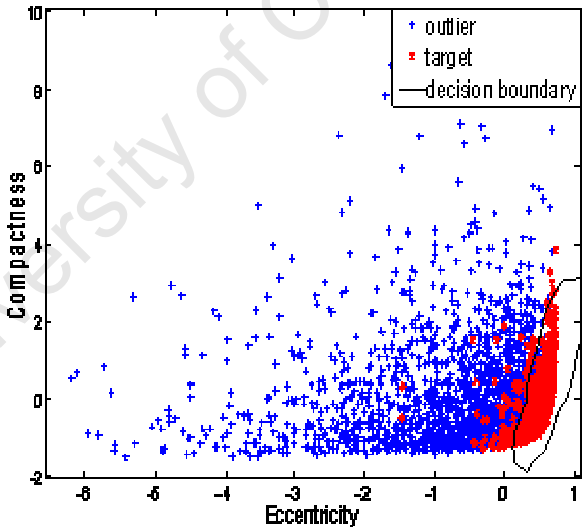

Figure 7.2: The decision boundary of the mixture of Gaussians classifier.

Although the two-class classification method presented performs better than the one class classification implementation, there is scope for further work on one-class classification, for instance, other proximity-based classifiers could be compared to the *k*NN classifier.

8 DISCUSSION, CONCLUSIONS AND RECOMMENDATIONS

The established identification route at 100x magnification took 53.39 seconds to label objects in an image. The time to label objects in one image, almost one minute, can be improved. The algorithms can be re-written with attention paid to implementing routines in a time efficient manner. The use of object-oriented programming could be explored to limit looping in algorithms. Further time gains would be made if parallel processing is employed to run the analysis algorithms. The pattern recognition algorithms are written in MATLAB; a future development might be to convert all software to the C programming language.

Two different microscopes have been used in this investigation. Because digital cameras capture and register light differently, bacilli and background may have different colour properties in images obtained with different cameras. The normalisation of colour images from different microscopes will allow the generalisation of algorithms for different platforms and save time in the construction of training and test datasets.

The performance of the bacillus identification procedure was good at 100x magnification. The procedure should be investigated at microscope objective lens magnifications between 20x and 100x to establish the trade-off between object classification accuracy and objective lens magnification, because, as magnification is lowered, the size of the microscope view field increases and the slide is scanned more quickly. Such time saving allows more slides to be processed and is important for laboratories in countries with a high TB burden.

The effect of the number of test images on the segmentation evaluation should be investigated. The number and the sampling of the test images may significantly affect the segmentation performance; however, since the classification is pixel-based the number of test images used might be sufficient to make the solution independent of the selection of test images, because of the large number of objects (pixels) used.

Further work can be done to probe the performance of classifiers on different settings on the ROC curve. The classifier results presented were calculated on the point on the ROC curve established by cross-validation. Classifiers could be tested on a point on the ROC curve that has higher sensitivity and lower specificity; this is in line with one of the principles of screening – not to lose positive cases. Subsequent steps in the screening process may be used to detect the false positives.

Future work should include a search for the most descriptive feature, because simple classifiers will have good performance with it. Simple classifiers are fast as they do not need to draw complex decision boundaries between object classes. Fast classifiers are very appealing because the aim of automation is to speed up the TB screening process.

The classification results presented are based on the distribution of objects in the training dataset. The results may be generalised to a larger population of objects if it may be assumed that the training dataset was properly sampled. The assumed prior distribution of the two classes may be verified by an investigation to determine if, for a large set of sputum slides, the segmentation method would yield the same ratio of positive to negative objects as obtained in this study. Such a large set of sputum smear slides should preferably be from different laboratories so that variations in sputum smear preparation may be captured.

The investigation was carried out using sputum smear slides diagnosed as positive. The identification path established should be tested on slides that were declared negative by a technician.

It should be determined if the automated identification path is able to produce a similar grading to that produced by a technician in classifying a slide with a + or ++ ranking (WHO, 2003).

For the automated microscope being developed, the number of non-bacillus objects will be reduced by the optimisation of the microscope optics and lighting for the detection only of acid-fast bacilli. The specification of the identification process as a novelty detection is appealing because, in theory, the only red rod-like objects in captured images are bacillus objects. One-class classification should be investigated further on images obtained with optimised optics and lighting.

The methods described may be used to reduce technician involvement in screening for tuberculosis, and will be particularly useful in laboratories in countries with a high burden of tuberculosis. The results may be useful in improving the sensitivity of conventional microscope TB screening: screening is normally done on a population containing more negative cases that positive cases; a technician would therefore have to examine a large number of additional fields to improve the detection of TB cases by a small fraction. However, a computer can process a large dataset to improve the detection of TB cases because it is consistent and does not suffer the fatigue humans suffer. Computer pattern recognition algorithms may be designed for very high sensitivity at the cost of specificity; a technician may be employed to find false positive cases among computer positive results. The technician would only have to examine those images deemed by the algorithm to be positive, rather than images from an entire slide.

APPENDICES

Appendix A

Geodesic Influence Zones

Suppose A is a connected set, the geodesic distance $d_A(x, y)$ between two pixels x and y in A is the length of the smallest path joining x and y that is totally contained in A.

If A contains a set B made of several connected components $B_1, B_2, ...B_k$; the geodesic influence zone $iz_A(B_i)$ of a connected set B_i of B in A is the locus of the points in A whose geodesic distance to B_i is smaller than their geodesic distance to any other component of B (Vincent and Sollie, 1991).

$$iz_A(B_i) = \left\{ A \ni p, \frac{[1,k]}{\{i\}} \ni \forall j, d_A(p, B_i) \leq d_A(p, B_j) \right\}$$

Appendix B

The generalised colour moments

A colour pattern is a function that assigns to each image pixel a 3 element vector representing its colour, for example RGB values (Mindru et al., 2004). The generalised colour moment of a colour pattern is defined by

$$M_{pq}^{abc} = \int\int_{\Omega} x^p y^q [R(x,y)]^a [G(x,y)]^b [B(x,y)]^c \, dxdy$$

M^{abc}_{pq} is said to be the generalised colour moment of order $p + q$ and degree $a + b + c$. Generalised colour moments of degree zero are shape moments of an image; and those of degree one are intensity moments of the RGB bands. Generalised colour moments of order zero are the non-central moments of the colour distribution in the image. Table 2 has further details.

Table 2: From Mindru et al. (2004). The invariants involve 1 or 2 colour bands; S_{cd} stands for 1-band invariants, and D_{cd} for 2-bands invariants of order c, and degree d, respectively. M^{i}_{pq} stands for either M^{i00}_{pq}, M^{0i0}_{pq} or M^{00i}_{pq}, depending on which colour band is used. M^{ij}_{pq} stands for either M^{ij0}_{pq}, M^{i0j}_{pq} or M^{0ij}_{pq}, depending on which 2 of the 3 colour bands are used.

Type 2 (GPSO)

$$S_{12} = \frac{\left\{ \begin{array}{l} M^2_{10}M^1_{01}M^0_{00} - M^2_{10}M^1_{00}M^0_{01} - M^2_{01}M^1_{10}M^0_{00} \\ +M^2_{01}M^1_{00}M^0_{10} + M^2_{00}M^1_{10}M^0_{01} - M^2_{00}M^1_{01}M^0_{10} \end{array} \right\}^2}{(M^0_{00})^2\left[M^2_{00}M^0_{00} - (M^1_{00})^2\right]^3}$$

$$D_{02} = \frac{\left[M^{11}_{00}M^{00}_{00} - M^{10}_{00}M^{01}_{00}\right]^2}{\left[M^{20}_{00}M^{00}_{00} - (M^{10}_{00})^2\right]\left[M^{02}_{00}M^{00}_{00} - (M^{01}_{00})^2\right]}$$

$$D^1_{12} = \frac{\left\{M^{10}_{10}M^{01}_{01}M^{00}_{00} - M^{10}_{10}M^{01}_{00}M^{00}_{01} - M^{10}_{01}M^{01}_{10}M^{00}_{00} + M^{10}_{01}M^{01}_{00}M^{00}_{10} + M^{10}_{00}M^{01}_{10}M^{00}_{01} - M^{10}_{00}M^{01}_{01}M^{00}_{10}\right\}^2}{(M^{00}_{00})^4\left[M^{20}_{00}M^{00}_{00} - (M^{10}_{00})^2\right]\left[M^{02}_{00}M^{00}_{00} - (M^{01}_{00})^2\right]}$$

$$D^2_{12} = \frac{\left\{ \begin{array}{l} M^{20}_{10}M^{01}_{01}(M^{00}_{00})^2 - M^{20}_{10}M^{01}_{00}M^{00}_{01}M^{00}_{00} - M^{20}_{01}M^{01}_{10}(M^{00}_{00})^2 + M^{20}_{01}M^{01}_{00}M^{00}_{10}M^{00}_{00} \\ +M^{20}_{00}M^{01}_{10}M^{00}_{01}M^{00}_{00} - M^{20}_{00}M^{01}_{01}M^{00}_{10}M^{00}_{00} + 2M^{10}_{01}M^{01}_{10}M^{10}_{00}M^{00}_{00} - 2M^{01}_{10}(M^{10}_{00})^2M^{00}_{01} \\ +2M^{01}_{01}(M^{10}_{00})^2M^{00}_{10} - 2M^{10}_{10}M^{01}_{01}M^{10}_{00}M^{00}_{00} + 2M^{10}_{10}M^{10}_{00}M^{01}_{01}M^{00}_{00} - 2M^{10}_{01}M^{10}_{00}M^{01}_{00}M^{00}_{10} \end{array} \right\}^2}{(M^{00}_{00})^4\left[M^{20}_{00}M^{00}_{00} - (M^{10}_{00})^2\right]^2\left[M^{02}_{00}M^{00}_{00} - (M^{01}_{00})^2\right]}$$

$$D^3_{12} = \frac{\left\{ \begin{array}{l} M^{02}_{10}M^{10}_{01}(M^{00}_{00})^2 - M^{02}_{10}M^{10}_{00}M^{00}_{01}M^{00}_{00} - M^{02}_{01}M^{10}_{10}(M^{00}_{00})^2 + M^{02}_{01}M^{10}_{00}M^{00}_{10}M^{00}_{00} \\ +M^{02}_{00}M^{10}_{10}M^{01}_{01}M^{00}_{00} - M^{02}_{00}M^{10}_{01}M^{01}_{10}M^{00}_{00} + 2M^{10}_{10}M^{01}_{01}M^{01}_{00}M^{00}_{00} - 2M^{10}_{10}(M^{01}_{00})^2M^{00}_{01} \\ +2M^{10}_{01}(M^{01}_{00})^2M^{00}_{10} - 2M^{01}_{10}M^{10}_{01}M^{01}_{00}M^{00}_{00} + 2M^{01}_{10}M^{10}_{00}M^{01}_{01}M^{00}_{00} - 2M^{01}_{01}M^{10}_{00}M^{01}_{00}M^{00}_{10} \end{array} \right\}^2}{(M^{00}_{00})^4\left[M^{20}_{00}M^{00}_{00} - (M^{10}_{00})^2\right]\left[M^{02}_{00}M^{00}_{00} - (M^{01}_{00})^2\right]^2}$$

$$D^4_{12} = \frac{\left\{ \begin{array}{l} M^{11}_{10}M^{10}_{01}(M^{00}_{00})^2 - M^{11}_{10}M^{10}_{00}M^{00}_{01}M^{00}_{00} - M^{11}_{01}M^{10}_{10}(M^{00}_{00})^2 + M^{11}_{01}M^{10}_{00}M^{00}_{10}M^{00}_{00} \\ +M^{11}_{00}M^{10}_{10}M^{01}_{01}M^{00}_{00} - M^{11}_{00}M^{10}_{01}M^{01}_{10}M^{00}_{00} + M^{10}_{10}M^{01}_{01}M^{10}_{00}M^{00}_{00} - M^{10}_{10}M^{10}_{00}M^{01}_{00}M^{00}_{01} \\ +M^{10}_{01}M^{10}_{00}M^{01}_{00}M^{00}_{10} - M^{01}_{10}M^{10}_{01}M^{10}_{00}M^{00}_{00} + M^{01}_{10}(M^{10}_{00})^2M^{00}_{01} - M^{01}_{01}(M^{10}_{00})^2M^{00}_{10} \end{array} \right\}^2}{(M^{00}_{00})^4\left[M^{20}_{00}M^{00}_{00} - (M^{10}_{00})^2\right]^2\left[M^{02}_{00}M^{00}_{00} - (M^{01}_{00})^2\right]}$$

$$D^5_{12} = \frac{\left\{ \begin{array}{l} M^{11}_{10}M^{01}_{01}(M^{00}_{00})^2 - M^{11}_{10}M^{01}_{00}M^{00}_{01}M^{00}_{00} - M^{11}_{01}M^{01}_{10}(M^{00}_{00})^2 + M^{11}_{01}M^{01}_{00}M^{00}_{10}M^{00}_{00} \\ +M^{11}_{00}M^{01}_{10}M^{00}_{01}M^{00}_{00} - M^{11}_{00}M^{01}_{01}M^{00}_{10}M^{00}_{00} - M^{10}_{10}M^{01}_{01}M^{01}_{00}M^{00}_{00} + M^{10}_{10}(M^{01}_{00})^2M^{00}_{01} \\ -M^{10}_{01}(M^{01}_{00})^2M^{00}_{10} + M^{01}_{10}M^{10}_{01}M^{01}_{00}M^{00}_{00} - M^{01}_{10}M^{10}_{00}M^{01}_{01}M^{00}_{00} + M^{01}_{01}M^{10}_{00}M^{01}_{00}M^{00}_{10} \end{array} \right\}^2}{(M^{00}_{00})^4\left[M^{20}_{00}M^{00}_{00} - (M^{10}_{00})^2\right]\left[M^{02}_{00}M^{00}_{00} - (M^{01}_{00})^2\right]^2}$$

Appendix C

Contents of the MATLAB M-files on the accompanying CD

watershed_segmentation – **Marker-Controlled Watershed Segmentation**
bayesPIXELclassifier – **Bayes' Pixel Classifier**
linPIXELclassifier – **Linear Pixel Classifier**
loglcPIXELclassifier – **Logistic Linear Pixel Classifier**
quadrcPIXELclassifier – **Quadratic Pixel Classifier**
combinations_final – **Combination of Multiple Pixel Classifiers Segmentation**
pixelcannysegmentation – **Contextual Pixel Classification using Canny Edge Detector Segmentation**
evalPIXELclassifiers - **Segmentation Evaluation**
run_extraction – **Performs Feature Extraction Using the Next 4 functions**
ecc_are_com_features – **Eccentricity and Compactness**
fourier_features – **Fourier Coefficients**
moments2_features – **Moment Features**
colour_features – **Colour Features**
normalize – **Normalise Feature Dataset**
nlfishermappling – **Linear Fisher Mapping**
labelling_finally_objects – **Manually Labelling Segmented Objects**
Expert_Labeller – **MATLAB compiled GUI to be run by a technician manually outlining bacillus contours or creating dataset of objects for testing classifiers**
PBIL – **Population Based Incremental Learning**
best_first_search – **Correlation Based Feature Subset Selection**
SFFS – **Sequential Floating Forward Selection**
BandB – **Branch and Bound Feature Subset Selection**
Bayesfinal – **Bayes' Classifier**
Knnfinal – **Nearest Neighbour Classifier**
knn_dissimilarity – **Dissimilarity Spaces**
Loglcfinal – **Logistic Linear Classifier**

Quadrcfinal – **Quadratic Discriminant Classifier**

train_DENSITIES – **Quadratic Discriminant Classifier Training**

PNNfinal – **Probabilistic neural networks**

train_PNN – **Training Probabilistic neural networks**

SVMfinal – **Support Vector Machines**

train_SVM – **Support Vector Machines Training**

subspaceclassi100x – **Subspace Learning**

roc_myNEW and auc_my – **Receiver Operating Characteristics Analysis**

one_class_gauss – **One-class Gaussian pixel classifier**

one_class_mog – **One-class mixture of Gaussians pixel classifier**

one_class_PCA – **One-class PCA pixel classifier**

one2_class_gauss – **One-class Gaussian object classifier**

one2_class_mog – **One-class mixture of Gaussians object classifier**

one2_class_pca – **One-class PCA object classifier**

one2_class_knn – **One-class kNN object classifier**

one2_class_auc – **One-class Receiver Operating Characteristics Analysis**

Different MATLAB datasets are also included:

SEGMENTATION_IMAGES – a dataset of captured microscope images,
PIXELCOMBINATION_SEGMENTATION_RESULTS – a dataset of pixel classifier mappings used to investigate a suitable colour space,
EXTRACTED_FEATURES_RESULTS – a dataset of training and testing objects,
CLASSIFIERS_FINAL – a dataset of object classifier mappings.

REFERENCES

Baluja S, 1994, 'Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization And Competitive Learning', Technical Report CMU-SC-94-163 Department of Computer Science, Carnegie-Mellon University

Bishop C, 1995, 'Neural Networks for Pattern Recognition', Oxford University press, Oxford

Bishop C, 1994, ' Novelty detection and neural network validation', IEE *Proceedings on Vision, Image and Signal Processing. Special Issue on Applications of Neural Networks,* 141(4), 217 - 222

Burdash N, Manos J, Ross D, Bannister E, 1976, 'Evaluation of the Acid-Fast Smear', *Journal of Clinical Microbiology*, 4(2), 190 – 191

Castleman K, 1998, 'Concepts in Imaging and Microscopy: Colour Image Processing for Microscopy', *Biological Bulletin*, 194, 100 – 107

Cawley G, 2000, 'MATLAB Support Vector Machine Toolbox', University of East Anglia School of Information Systems

Digabel H, Lantuejoul C, 1977, 'Iterative Algorithms', *Proc. of the 2nd European Symposium on Quantitative Analysis of Microstructures in Material Science, Biology and Medicine*, Caen

Duin R, Juszczak P, Paclik P, Pekalska E, de Ridder E, Tax D,2004, 'PRTools4, a Matlab Toolbox for Pattern Recognition', Delft University of Technology

Duda R, Hart P, Stock D, 2001,'Pattern Classification', John Wiley, New York

Egmont-Petersen M, de Ridder D, Handels H, 2002, 'Image Processing with Neural Networks – A Review', *Pattern Recognition,* 35, 2279 – 2301

Forero M, Cristobal G, Desco M, 2006, 'Automatic Identification of Mycobacterium Tuberculosis by Gaussian Mixture Models', *Journal of Microscopy,* 223, 120–132

Franco A, Lumini A, Maio D, Nanni L, 2006, ' An Enhanced Subspace Method For Face Recognition', *Pattern Recognition Letters,* 27, 76 - 84

Fukunaga K, 1990, 'Introduction to Statistical Pattern Recognition', Academic Press, San Diego

Gonzalez R, Woods R, Eddins S. 2004, 'Digital Image Processing Using MATLAB', Pearson Prentice Hall, New Jersey

Greene J, 2001, 'Feature Subset Selection Using Thornton's Separability Index and its Applicability to a Number of Sparse Proximity-Based Classifiers', *Proc. of the 12th Annual Symposium of the South African Pattern Recognition Association,* Franschoek

Hall M, 2000, 'Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning', *Proc. of the 17th International Conference on Machine Learning,* Stanford

Joachims T, 1999, 'Estimating the Generalization Performance of a SVM Efficiently', LS-8 Report 25, Universitat Dortmund, Fachbereich Informatik

Kaufman L, Rousseeuw P, 1990, 'Finding Groups in Data: An Introduction to Cluster Analysis', Wiley, New York

Lenseigne B, Brodin P, Christophe T, Genovesio A, 2007, 'Support Vector Machines for Automatic detection of Tuberculosis Bacteria in Confocal Microscopy Images', *Proc. of the 4th IEEE Symposium on Biomedical Imaging: From Nano to Macro*, Arlington

Lesarson K, Yen N, Thornton C, Mai V, Jones W, An D, Phuoc H, Trinh N, Nhung D, Lien T, Lan N, Wells C, Binkin N, Cetron M, Maloney S, 2005, 'Improved Sensitivity of Sputum Smear Microscopy after Processing Specimens with C_{18}-Carboxypropylbetaine to Detect Acid-Fast Bacilli: a Study of United States-Bound Immigrants from Vietnam', *Journal of Clinical Microbiology*, 43(7), 3460 – 3462

Long X, Cleveland W, Yao Y, 2004, 'Automatic Detection of Unstained Viable Cells in Bright Field Images Using a Support Vector Machine with an Improved Training Procedure', *Computers in Biology And Medicine*, 36(4), 339 – 362

MATLAB R2007a, Documentation, MathWorks, Natick, MA

Metz 1978, 'Basic Principles of ROC Analysis', *Seminars in Nuclear Medicine*, 8(4), 283 - 298

Meuric C, Charrier C, Lezoray O, Elmoataz A, 2005, 'Combination of Multiple Pixel Classifiers for Microscopic Image Segmentation', *International Journal of Robotics and Automation*, 206(2), 2780

Meuric C, Lebrun G, Lezoray O, Elmoataz A, 2003, 'A Comparison of Supervised Pixel-Based Colour Image Segmentation Methods. Application in Cancerology', *WSEAS Transactions on Computers, special issue on ICOSSIP'03*, 2, 739-744

Mindru F, Tuytelaars T, Van Gool L, Moons T, 2004, 'Moment Invariants for Recognition under Changing Viewpoint and Illumination', *Computer Vision and Image Understanding*, 94, 3 - 27

Nattkemper T W, Twellmann T, Ritter H, Schubert W, 2003, ' Human vs. Machine: Evaluation of Fluorescence Micrographs', *Computers in Biology and Medicine,* 33(1), 31 - 43

Pekalska E, Duin R, Paclik P, 2006, 'Prototype Selection for Dissimilarity-Based Classification', *Pattern Recognition,* 39(2), 180 - 208

Platt J, 1998, 'Sequential Minimal Optimisation: A Fast Algorithm for Training Support Vector Machines', Technical Report MSR – TR – 98 – 14, Microsoft Research

Pudil P, Ferri F J, Novovicova J, Kittler J, 1994, 'Floating Search Methods for Feature Selection with Non-monotonic Criterion Functions', *Pattern Recognition,* 2, 279 - 283

Russell M, 2006, 'Autofocusing and Image Segmentation in Microscopy for Automatic Detection of Tuberculosis in Sputum Smears', MSc (Med) Thesis, Department of Human Biology, University of Cape Town

Sadaphal P, Rao J, Comstock G, Beg M, 2008, 'Image Processing Techniques for Identifying *Mycobacterium Tuberculosis* in Ziehl-Neelsen Stains', *International Journal of Tuberculosis and Lung Disease,* 12(5), 579 – 582

Santiago-Mozos R, Fernandez-Lorenzana R, Perez-Cruz F, Artes-Rodriguez A, 2008, 'On the Uncertainty in Sequential Hypothesis Testing', *Proc. of the 5th IEEE Symposium on Biomedical Imaging: From Nano to Macro*, Paris

Sollie P, 2003, 'Morphological Image Analysis: Principles and Applications', Springer, Berlin

Somol P, Pudil P, 2000, 'Oscillating Search Algorithms for Feature Selection', *Proc. of the 15th IAPR International Conference on Pattern Recognition,* Barcelona

Somol P, Pudil P, Kittler J, 2004, 'Fast Branch & Bound Algorithms for Optimal Feature Selection', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7), 900 – 912

Sonka M, 2008, 'Image Processing, Analysis, and Machine Vision', Thomson, Toronto

Steingart K, Henry M, Ng V, Hopewell P, Ramsay A, Cunningham J, Urbanczik R, Perkins M, Aziz M, Pai M, 2006, 'Fluorescence versus Conventional Sputum Smear Microscopy for Tuberculosis: a Systematic Review', *Lancet Infectious Diseases*, 6, 570 – 581

Tax D, 2008, 'DDtools, the Data Description Toolbox for Matlab', version 1.7.0

Tax D, 2001, 'One-class Classification: Concept-learning in the Absence of Counter-examples', PhD Thesis, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology

Tax D, Muller K, 2003, 'Feature extraction for one-class classification', *Proc. of the International Conference on Artificial Neural Networks and Neural Information Processing*, Istanbul, 342-349

Theodoridis S, Koutroumbas K. 2003, 'Pattern Recognition', Elsevier, San Diego

Toman K, 2004, 'What are the advantages and disadvantages of fluorescence microscopy?' In: Frieden T, ed. Toman's tuberculosis: case detection, treatment, and monitoring—questions and answers (2nd edn). Geneva: World Health Organization, 31–34.

Trattner S, Greenspan H, Tepper G, Abboud S, 2004, 'Automatic Identification of Bacterial Types Using Statistical Imaging Methods', *IEEE Transactions on Medical Imaging*, 23(7), 807 – 821

Twellmann T, Nattkemper T W, Schubert W, Ritter H, 2001, 'Cell Detection in Micrographs of Tissue Sections Using Support Vector Machines', *Proc. of the ICANN Workshop on Kernel and Subspace Methods for Computer Vision*, Vienna

Vapnik V, 1998, 'Statistical Learning Theory', Wiley, New York

Veropoulos K, 2001, 'Machine Learning Approaches to Medical Decision Making', PhD Thesis, Department of Computer Science, University of Bristol

Veropoulos K, Learmonth G, Campbell C, Knight B, Simpson J, 1999, 'Automated Identification of Tubercle Bacilli in Sputum: A Preliminary Study', *Analytical and Quantitative Cytology and Histology*, 21(4), 277-281

Vincent L, Soille P, 1991, 'Watersheds in Digital spaces: An efficient Algorithm Based on Immersion Simulations', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6), 583 - 598

World Health Organisation, 2007, 'Tuberculosis Facts Sheets', http://www.who.int/mediacentre/factsheets/fs104/en/ Last date accessed: 21 August 2008

World Health Organisation, 2003, 'Supporting Laboratory Services', http://whqlibdoc.who.int/hq/2003/WHO_CDS_TB_2002.310_mod9_eng.pdf Last date accessed: 8 July 2008

Ypma A, Tax D, Duin R, 1999, 'Robust Machine Fault Detection with Independent Components Analysis and Support Vector Data Description', *Proc. of the IEEE International Workshop on Neural Networks for Signal Processing*, Madison

Yusuf A, Xiaolei H, 2008, 'Combining Multiple 2ν-SVM Classifiers for Tissue Segmentation', *Proc. of the 5th IEEE Symposium on Biomedical Imaging: From Nano to Macro*, Paris