Automated Tuberculosis Diagnosis Using Fluorescence Images from a Mobile Microscope

by

Jeannette Nancy Chang

A thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science
in

Engineering - Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Jitendra Malik, Co-Chair
Professor Daniel Fletcher, Co-Chair
Professor Michael Lustig

Spring 2012

Automated Tuberculosis Diagnosis Using Fluorescence Images from a Mobile Microscope

Abstract

Automated Tuberculosis Diagnosis Using Fluorescence Images from a Mobile Microscope

by

Jeannette Nancy Chang

Master of Science in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Jitendra Malik and Professor Daniel Fletcher, Co-Chairs

In low-income countries, the most common method of tuberculosis (TB) diagnosis is visual identification of rod-shaped TB bacilli in sputum smears by microscope. We present an algorithm for automated TB detection in smear images taken by digital microscopes such as CellScope [1], a novel low-cost, portable device capable of brightfield and fluorescence microscopy. Automated processing on such platforms could save lives by bringing healthcare to rural areas with limited access to laboratory-based diagnostics. Though the focus of this study is the application of our automated algorithm to CellScope images, our method may be readily generalized for use with images from other digital fluorescence microscopes.

Our algorithm applies morphological operations and template matching with a Gaussian kernel to identify TB-object candidates. We then use moment, geometric, photometric, and oriented gradient features to characterize these objects and perform discriminative, support vector machine classification. We test our algorithm on a large set of CellScope fluorescence images from sputum smears collected at clinics in Uganda (594 images corresponding to 290 patients). Our object-level classification is highly accurate, with Average Precision of 89.2% ± 2.1%. For slide-level classification, our algorithm performs at the level of human readers, demonstrating the potential for making a significant impact on global healthcare.

Dedicated to my dear family
...for inspiring me to remember what is most meaningful in life.

Contents

Acknowledgements

I am greatly indebted to many individuals who generously contributed to this study. I am grateful for the patient guidance of my advisers, Jitendra Malik and Daniel Fletcher, from UC Berkeley's Electrical Engineering/Computer Sciences and Bioengineering Departments, respectively. Thank you for embracing this interdisciplinary effort in its potential to impact global healthcare.

My utmost gratitude to Pablo Arbelaez, researcher in the Malik Group, for his many helpful insights and discussions. I appreciate your thoughtful feedback and suggestions throughout various research cycles. You have been an immense source of support and encouragement, and I am thankful for the good-natured positivity you bring to the office each day.

I am also deeply thankful for Neil Switz, PhD candidate in the Fletcher Lab, who always made the time to discuss the technicalities of the CellScope device and related topics. Thank you for playing an integral role in bridging the computer vision, bioengineering, and medical communities involved in this project.

Thank you, Clay Reber, for helpful discussions about data collection and logistics. I appreciate your cooperativeness in meticulously generating the object-level human annotations. Thank you also to Asa Tapley for your insightful comments, especially in terms of the medical and tuberculosis aspects of the project. Similarly, I would like to thank Adithya Cattamanchi and Lucian Davis from the San Francisco General Hospital/UCSF for bringing the invaluable and practical perspective of the medical community.

My sincere thanks go to Ilge Akkaya for her tremendous help in investigating and discussing the papers presented in Chapter 2. Chunhui Gu, Saurabh Gupta, Jon Barron, and Bharath Hariharan, thank you for miscellaneous, enlightening discussions in the Malik Bay. Thank you also to Mike D'Ambriosio and Lina Nilsson from the Fletcher Lab for your contributions in moving towards the deployment of our algorithm in the field.

I would also like to thank our collaborators in Uganda, both the patients who provided the data samples and the medical personnel who collected and prepared them. I hope this work will be of service to you in the near future.

Last but not least, my heartfelt thanks to my father, mother, and sister for their unwavering support. Thank you for always standing by me.

Chapter 1

Introduction

1.1 Brief Background on Tuberculosis

The bacteria that causes TB disease, known as *Mycobacterium tuberculosis*, was first identified in 1882 by Robert Koch, a German physician and microbiologist [2]. Infection by TB bacteria usually happens via exposure to airborne droplets carrying the bacteria [2], [3]. Though the likelihood of developing TB disease upon being infected with TB bacteria is typically low, certain populations are at substantially higher risk (e.g., youth, the elderly, and those who are malnourished or have compromised immune systems). TB most often infects the lungs but may also spread to almost any other part of the body, such as bone marrow and lymphatic vessels. Symptoms of TB are diverse, including "chest pain, shortness of breath, fever, night sweats, fatigue, appetite loss, and unintentional weight loss" [2]. Nevertheless, given TB bacteria's propensity for pulmonary infection, the most common symptom associated with active TB infection is prolonged cough.

Though tuberculosis (TB) garners relatively little attention in high-income countries today, it remains the second leading cause of death from infectious disease worldwide (second only to HIV/AIDS) [2], [3]. In 2010 alone, 1.2-1.5 million deaths were attributed at least in part to TB. Low-income parts of the world suffer a disproportionately high fraction of TB-related fatalities, with approximately 85% of TB cases occurring in Asia and Africa.

1.2 Methods of Tuberculosis Diagnosis

The majority of TB cases may be treated successfully with the appropriate course of antibiotics, but diagnosis remains a large obstacle to TB eradication. Presently, the most common method of diagnosing patients with TB is visually screening stained smears prepared from sputum. Technicians use microscopes to view the smears, looking for rod-shaped objects (sometimes characterized by distinct beading or banding) that may be *Mycobacterium tuberculosis.* Apart from the costs of trained technicians, laboratory infrastructure, microscopes and other equipment, this process

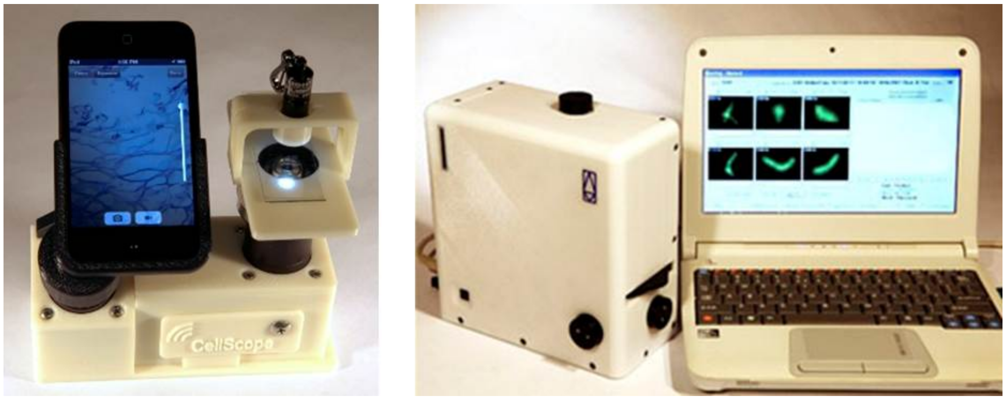


Figure 1.1. Two versions of CellScope, a novel mobile microscope. The device may be used in multiple ways, such as for point-of-care diagnostics or for transmitting images from rural areas to medical experts. Images used in this study were taken by the prototype on the right.

suffers from low recall rates, inefficiency, and inconsistency due to fatigue and inter-evaluator variability [4]. Hence, with the advent of low-cost digital microscopy, automated TB diagnosis presents a ready opportunity for the application of modern computer vision techniques to a real-world, high-impact problem.

Additional TB diagnostic procedures include culture and polymerase chain reaction (PCR)-based methods. Culture results are ideally used to verify smear screenings and are currently the gold-standard for diagnosis. However, culture assays are more expensive and technically challenging to perform than smear microscopy and require prolonged incubation: about 2-4 weeks to allow accurate evaluation of bacteria [5]. Such a delay is far from ideal for a patient who should already be engaged in an antibiotic treatment to prevent further spread of the disease. PCR-based methods such as Cepheid's GeneXpert assess the presence of TB bacterial DNA and are rapid, more sensitive than smear microscopy, and capable of testing resistance to a common anti-TB antibiotic [6]. Notwithstanding, they continue to lag in sensitivity compared to culture and rely on costly equipment that is poorly suited for low-resource, peripheral healthcare settings [7]. Sputum smear microscopy continues to be by far the most widely used method of TB diagnosis, suggesting that enhancements to microscopy-based screening methods could provide significant benefits to large numbers of TB-burdened communities across the globe.

1.3 CellScope and Related Devices

The CellScope [1] is a novel digital microscope developed by the Fletcher Lab at UC Berkeley's Bioengineering Department (Figure 1.1). Given its compact form factor (20x20x10 cm), light weight (3 kg), and battery-powered design, the CellScope is a very portable device [2]. It uses a 20x 0.4 numerical aperture (NA) microscope objective, which affords Rayleigh resolution of 0.76 μm (640x490 μm sample-referenced field of view). CellScope images are sampled above Nyquist frequency, enabling digital magnification via interpolation to effective magnifications of 2000-3000x. Fluorescence excitation in the device is supplied by a 1 Watt, 460 nm LED. Exposure times for the

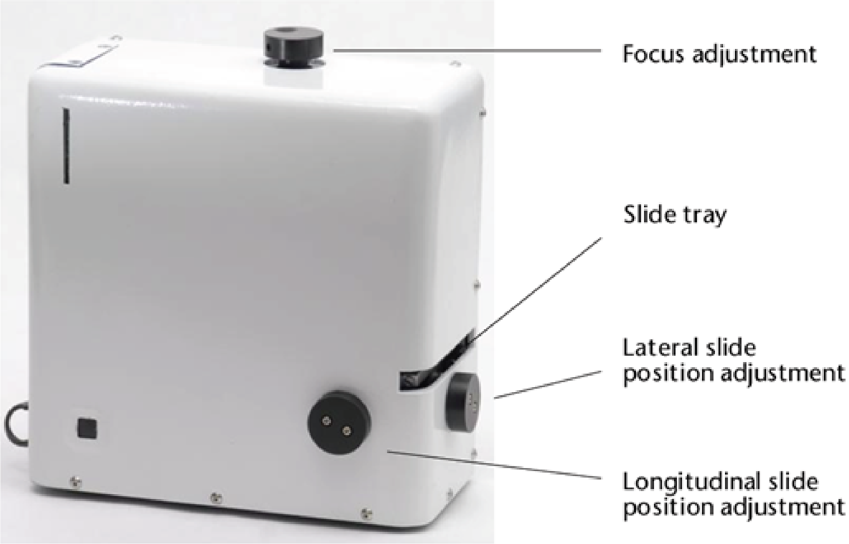


Figure 1.2. Main features of the CellScope prototype used in this study. Slides with the stained sputum smears are inserted at the slide tray. Position and focus are adjusted manually using the knobs shown. Image from [2].

CellScope's 8-bit monochrome CMOS image sensor generally fall within the 100-500 ms range depending on the staining process (though dark noise is negligible in all cases). Given that CellScope uses components found in commercial cellphones, the target price range for CellScopes is significantly lower than costs of alternatives such as conventional fluorescence microscopy (FM) and PCR-based methods.

Various CellScope prototypes have been developed for applications ranging from medical (e.g., TB diagnosis and dermatological imaging) to educational uses. In this study, we use images taken by the prototype seen in Figure 1.2. CellScope is capable of both brightfield microscopy and FM, but we focus on FM in our discussion because studies suggest it is more sensitive [8], [9] and faster [6].

A number of other research groups have advanced the development of low-cost, portable microscopy. In 2006, Yang *et al.* at the California Institute of Technology demonstrated the use of microfluidics-based, lensfree microscopy (termed "optofluidic microscopy") for imaging larval *Caenorhabditis elegans* (approximately 10 μm in width) [10]. Ozcan *et al.* at the University of California, Los Angeles, have proposed lensfree, holographic cellphone-based microscopy [11]. In these systems, incoherent LED light illuminates the sample, and light scattering off of the sample interferes with background light to create a hologram on the CMOS sensor. The microscopic image of the sample may then be reconstructed from the holographic signatures using digital processing. Ozcan's lensfree systems offer a larger field of view (FOV) and could thus provide the benefit of faster imaging. However, to the best of our knowledge, the current resolution of these lensfree microscopes is not suitable for performing TB diagnosis. With some of these alternative methods, it's also possible that higher per-test costs (compared to standard slide-based diagnostic procedures) may pose an additional obstacle to adoption in low-resource areas. More recently, researchers at University of California, Davis, have been exploring single-lens cellphone-based microscopes. These devices have the advantage of being lower cost and could be very useful in educational settings, but their current Rayleigh resolution (1.5μm) is insufficient for TB diagnosis [12], which requires submicron resolution.

1.4 Automated Tuberculosis Diagnosis Using CellScope

In this paper, we propose an algorithm for automated TB detection that may be used with digital imaging devices such as CellScope. CellScope provides an affordable and portable alternative to standard laboratory-based microscopes, making it a vehicle for bringing healthcare to underdeveloped areas. To facilitate implementation using the modest computational power of these mobile phone-based platforms, we seek to develop an effective and robust algorithm. We present results from a large dataset of sputum smears collected under real-field conditions in Uganda. Our algorithm is highly accurate, performing at the level of human readers when classifying slides, which opens exciting opportunities for deployment in large-scale clinical settings. We achieve this result via modern computer vision techniques in object recognition, including segmentation, suitable feature-based representation, and support vector machine classification.

That our automated method performs on par with human readers could greatly increase the impact of CellScope in rural settings, where skilled microscopists are scarce. For instance, automated local processing on CellScope could serve as a first-stage screening and improve point-of-care efficiency. In addition to the possibility of automated diagnostic capabilities, CellScope's connectivity as a cellphone-based device holds potential for further extending the reach of healthcare to low-resource regions. Surprisingly, many of these areas with limited access to healthcare are already equipped with substantial mobile phone infrastructure. One may thus imagine transmitting images of microscopic specimen from remote areas to medical experts in urban centers or establishing an easy-access online image repository.

The remainder of our paper is organized as follows: Chapter 2 provides a summary of related work by other groups. Chapters 3 and 4 outline our algorithm and dataset, respectively. Chapter 5 contains our experimental results and compares our algorithm's performance to that of human readers as well as other automated methods. Chapter 6 is a user's manual that describes how to run our algorithm evaluation and training code. As mentioned in the user's manual, the dataset and code from this study will be publicly available.

Chapter 2

Previous Work

2.1 Automated Tuberculosis Diagnosis For Smear Microscopy

The two main methods of screening sputum samples are fluorescence microscopy (FM) and brightfield microscopy, in which the sputum smears are stained with auramine-O and Ziehl-Neelsen respectively (see Figure 2.1). With FM, smears may be screened at lower magnifications and can thus be examined in less than half the time it takes with brightfield microscopy [6]. Studies also indicate that FM yields about 10% higher recall than brightfield microscopy [8], [9]. Hence, though CellScope is capable of both types of microscopy, we choose to use FM images in this study.

Several groups have explored automated TB detection with both FM and brightfield microscopy. Veropoulos *et al.* (1999, 2001) [13], [14] and Forero *et al.* (2004, 2006) [4], [15] considered TB detection with images from FM. Other groups have devised algorithms for brightfield microscopy, but many of these algorithms rely on the distinct color characteristics of TB-bacilli stained by ZN [16]–[18]. As seen in Figure 2.1, color imaging plays a large role in evaluating ZN-stained smears whereas grayscale imaging suffices for FM smears. Hence, the two types of stained smears often require different automated classification techniques. In this chapter, we present an overview of previous automated TB diagnosis studies for both FM and brightfield microscopy.

2.1.1 Fluorescence Microscopy (FM)

Veropoulos' Algorithm

In the first stage of their algorithm, Veroupoulos *et al.* applied Canny edge detection, filtered objects based on size, and used boundary tracing [13], [14]. Fourier descriptors, intensity features, and compactness were chosen using Branch and Bound or Sequential Forward Selection. A number of probabilistic methods were then employed to classify objects, with a multilayer neural network achieving the best performance.

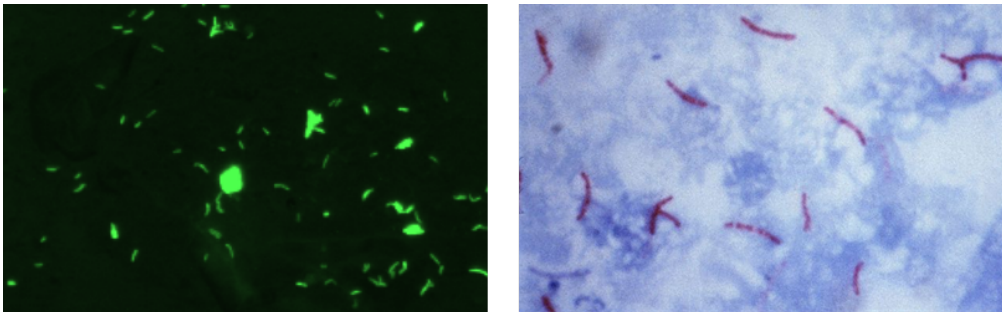


Figure 2.1. Sample CellScope fluorescence image (*left*) and sample brightfield image [19] (*right*). Best viewed in color.

Segmentation

After initial image capture and normalization, a Canny edge detector was applied to determine regions in the image. Based on prior knowledge of TB bacteria sizes, regions that were too small or too large were removed. Then, regions with incomplete contours (generally resulting from occluded or faint bacilli) were further eliminated to simplify the learning process. It was reasoned that, in these cases, there are usually enough bacilli in the rest of the sample for reliable positive diagnosis. Finally, in preparation for finding shape descriptors, Veropoulos implemented boundary tracing and obtained the inner boundaries of each region.

Feature Selection

The authors employed the Branch and Bound (B&B) and Sequential Forward Selection (SFS) methods for feature selection (see Appendix B). A separability criterion proposed by Fukunaga was used to determine the discriminative power of the feature subsets [20]. This metric considers inter- and intra-cluster differences, rewarding large inter-cluster and small intra-cluster distances.

The features considered were the first 14 Fourier descriptors; compactness; average intensity inside/around region; standard deviation of intensity/around region; and the intensity of the centroid (see Appendix A). Results from experimenting with various feature subsets suggested that the most discriminative features included the shape-based features (namely, the 3rd and 4th Fourier descriptors as well as compactness) and low standard deviation inside the region.

Classification

Veropoulos *et al.* implemented various neural network and support vector machine classification methods on a dataset with 65 slides (50 centrifuged and 15 direct auramine-stained smears). The two sets of smears (centrifuged and direct) were considered separately because of their distinct characteristics, with centrifuged smears exhibiting bacilli clustered at the center of the smear and large amounts of background debris. Note that Veropoulos *et al.* consider the algorithm performance at the TB-object level rather than the slide level. In the case of centrifuged smears, the highest overall accuracy was achieved using a neural network classifier with ten hidden units, trained using early stopping and the back-propagation (BP) learning rule (see also Neural Networks in Appendix C). This optimal classifier achieved object-level performance of 93.9% sensitivity and 79.4% specificity. The support vector machines achieved slightly lower overall accuracy, while neural network classifiers using BP without early stopping performed substantially worse. With the direct smears, the neural network classifier trained using early stopping for training achieved superior performance, though with significant differences between sensitivity and specificity. The neural network classifier trained using early stopping and the scaled conjugate gradient (SCG) algorithm yielded the highest overall object-level accuracy (91.4%), with 98.6% sensitivity and 44.6% specificity. The slides used in our study are comparable to the direct smears, as they have not been centrifuged nor digested.

Forero's Algorithm

Forero *et al.* took a generative approach, representing the TB-bacilli class with a Gaussian mixture model and using Bayesian classification techniques [4], [15]. Features used in the model were Hu moments, chosen for their invariance to rotation, scaling, and translation. Other candidate features were considered but eliminated due to redundancy.

Segmentation

Because the green channel contains changes in color intensity due to TB bacilli, Forero extracted the green channel from the original RGB image for the first segmentation phase. The group then applied a Canny edge detector, and closing morphological operators were used to mend gaps in broken edges. Closed regions were then filled, and superfluous edges were removed via an opening operator. Finally, objects that did not exhibit color within a certain range were eliminated by thresholding.

Shape and Size-Based Filtering

Once the bacilli-colored objects were obtained from segmentation, they were filtered based on shape and size using prior knowledge about bacilli. In particular, objects were discarded if their eccentricity (Appendix A) was too low or their area did not fall within an acceptable range (determined empirically).

Feature Extraction

Following the shape and size filtering, Forero's group proposed various shape descriptors for the objects: area, compactness, major and minor axis lengths, eccentricity, perimeter, solidity, Hu moments, and Fourier descriptors (see also Appendix A and Section 3.2). Requiring invariance to translation, rotation, and scaling reduced the list to compactness, eccentricity, Hu moments, and Fourier descriptors. Among these, Hu moments were chosen for their succinctness and robustness to noise. Eccentricity was discarded because of its dependence on moments, while compactness was excluded because it did not improve results in initial experiments. The final remaining features were thus only Hu moments, which were calculated from the normalized central moments of the binarized objects. Only the first three and eleventh Hu moments were used, as the symmetric shape of TB-bacilli make the other Hu moments redundant.

Editing the Training Set

Clinical collaborators mentioned that some ovoid shapes were ambiguous cases; these atypical bacilli would be marked as non-bacilli when seen in isolation but labeled as bacilli when other prototypical bacilli were present close by. Keeping these objects in the positive-bacilli training set could severely degrade the performance of the algorithm. To address this issue, Forero and his colleagues assumed that the distribution of the bacilli shapes could be modeled as a Gaussian distribution and eliminated outlier objects (features lower than 3 standard deviations below the average). In this way, the original training data set of 1412 TB-objects was reduced to 929.

K-Means Clustering and Gaussian Mixture Model (GMM)

To account for inherent variability in TB bacilli shapes (length, thickness, etc.), Forero represented the bacillus class using multiple clusters rather than a single cluster. In order to estimate a suitable number of clusters, the group first implemented the k-means clustering algorithm and verified the effectiveness of the resulting clusters using silhouette plots [21]. The group then characterized the 4-dimensional feature space using a Gaussian Mixture Model (GMM), with a 4-variate Gaussian corresponding to each cluster in the bacillus class. Expectation maximization was used to determine the GMM parameters. See Appendix C for more about k-means clustering and GMMs.

Minimum Error Bayesian Classifier

For the final stage of classification, Forero introduced discriminant functions (one associated with each cluster). The probability of observing a sample feature vector given that the object belongs to a particular cluster was modeled as a 4-variate Gaussian as described in the previous section. The algorithm identified the cluster with the maximal discriminant function and then invoked the Bayesian threshold decision rule corresponding to that cluster. Intuitively, each cluster's decision rule may be expressed in terms of the Mahalanobis distance between the sample feature vector and cluster of interest. See Appendix C for more details.

2.1.2 Brightfield Microscopy

Historically, brightfield or conventional light microscopy has been more popular than FM in low-income countries because of lower equipment costs. It was only recently that advances in FM technology and stainings procedures have enabled more widespread adoption of FM microscopy for TB diagnosis.

In brightfield microscopy, smears are stained with Ziehl-Neelsen (ZN), which causes the TB bacilli to appear magenta against a light blue background. Like mentioned earlier, automated algorithms for brightfield microscopy often differ greatly from those for FM because of the distinct appearance of the two types of microscopic images. Nevertheless, we provide a brief summary here of three proposed algorithms for automated TB identification in brightfield microscopy [16]–[18].

Costa's Algorithm

Preprocessing and Segmentation

Costa's algorithm initially aims to identify the background in the smear images to later segment the image correctly. The background characterization of the image is performed in the RGB color space by obtaining the R minus G (R-G) channel, in which the contrast between ZN-stained bacilli and the background is maximized. This result was achieved after observing the hue histogram and several RGB transforms of the image.

For the segmentation procedure, the 10-bin histogram of the R-G channel was used to binarize the image. A global parameter x was chosen empirically and then used to determine a threshold value L and segment the image in the following way: if the percentage of pixels in a particular bin is greater than x%, the value of these pixels are assigned to the threshold value, L. Otherwise, the pixel intensities are set to zero. The motivation behind this binarization scheme is that bacteria usually appear in a certain shade in ZN-stained images. This thresholding in the R-G channel classifies the background pixels based on their color information and to highlight potential bacillus pixels.

Classification

This binarization procedure of the image introduced two types of artifacts. Namely, large regions and small artifacts (area <200 pixels) in the (R-G) channel were identified as positive (bacilli) but actually belonged to the background. Morphological filters were used to eliminate these false positive regions.

Khutlang's Algorithm

Segmentation

The ZN-stained sputum smear images were segmented in Khutlang's paper by applying pixel classifiers to the RGB image pixels. A number of classifiers were combined to attain better classification results. Bacilli pixels were manually labeled as +1 in 28 training images, while a subset of the background pixels were labeled as -1. Each pixel was treated as an object and fed into the classifiers. The data consisted of 40666 pixel objects total, 20637 of which were bacilli pixels (+1). The outputs of all classifiers were two values per pixel, corresponding to the posterior probability of the pixel being bacillus/non-bacillus. The main classes of classifiers used were Bayes', linear regression, quadratic discriminant and K-nearest neighbor (kNN) (see also Appendix C).

Various classifier combinations were tested on a subset of the images, and their performance was compared to the manually segmented ground truth. The combination schemes were the mean, median, minimum, maximum and the products of classifier outputs (posterior probabilities). Each pixel classification scheme was evaluated based on the common and difference rates (Appendix D), from which percentages of pixels correctly classified and incorrectly classified were derived respectively. The best combination of pixel classifiers was determined using these metrics and applied to the final segmentation.

Feature Extraction

Fourier descriptors, generalized RGB moments, eccentricity and pixel color values were then extracted from these segmented images as shape and color descriptors (Appendix A). In brief, Fourier descriptors are obtained by assigning complex values to each point of a closed boundary in the image and then taking the DFT of the resulting complex sequence. Thus, the reconstruction of the shape of the object may be achieved using the inverse transform. Here, the number of coefficients to be kept was determined by the classification accuracy of the nearest neighbor classifier (Appendix C).

In terms of color description, Khutlang calculated the value of the center pixel of the closed-contour object in each color channel. Other color information such as the mean of all object pixel values, mean of perimeter pixel values and the corresponding standard deviations were also defined as color features.

Feature Selection

A. Feature Subset Selection: One method of feature selection is to choose a subset of the features defined in the previous section. The most effective subset selection methods presented in the paper generally used several techniques to ensure that chosen features were highly correlated with the data set while simultaneously highly uncorrelated with other features in the subset. For details on some common feature subset selection schemes, see Appendix B.

B. Fisher Mapping: Fisher mapping is used to reduce the feature dimensionality after the features of the images are calculated. This mapping is an alternative to feature subset selection algorithms and is shown to yield the best bacillus detection results. It performs an optimization on the feature set, prior to classification.

Classification

In Khutlang's paper, a variety of classifiers were implemented, including Probabilistic Neural Networks (PNNs), Support Vector Machines (SVMs) and kNN classifiers (Appendix C). Classification results were calculated for feature sets from (1) various combinations of feature subset selection schemes and (2) Fisher mapping.

Sadaphal's Algorithm

Color Segmentation

Using the manual segmentation of TB-positive images in the data set, Sadaphal derived a 3-dimensional probability density function histogram denoting the likelihood of a pixel being a bacillus pixel for a specific triplet of red, green, and blue pixel values. The density function revealed that the majority of the bacillus-positive pixels had distinctive RGB values compared to the non-bacillus pixels. A binary mask was obtained by thresholding the pixel intensities in each channel (RGB) and enhanced using morphological dilation with a circular structuring element.

Classification

After this color-based segmentation, pixels were tested against various known bacilli shapes and orientations. Two descriptors were used in this stage, the first being axis ratio (1 for circular objects and greater than 1 for objects closer to line segments). Objects with axis ratio smaller than 1.25 were classified as non-TB. The next descriptor considered was eccentricity, which generally varies between 0.9 and 0.96 for bacilli objects. Observing that eccentricity values are centered around zero for non TB objects, the objects having eccentricity <0.65 were classified as non-TB objects at this stage.

Following these two stages of shape-based elimination of potential TB objects, the mean object size and the standard deviation were computed. Objects whose sizes fell outside the $1.5\sigma \pm \mu$ range were labeled as possible TB-object, while objects within the range were labeled as positive-TB objects.

No quantitative results were provided in this paper, but the algorithm appeared to perform acceptably based on qualitative observations.

2.2 Evaluation of Automated Algorithms

In these previous studies, inconsistency in the algorithm performance metrics (object-level, image-level, slide-level) makes it difficult to compare algorithms [4], [14], [16]–[18]. Note that each patient generally provides a few sputum samples, each of which gives rise to one slide or smear.

The patient is considered TB-positive if any of his or her slides results in a positive diagnosis. (For the Uganda dataset used in this study, we have one sputum slide or smear per patient.) Each sputum slide or smear in turn usually corresponds to multiple images, and each image may contain anywhere from zero to hundreds of TB bacilli objects. Depending on the type of microscopy and method of segmentation, the definition of a "negative" object may also vary greatly. Moreover, independent collection of sputum smear image datasets using different equipment and sample preparation methods results in time-consuming and redundant efforts. We thus intend to make the dataset used in the development of our algorithm publicly available. We futher recommend the designation of both a standard dataset and performance metric to facilitate the development and improvement of automated TB detection algorithms.

2.3 Object Recognition

As a whole, the field of object recognition is a very large and active community, and an exhaustive review of its literature is beyond the scope of the discussion here. We refer the interested reader to textbooks such as Forsyth and Ponce [22] for an updated overview of the subject. The most closely related family of approaches to our work is that of producing object candidates though bottom-up segmentation and classifying them, as opposed to multi-scale sliding-window paradigms. In this tradition, some recent representative papers are [23]–[26]. Here, we draw on classical Hu moments, geometric/photometric properties, and histograms of oriented gradients as features, and then use discriminative classifiers such as the intersection-kernel support vector machine (IKSVM).

Chapter 3

Algorithm

Given the opportunities presented by portable digital microscopes like CellScope, we seek to develop an automated TB detection algorithm for FM that may be implemented on these mobile platforms. While our immediate goal is to apply the algorithm to CellScope images, our method may be generalized for use with laboratory-based digital FM microscopes as well. We draw from modern computer vision tools and consider techniques previously proposed in the related literature. Our algorithm consists of three stages: (1) TB-object candidate identification, (2) feature representation of TB-object candidates, and (3) discriminative classification of the TB-object. A block diagram of the algorithm is shown in Figure 3.1.

3.1 TB-Object Candidate Identification

In this step, our objective is to identify any bright object that is potentially a TB bacillus. We initally focused on minimizing missed detections rather than false positives because candidate objects that are not bacilli may be discarded in the classification stage (whereas missed detections in the first step have no chance of being recovered). Nevertheless, we found that some candidate identification-methods resulted in prohibitive numbers of false positive objects, which in turn increased computational time in the subsequent stages of the algorithm.

Balancing these two extrema, we decided on the following strategy. We first perform two methods in parallel (both operating on the original grayscale image): a white top-hat transform and template matching with a Gaussian kernel. The white top-hat transform reduces noise from fluctuations in the background staining, and the template matching picks out areas that resemble bright spots. The resulting images from these two methods are thresholded to obtain binarized images and then combined via pixel-wise multiplication. From the final binarized image, we extract connected components as TB-object candidates.

We then consider a region of interest or patch from the input image centered around each candidate. The patch-size (24x24 pixels) is chosen based on the known size of the TB bacilli (typically 2-4 μm in length and 0.5 μm in width) and CellScope's sample-referenced pixel spacing

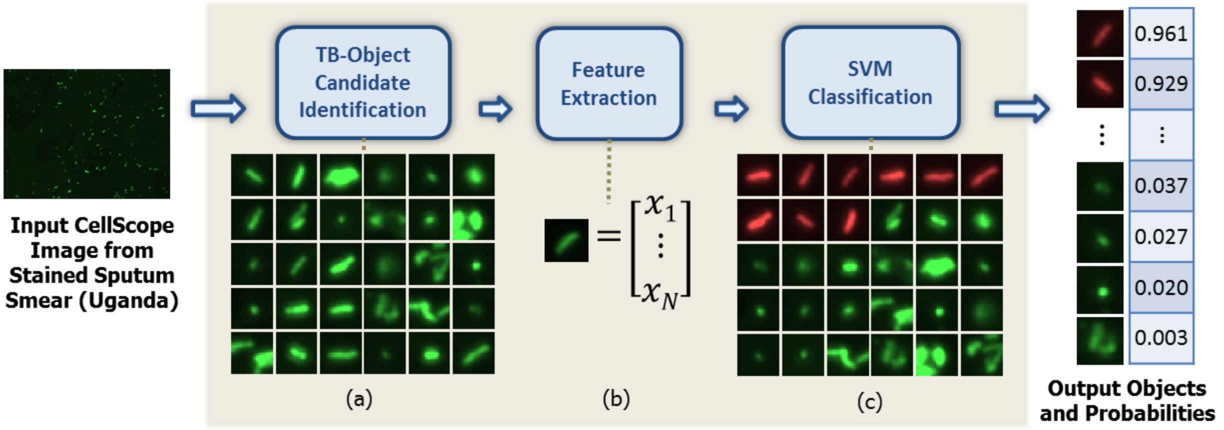


Figure 3.1. Overview of algorithm: TB-object candidate identification, feature extraction, and discriminative classification via SVMs. (a) Array of TB-object candidates. (b) Each candidate is characterized by a N-dimensional feature vector (N=102). (c) Candidates sorted by decreasing SVM confidence scores (row-wise, top to bottom). A sample subset of TB-object candidates with corresponding probabilities of being TB bacilli are shown at the output. These object-level SVM scores are subsequently used to determine slide-level diagnosis.

of $0.25\mu m$/pixel. Moreover, we empirically found that 24x24 patches provide a good tradeoff between capturing the TB-object candidates in their entirety and avoiding extraneous neighboring objects.

3.2 Representation of TB-Object Candidates

Next, we characterize each TB-object candidate using Hu moment invariants [27], geometric and photometric properties, and histograms of oriented gradients (HoG) features. The Hu moment, photometric, and HoG features are calculated from the grayscale patch, whereas geometric properties are determined from a binarized version of the image patch. We calculate eight Hu moment features, fourteen geometric and photometric descriptors, and eighty HoG features. We thus obtain a 102-dimensional feature vector representing the appearance of each candidate TB object. In Section 5.1, we discuss object-level classification results for various subsets of these features.

3.2.1 Hu Moment Invariants

Hu moment invariants are based on normalized central moments and provide a succinct object-level representation that is invariant to rotation, translation, and scaling [27]. In this study, we consider eight Hu moments (first seven and the eleventh), as motivated by Forero *et al.* [4], [15]. These moment invariants interpret a binarized or grayscale image as the probability density function of a 2-D random variable. Whereas Forero *et al.* calculated the Hu moment invariants from binarized versions of the TB-objects, we calculate them directly from the grayscale patches mentioned above.

Forero *et al.* excluded moments four through seven on the basis that they will often be close to zero because of the symmetric shape of bacilli. We, however, include all eight moments and allow the classifier to determine the relative significance of the moments invariants. The Hu moments were calculated using the following equations, where $f(x, y)$ represents a binary or grayscale image with centroid (x_c, y_c) [4], [15], [28]:

$$
\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{21})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} - \eta_{21})^2 + (\eta_{21} - \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{21})(\eta_{30} + \eta_{21})[(\eta_{30} + \eta_{12})^2 \\
&\quad - 3(\eta_{21} - \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
\phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
\phi_{11} &= \eta_{40} - 2\eta_{22} + \eta_{04}
\end{aligned}
$$

where $\eta_{rs} = \frac{\mu_{rs}}{\mu_{00}^{\gamma}}$ and $\gamma = \frac{r+s}{2} + 1, r + s = 2, 3, ..., \infty$.

$$\mu_{pq} = \sum_{xy} (x - x_c)^p (y - y_c)^q f(x, y) \tag{3.1}$$

3.2.2 Geometric and Photometric Features

In addition, we include fourteen geometric and photometric descriptors: area, convex area, eccentricity, equivalent diameter, extent, filled area, major/minor axis length, max/min/mean intensity, perimeter, solidity, and Euler number. The photometric features are calculated from grayscale TB-object candidates, while geometric descriptors are derived from binarized versions. Binarization here is achieved using Otsu's method [29] on the patch, which minimizes the variance within each of the two resulting pixel classes. We found renormalization at the patch-level prior to applying Otsu's method helpful in preventing saturated false positives at the edge of patches. Some patches contain multiple objects, in which case only the object closest to the center of the patch is used in calculating the geometric features. We describe several of these descriptors in more detail below.

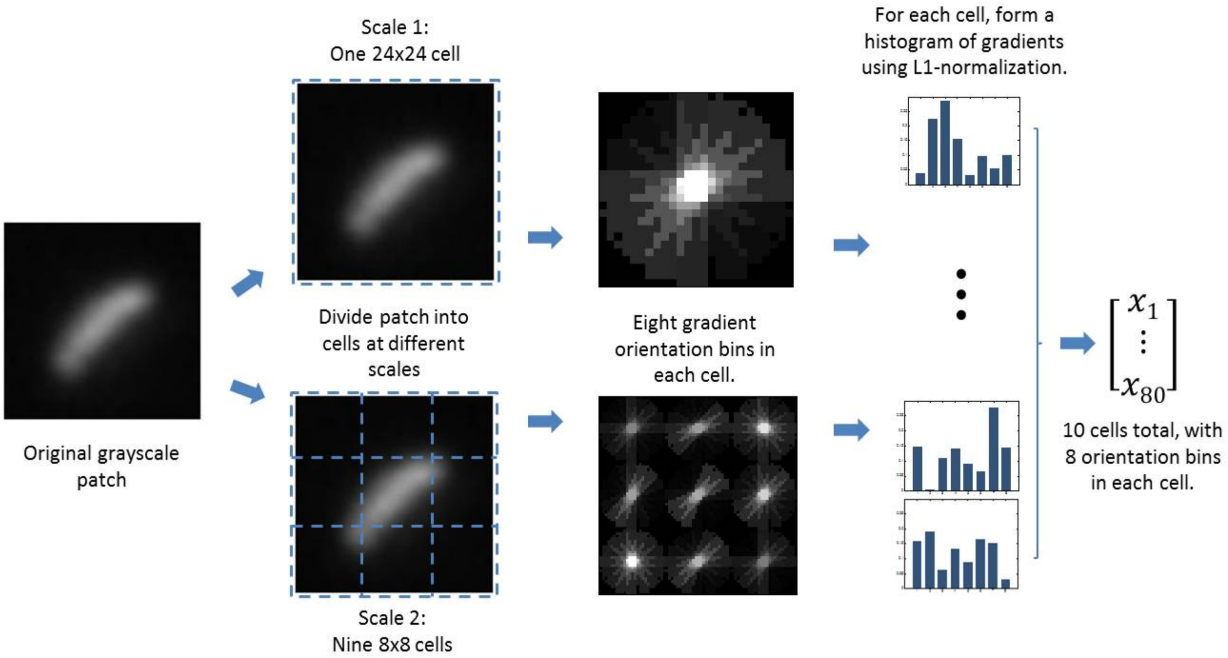


Figure 3.2. Visualization of HoG feature calculation. We divide each patch into cells at two scales, giving 10 cells total with 8 gradient orientation bins per cell (resulting in 80 feature values).

Feature Name	**Description**
Convex Area	Number of pixels in the smallest convex polygon that contains the object
Eccentricity	Ratio of distance between an ellipse's foci and its major axis length, where ellipse has the same second moment as the object
Equivalent Diameter	Diameter of a circle that has the same area as the object
Extent	Ratio between pixels in object and pixels in object's bounding box
Solidity	Proportion of pixels in the convex hull that are also in object
Euler Number	Difference between number of object(s) and number of holes in the objects (In our case, only one object is considered in each patch)

3.2.3 Histograms of Oriented Gradients

Finally, we incorporate histograms of oriented gradients to achieve robustness against variations in illumination across patches and images. Histograms of oriented gradients (HoG) are among the most widespread image descriptors used in the contemporary computer vision community [30], [31]. To calculate the HoG features, we first divide the image of interest into cells at various scales. Within each cell, we compute the gradient at each pixel and bin them into K different orientation bins. We then use L1-normalization in each cell to form a histogram of oriented gradients. Con-

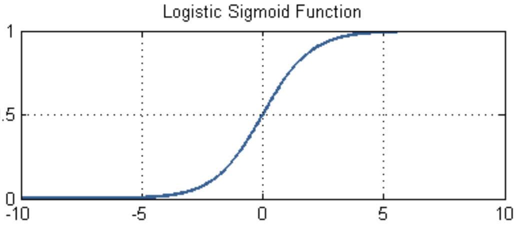


Figure 3.3. Logistic sigmoid function. Maps real numbers to the interval between 0 and 1.

catenating the histograms of all the cells, HoG gives rise to a KN-dimensional feature vector, where N is the total number of cells.

In our case, we extract HoG features from each patch at two scales, with one 24x24 pixel cell containing the entire patch and nine 8x8 pixel cells (Figure 3.2). Within each cell, we bin the gradients in 8 different orientation bins, resulting in a total of 80 gradient-based features. Note that HoG is not fundamentally orientation invariant, but the hope is that the training set provides exemplars for a representative set of bacilli orientations.

3.3 Classification of TB-Object Candidates

For object-level classification, we adopt a discriminative approach, focusing on logistic regression and support vector machines (SVMs) [32], [33]. Support vector machines, in particular, have enjoyed tremendous success in practice and hence become one of the most popular tools in machine learning.

3.3.1 Logistic Regression

Logistic regression is a simple linear classifier based on the logistic sigmoid function (Equation 3.2). Under this model, the posterior probability of a class C_1 is assumed to take the form of a logistic function whose argument is a linear function of the input feature vector x (Equation 3.3) [32].

$$\sigma(a) = \frac{e^a}{e^a + 1} = \frac{1}{1 + e^{-a}} \tag{3.2}$$

$$P(C_1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) \tag{3.3}$$

As seen in Figure 3.3, the logistic function maps any arbitrarily large real number to a value between 0 and 1, which can be interpreted as the probability of the input feature vector belonging to C_1. The vector w contains weights corresponding to each input feature value. Large positive weights indicate that a particular feature is a good indicator of association with the class of interest.

3.3.2 Linear Support Vector Machines

Support vector machines (SVM) are arguably one of the most effective and widely-used off-the-shelf classification tools [33], [34]. Intuitively, SVMs attempts to find an optimal hyperplane in the feature space that separates positive and negative classes, which correspond to TB-positive and TB-negative here (see Figure 3.4).

Throughout our discussion of SVMs, we will adopt the notation of [35], to which we refer the reader for more detailed descriptions. Let us denote $dist_+$ ($dist_-$) to be the smallest distance between the hyperplane and the closest positive (negative) sample. Further, we define the "margin" corresponding to that particular hyperplane to be $dist_+ + dist_-$. In the case of linearly separable data, the SVM attempts to find the hyperplane that maximizes the resulting margin.

Suppose our training data are constrained by the following inequalities:

$$\begin{gathered}\mathbf{x}_i \cdot \mathbf{w} + b \geq +1 \text{ for } y_i = +1 \\ \mathbf{x}_i \cdot \mathbf{w} + b \leq -1 \text{ for } y_i = -1\end{gathered}$$

or, equivalently,

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \ \forall i \tag{3.4}$$

where we have input feature vectors $\mathbf{x}_i$ and corresponding labels y_i.

If we consider the sample points for which equality holds in Equation 3.4, we have a pair of hyperplanes that are parallel to each other and between which no training points fall. Both of these hyperplanes have normal vector $\mathbf{w}$. Using geometry, it can be shown that $dist_+ = dist_- = 1/\|\mathbf{w}\|$ and hence the margin may be expressed as $2/\|\mathbf{w}\|$. Thus, to maximize the margin, we may equivalently minimize the objective function $\|\mathbf{w}\|^2$, given the constraints in Equation 3.4.

We now turn to the Lagrangian formulation of this optimization problem. The reason for doing so is two-fold: (1) the resulting constraints on the Lagrange multipliers are much easier to work with and (2) the training data show up only in the form of dot products, which enables us to generalize to the nonlinear case [35]. Introducing positive Lagrange multipliers, α_i (one corresponding to each inequality in Equation 3.4), we have the following Lagrangian:

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^{l} \alpha_i \tag{3.5}$$

We want to minimize L_P with respect to $\mathbf{w}$ and b. Simultaneously constraining the gradient of L_P with respect to $\mathbf{w}$ and b to go to zero results in the following:

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \tag{3.6}$$

$$\sum_i \alpha_i y_i = 0 \tag{3.7}$$

We may also consider the equivalent dual formulation of this convex optimization problem. Namely, we can maximize the following L_D with respect to α_i:

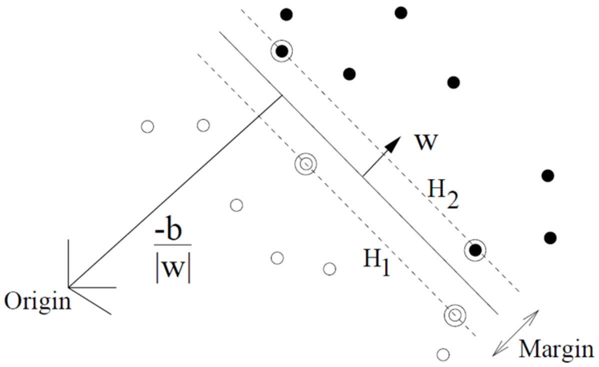


Figure 3.4. A schematic overview of a support vector machine, a max-margin classifier. Linearly separable case shown. Image from [35].

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{3.8}$$

In summary, the support vector algorithm seeks to maximize L_D (or equivalently minimize L_P) with respect to α_i, given the constraints set forth by Equation 3.7 and $\alpha_i \geq 0$. The solution is given by Equation 3.6.

To address the case of nonseparable data, we relax the constraints in Equation 3.4 by incorporating positive slack variables ξ_i. Our new objective function is thus $\|\mathbf{w}\|^2/2 + C(\sum_i \xi_i)^k$ instead of just $\|\mathbf{w}\|^2/2$. The penalty or cost of making erroneous classification decisions may be adjusted by the user-defined parameter C, with higher C corresponding to larger penalties. In the dual formulation, the nonseparable solution takes on the same form as in the separable case (Equation 3.6) apart from a modified constraint ($0 \leq \alpha_i \leq C$ in place of $0 \leq \alpha_i$).

3.3.3 Nonlinear Support Vector Machines

SVMs may be further generalized to decision functions that are not linear in the data. To do so, we first observe that the data only show up in dot product form, $\mathbf{x}_i \cdot \mathbf{x}_j$, in the training procedure (Equation 3.8). We then introduce a mapping Φ that takes the data to another space (usually some higher – perhaps even infinite – dimensional space). With this mapping, we note that we would have to calculate dot products of the form $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. We define a "kernel" function $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. Hence, the solution to our nonlinear SVM only depends on K, and the complicated mapping Φ need not be known explicitly. Much research effort has been devoted to finding effective kernel functions, but in computer vision applications, the Gaussian radial basis function (GRBF) and intersection kernel (IK) functions are most commonly used:

GRBF Kernel: $K(\mathbf{u}, \mathbf{v}) = exp(-\gamma||\mathbf{u} - \mathbf{v}||^2)$
Intersection Kernel: $K(\mathbf{u}, \mathbf{v}) = \sum_i min(\mathbf{u}_i, \mathbf{v}_i)$

Nonlinear SVMs may be computationally expensive because one must calculate the kernel function of a given test vector with each of the support vectors. Nevertheless, Maji *et al.* introduced an efficient method of implementing additive kernel (e.g., IK) SVMs such that the runtime complexity of the classifier depends logarithmically on the number of support vectors rather than linearly [36]. Further, an approximate classifier is shown to have constant runtime and space requirements regardless of the number of support vectors. In our experiments, we found that running the exact version of IKSVM yielded acceptable running times. Nevertheless, these recent advances in computing nonlinear SVMs enable us to consider nonlinear methods even for computionally-limited applications.

In our study, we train object-level classifiers and optimize model parameters using slide-level performance. Input feature vectors are normalized using maximum-minimum standardization, and we apply logistic regression to the output of the SVM to obtain output probabilities [37] indicating the likelihood of each object being a TB bacillus [37]. We consider logistic regression, linear SVM, and IKSVM classifiers with increasing discriminative power (and hence increasing computational cost) [34], [38]. With the confidence score outputs, one could provide a microscopist with a ranked list of potential TB objects for efficient diagnosis. Alternatively, one could establish a threshold based on an acceptable tradeoff point between recall and precision. These object-level confidence scores may also serve as the input to a second-level classifier that outputs image-level or slide-level decisions.

3.4 Slide-Level Classification

We survey a number of methods for determining slide-level diagnosis from object-level scores. Motivated by the way microscopists rate slides manually, we first propose taking the average of the top K object scores to obtain a final slide-level confidence score. Let us refer to this method as the direct average method. An immediately apparent alternative to the direct average method would be using a linear SVM with a K-dimensional input containing the top K object scores. Let us call this method the K-top linear SVM method.

One possibility for determining the object-level scores is to concatenate all 102 features together and train an object-level classifier on the full feature vector. However, as will be shown in Chapter 5, different feature subsets provide complementary information about the TB-object candidates. Instead of combining all 102 features together in one pool, one may consider grouping the features into multiple categories in a way that incorporates our knowledge of the construction of these features. In our case, we split the features into two categories: MPG (moment, photometric, and geometric features) and HoG. One may envision various multi-level SVM approaches that strategically combine the complementary information from these two sets of object descriptors.

In particular, we explored variations of the following three schemes (which we term Method I, II, and III). In Method I, we combine the MPG and HoG information at the object-level and proceed with our direct-average pipeline at the slide level. That is, we first train two object-level classifiers, one using the MPG features and another using HoG features, such that each object is assigned two scores. We then train a linear SVM that takes the two scores corresponding to each object and outputs a final object-level score. Once each object has one score, we proceed as before with the direct average method.

In Method II, we again begin by running two pipelines in parallel, one for MPG features and another for HoG features. This time, however, we run both pipelines all the way through the direct average method. Each of the pipelines thus outputs a final score at the slide-level. We then train a second-level linear SVM to combine the two slide-level scores to return a final single slide-level score.

Method III runs two separate pipelines through the object-level score stage. Then, we consider the top K scores from each of the two pipelines and concatenate them to form a 2K-dimensional vector corresponding to each slide. We train a linear SVM followed by logistic regression to obtain a single slide-level score from the 2K-dimensional input. We further explored taking the top K scores from each of the two object-level SVMs along with the corresponding scores from the other object-level SVM. Doing this for each of the SVM pipelines would result in a 4K-dimensional array of scores for each slide.

In our experiments with the Uganda dataset, these methods of combining MPG and HoG feature information do not result in any significant slide-level performance gain. Collection of more data could help in further discerning if there is any substantial potential gain from these multi-level methods. For the remainder of the study, we opt to use the K-top direct average method because of its computational efficiency (and comparable performance).

Chapter 4

Data Collection and Performance Metrics

4.1 Dataset and Ground Truth

Our dataset consists of sputum smear slides collected in the field in Uganda [2], [8]. Smears were derived from sputum specimen from patients with cough that lasted between 2 and 24 weeks. These patients were part of the International HIV-Associated Opportunistic Pneumonias (IHOP) study in Mulago Hospital of Kampala, Uganda (2007-2008). Each patient provided two sputum samples for the study, one for FM-based diagnosis and one for culture diagnosis. Technicians at Uganda's National Tuberculosis Reference Laboratory (NTRL) prepared smears from the FM samples, staining them with auramine-O.

Monochromatic fluorescence images of the smears were taken using the CellScope device, a novel mobile microscope with an 8-bit monochrome CMOS camera. CellScope uses a 0.4NA objective, and the fluorescence emission has $\lambda \approx 500$nm, giving a Rayleigh resolution of 0.76μm. See Section 1.3 for more details about the CellScope digital microscope. Each CellScope image is 1944x2592 pixels, covering a 640x490 μm field of view at the smear. We use 594 CellScope images (296 TB-positive, 298 TB-negative), which correspond to 290 patients (143 TB-positive, 147 TB-negative). A human annotator labeled high-confidence and borderline TB-objects in a subset of the culture-positive CellScope images (92 of the 296 images), resulting in 1597 positive TB objects.

In addition, for all 290 slides, we have slide-level diagnosis results determined by (1) the culture method, (2) Ugandan technicians using traditional LED FM, and (3) CellScope human readers. Experienced technicians in Uganda cultured each patient's first sputum sample using Lowenstein-Jensen media, providing the gold-standard diagnosis. Using each patient's second sample, the technicians then diagnosed patients based on smear readings with a professional LED FM (400x magnification). Finally, these smears were transported to the U.C. Berkeley under controlled conditions, where they were restained and read by human readers using the CellScope device.

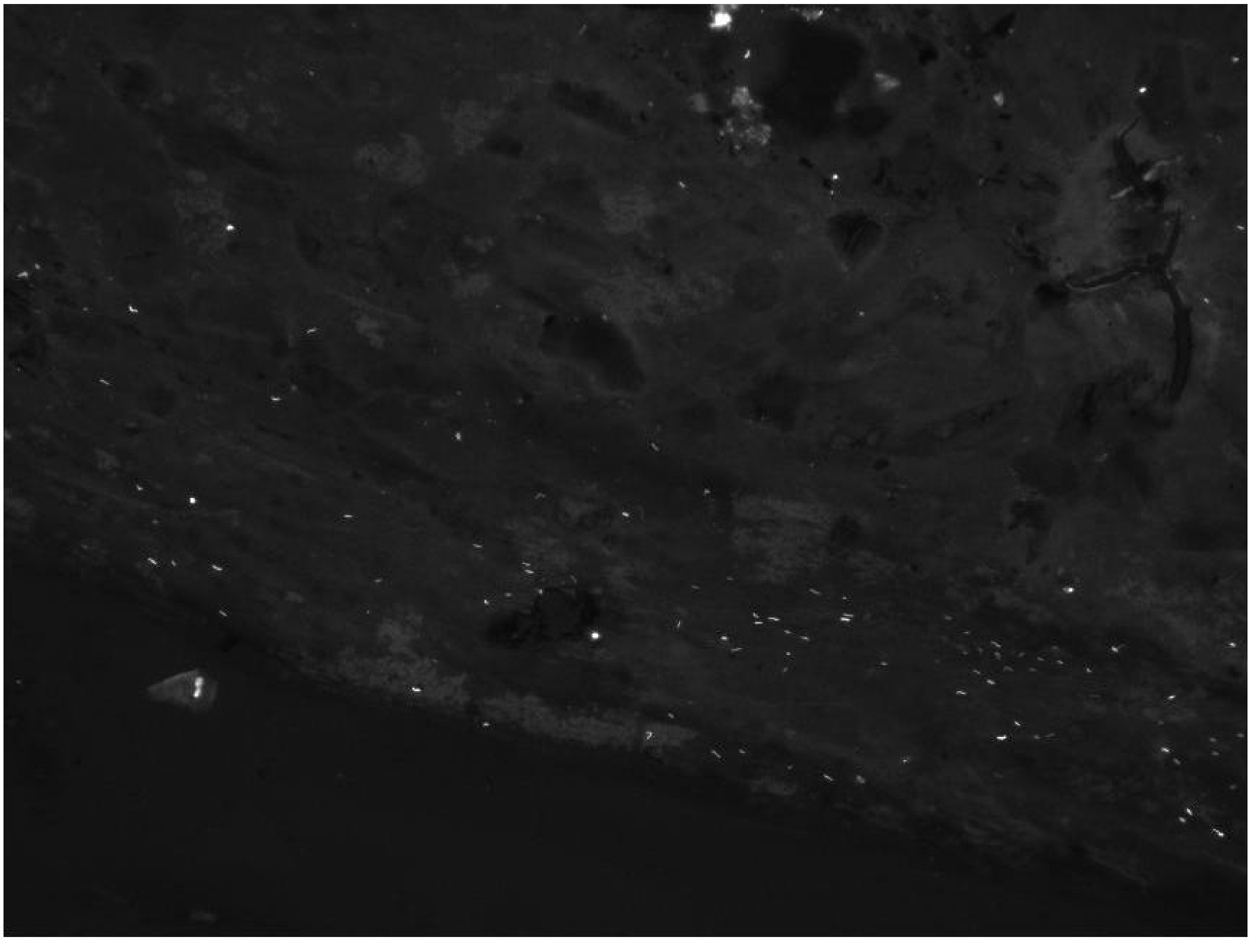
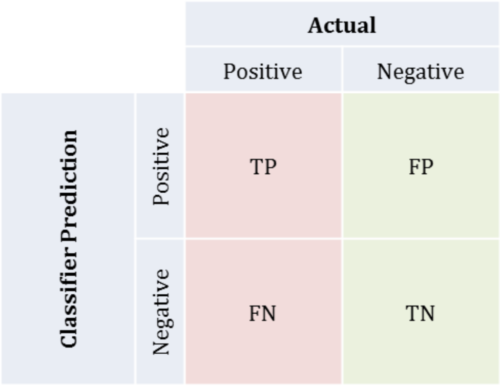


Figure 4.1. A fluorescent CellScope image of a sputum smear collected in Uganda. Taken using 0.4NA objective, with $\lambda = 500$nm, for a Rayleigh resolution of 0.76μm. Acquired using an 8-bit monochrome CMOS camera digitally sampling the image at above Nyquist frequency.

Data Entity	Positive	Negative	Total
Slides (Sputum Smears)	143	147	290
Images	296	298	594
TB Bacilli Objects	1597	–	–

Table 4.1. Overview of dataset used in this study.



		Actual	
		Positive	Negative
Classifier Prediction	Positive	TP	FP
	Negative	FN	TN

Figure 4.2. Overview of true positive (TP), false positive (FP), false negative (FN), and true negative (TN).

Performance Metric	Definition
Recall	$\frac{TP}{TP+FN}$
Precision	$\frac{TP}{TP+FP}$
Sensitivity	$\frac{TP}{TP+FN}$
Specificity	$\frac{TN}{TN+FP}$

Table 4.2. Performance metrics defined in terms of values from Figure 4.2.

We plan to make our dataset and human annotations publicly available. We will also release our algorithm and evaluation code, which we hope will provide a helpful reference for future quantitative study of TB detection.

4.2 Performance Metrics

We present our experimental results using two sets of performance metrics: Recall/Precision and Sensitivity/Specificity (see Table 4.2), which are widely used in the computer vision and medical communities, respectively. They are briefly described here:

Recall refers to the fraction of positive objects correctly classified as positives. Range: $[0, 1]$, with 1 being perfect performance.

Precision refers to the fraction of objects classified as positive that are indeed positive. Range: $[f, 1]$, where f the fraction of total objects that are positive, and 1 still corresponds to perfect performance in terms of precision.

Sensitivity is the same as Recall.

Specificity refers to the fraction of negative objects correctly classified as negative. Range: $[0, 1]$, with 1 being perfect performance in terms of specificity.

Recall/Precision are more appropriate for gauging object-level performance because our negative class size is much larger than the positive class size. At the slide-level, our data has balanced class sizes, so we consider both Recall/Precision and Sensitivity/Specificity. In this study, we optimize over Average Precision (AP) at the slide level, which places equal weight on Recall and Precision.

Often in practice it is more useful to have *either* very high specificity *or* very high sensitivity (rule-in or rule-out value, respectively) rather than moderately high values for both. In these cases, one could consider optimizing over the maximum F_β-measure, defined as

$$F_\beta = (1+\beta^2)\frac{Precision * Recall}{(\beta^2 * Precision) + Recall} \tag{4.1}$$

where $\beta < 1$ gives more weight to Precision than Recall ($\beta = 1$ gives equal weight to both while $\beta = 0.5$ gives Precison twice the weight of Recall). An alternative to the F_β-measure is to preset the specificity (sensitivity) value and determine the maximum achievable sensitivity (specificity) value. This provides a more concrete interpretation of diagnostic performance for the medical community. For example, acceptable performance for rule-in tuberculosis diagnostic tests at the patient level is sensitivity of ≥ 0.5 for specificity of 0.96.

Chapter 5

Experimental Results and Discussion

5.1 Object-Level Evaluation

For the object-level classification task, we use the subset of the TB-positive images for which we have human annotations and all TB-negative images. Applying our object identification procedure, we retain 98.8% of the positive TB objects in the dataset after the first step. All objects identified in TB-negative images are considered negative objects. This results in 1597 positive and 34948 negative objects, which correspond to 390 images (92 positive and 298 negative). Sample positive and negative objects are shown in Figure 5.1.

We generate five random training-test splits with our object-level data, one of which is reserved for model parameter selection. The four remaining splits are used to gauge the robustness of the algorithm's results. We perform systematic ablation studies as summarized in Figure 5.2, dividing the features into two subsets: MPG (Hu moment, photometric, and geometric features) and HoG features. That is, we train a classifier using each of the feature subsets as well as on a combined feature set in which we concatenate the MPG and HoG features to obtain 102-dimensional feature

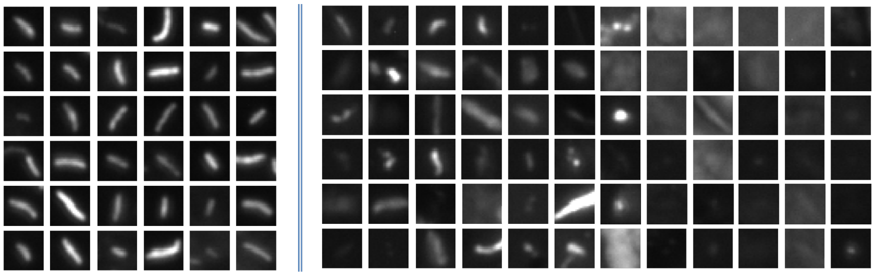
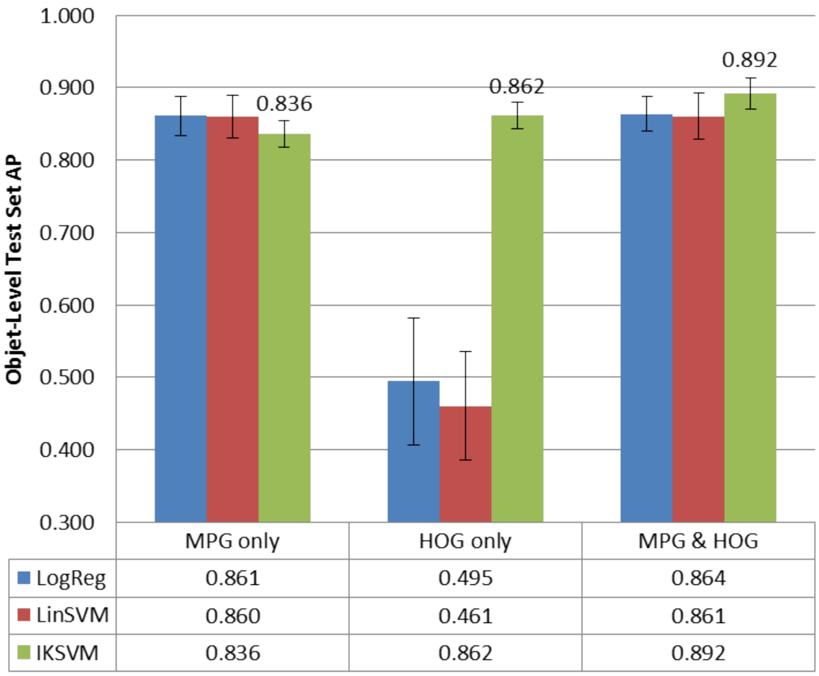


Figure 5.1. Sample positive (*left*) and negative (*right*) objects from our test set. Objects were detected in the first step using a white-top transform and cross-correlation with a Gaussian kernel.



	MPG only	HOG only	MPG & HOG
🟦 LogReg	0.861	0.495	0.864
🟥 LinSVM	0.860	0.461	0.861
🟩 IKSVM	0.836	0.862	0.892

Figure 5.2. Object-level test set AP across different (i) classifiers (logistic regression, linear SVM, and IKSVM) and (ii) feature subsets. Two categories of features: MPG (Hu moment, geometric, and photometric features) and HOG (histograms of oriented gradients features).

vectors. We also consider three types of discriminative classifiers: logistic regression, linear SVMs, and intersection kernel SVMs (IKSVMs).

For the logistic regression and linear SVM cases, the MPG feature subset are substantially more discriminative than the HoG features. Recall that TB-bacilli show up in different orientations and that HoG features are sensitive to changes in orientation. Thus, it is not surprising that the HoG features have low performance when used with linear classifiers.

With the nonlinear IKSVM, we see that the MPG and HoG features provide complementary information. Including both MPG and HoG features enable us to achieve the highest object-level performance ($89.2\% \pm 2.1\%$ over the four test sets), higher than the MPG only and HoG only performance ($83.6\% \pm 1.8\%$ and $86.2\% \pm 1.8\%$, respectively).

An array of the test objects sorted by descending output SVM confidence scores is shown in Figure 5.3. As expected, the objects with the highest SVM confidence scores exhibit the characteristic rod-like morphology of TB bacilli. Objects resulting in low confidence scores are generally small “hot pixels” or fluctuations in the background staining of the smear sample. Note that there are a few high-confidence, rod-shaped objects that are negative objects. These objects may be due to non-TB bacilli that are visually similar to TB bacilli. Alternatively, they may correspond to TB-positive but culture-negative samples (culture, though the detection gold standard, is not perfect).

Veropoulos *et al.* explored a variety of algorithms and evaluated performance at the object level. We consider their results from direct sputum smears because these slides most closely resemble the quality of the Uganda smears used in our study [14]. For this part of their study, Veropoulos *et al.*

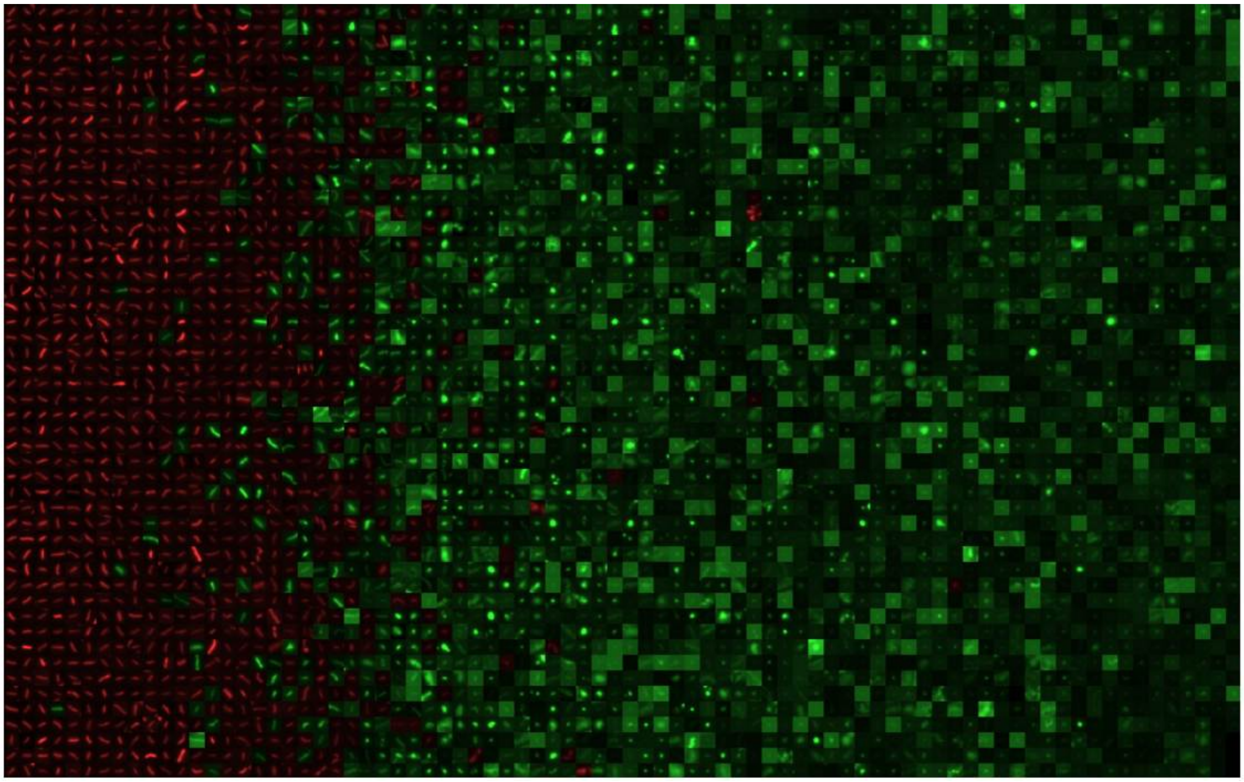


Figure 5.3. Test set objects sorted by their SVM output confidence scores in descending order (column-wise, from left to right). Red boxes correspond to objects that were labeled as TB-positive objects by the CellScope human reader. As expected, we generally observe higher confidence scores for objects that are rod-shaped. With decreasing confidence values, we see more 'hot pixel' objects and broad fluctuations (background staining artifacts). Best viewed in color.

Object-Level Classifier	Slide-Level AP(%)	Slide-Level AS(%)	Slide-Level $F_{0.5}$-measure(%)
Logistic Regression	91.4±0.5	87.1±1.2	87.0±1.3
Linear SVM	91.1±1.2	86.4±1.3	87.1±1.3
IKSVM	92.3±0.9	88.0±1.3	87.9±1.3

Table 5.1. Slide-level performance for different types of object-level classifiers. Performance metrics listed are Average Precision (AP), Average Specificity (AS), and $F_{0.5}$-measure, where $F_{0.5}$-measure is calculated using recall and precision. Slide-level decision determined from object-level scores using direct averaging method.

used 15 direct sputum smears (corresponding to 3509 positive objects and 680 negative objects). They measured performance in terms of the area under the ROC curve, which was defined as (1-specificity) vs. sensitivity. The multi-layer neural network (scaled conjugate gradient algorithm) yielded the best results, with an area under the curve of 0.892. At the object level, the area under our algorithm's (1-specificity) vs. sensitivity curve is 88.9%±1.3%. Note that these performance metrics were obtained from running each algorithm on each group's respective dataset, which were acquired by significantly different microscopes. Unfortunately, the code and dataset used in Veropoulos' study are not available publicly, so a fair, direct comparison is not possible.

5.2 Slide-Level Evaluation

We also consider algorithm performance at the slide level, which is more relevant to practical diagnosis. (Our Uganda dataset has one slide per patient, but in general it is common to have multiple slides for a single patient.) Because our dataset includes culture results at the slide level, doing so frees us from the constraints of human reader-based ground truth. We train an object-level classifier as before, using the slide-level performance as the optimizing criterion for parameter selection. As mentioned in Section 3.4, one may consider other methods of determining slide-level performance from object-level performance.

We found that the various methods resulted in similar performance with our current dataset, so we opt for the direct average method because of its low computational cost. For each slide, we gather the output SVM confidence scores of all the objects and average the top K object confidence scores, where $K = 3$ is chosen via validation experiments. We classify the *slide* as positive if the averaged score falls above a given threshold. By varying this threshold, we obtain a Recall-Precision curve such as shown in Figure 5.4. We again use five random training-test splits, with one of the splits reserved for object-level classifier parameter selection and the other four to estimate the robustness of the results.

We consider logistic regression, linear SVM, and IKSVM, and find that all three classifiers yield high slide-level performance (Table 5.1). We adopt the IKSVM approach because it gives slightly better performance (at both the object and slide levels). On the test sets of the remaining four splits, we achieve an Average Precision of 92.3%±0.9%, Average Specificity of 88.0%±1.3%, maximum $F_{0.5}$-measure of 87.9%±1.3%, and maximum F_1-measure of 84.9%±2.4%.

Method	AP(%)	AS(%)	Max $F_{0.5}$-meas(%)	Max F_1-meas(%)
Standard FM Readers	-	-	89.2±1.7	88.3±1.1
CellScope Readers	-	-	82.9±1.8	85.9±1.3
IKSVM Approach	92.3±0.9	88.0±1.3	87.9±1.3	84.9±2.4
Baseline	79.7±3.3	71.9±4.2	74.3±2.1	78.8±1.8

Table 5.2. Comparison of our IKSVM-based algorithm's slide-level performance to that of (1) human readers using conventional FM, (2) human readers using CellScope, and (3) the baseline method (GMM approach [4]). Average Precision, Average Specificity, Maximum $F_{0.5}$-measure, and Maximum F_1-measure across four test sets.

5.3 Slide-Level Comparison with Baseline and Human Readers

We compare our algorithm's slide-level performance against that of human readers and Forero's GMM approach. Experienced microscopists in Uganda also used standard fluorescence microscopes to read the smears, achieving an $F_{0.5}$-measure of 89.2%±1.7% and F_1-measure of 88.3%±1.1% across the four test sets. Human readers then inspected CellScope images from the same patients visually and classified each slide with a binary positive-negative decision. These readings resulted in an $F_{0.5}$-measure of 82.9%±1.8% and F_1-measure of 85.9%±1.3% across the four test sets. Finally, we train Forero's algorithm using our data, where the color filtering is reduced to intensity filtering because CellScope images are monochromatic [4]. Their method achieves Average Precision of 79.7%±3.3%, Average Specificity of 71.9%±4.2%, maximum $F_{0.5}$-measure of 74.3%±2.1%, and maximum F_1-measure of 78.8%±1.8% (see Table 5.1).

Our algorithm's slide-level performance is similar to that of CellScope human readers (F_1-measure of 84.9%±2.4% versus 85.9%±1.3%). In addition, experienced Ugandan technicians perform at a similar level: F_1-measure of 88.3%±1.1% ([2] suggests that the performance difference between CellScope readers and Ugandan technicans is not statistically significant). In the context of the CellScope dataset, our algorithm outperforms the GMM approach proposed in [4]. Note, however, that their GMM-based algorithm was developed using RGB images and incorporated color filtering whereas CellScope provides monochromatic images. Figure 5.4 presents the Recall/Precision curves across different methods for a sample training-test split. Again, our algorithm's performance is comparable to that of CellScope human readers and close to that of standard FM human readers. Our algorithm also achieves higher performance than Forero's GMM approach, with a higher fraction of true positives for most recall values.

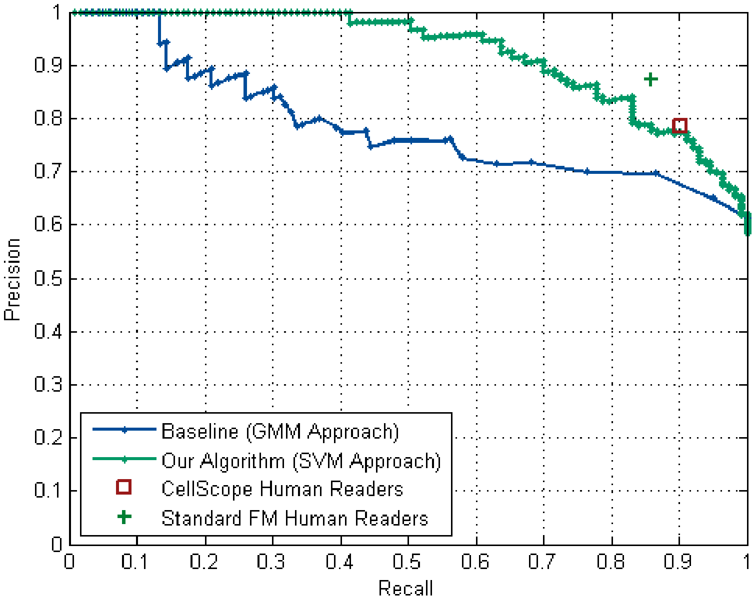


Figure 5.4. Slide-level Recall-Precision curves across different methods for one test set.

Chapter 6

User's Manual

This user's manual is intended to provide some guidance for running the code developed in this study. Both the dataset and code from this project will be publicly available through UC Berkeley's Computer Vision website (http://www.eecs.berkeley.edu/Research/Projects/CS/vision/). We thank our collaborators in Uganda and at the University of Caliornia, San Francisco, for making the data set available to the automated tuberculosis diagnosis research community.

The dataset consists of monochromatic TIF images of the Uganda slides taken by the CellScope device (see Chapter 4 for more information). As shown in Figure 6.1, the image directory structure encodes the culture results. There are 594 images (296 culture-positive, 298 culture-negative) corresponding to 290 (143 culture-positive, 147 culture-negative) patients. Among the 296 culture-positive images, 92 images have object-level human annotations.

We provide both evaluation code and training code, as described in Sections 6.1 and 6.2, respectively. The evaluation code is used to run the algorithm on test images. The training code re-trains the algorithm and should only be needed when there has been some fundamental change in the data (e.g., using images from a different device).

6.1 Using the Algorithm

At its core, the algorithm takes a fluorescent image as input, identifies potential TB bacilli objects in the image, and generates confidence scores corresponding to the candidate objects (indicating likelihood of an object being a bacillus). We provide two files for running this part of the algorithm: "runsingim_libsvm.m" and "runonim_libsvm.m."

The MATLAB *function* "runsingim_libsvm.m" takes a single fluorescent TIF image's file and path names as inputs. It generates an output CSV file with centroid coordinates and confidence scores corresponding to each candidate object.

The MATLAB *script* "runonim_libsvm.m" is similar but prompts the user to select input images through MATLAB's GUI rather than requiring the user to provide file and path names directly as input values. The output is generated in the same format as in "runsingim_libsvm.m," with one CSV file for each input image specified.

In order to run either of these files, one must provide an SVM model file. We include two sample MATLAB model files: "model_out_whog.mat" and "model_out_wohog.mat," which do and do not incorporate HoG features, respectively. These models were trained using the Uganda dataset outlined in Chapter 4.

Both "runsingim_libsvm.m" and "runonim_libsvm.m" automatically call a number of other functions, including some from the LibSVM implementation of SVMs [38] and Piotr Dollar's Image and Video Matlab Toolbox [39].

Using "runsingim_libsvm.m"

USAGE: runsingim_libsvm(fname, pname, dohog)

INPUTS:

- **fname**: TIF image file name (file extension included).
- **pname**: Directory that contains TIF image file.
- **dohog**: 1 to include HoG features (0 to exclude HoG features). Function will automatically load model_out_whog.mat or model_out_wohog.mat accordingly.

OUTPUT: CSV file containing centroids and scores corresponding to candidate objects (see below). Name of output file is "out_fname2.csv," where "fname2" is the same as the input "fname" without its file extension. Each row in this file corresponds to one candidate object.

- **Centroids**: Centroid coordinates of each candidate object, sorted by descending confidence score. Columns 1 and 2 in each row of the CSV file contains the candidate object's row and column centroid coordinates, respectively.
- **Confidence Scores**: Corresponding score indicating likelihood of candidate object being bacillus (Column 3 in each row of the CSV file).

Using "runonim_libsvm.m"

USAGE: Run "runonim_libsvm.m" script.

INPUTS: None. User will be prompted to select an input image files (or multiple input image files).

OUTPUTS: One CSV file generated for each input image. Same format as OUTPUT of "runsingim_libsvm.m."

6.2 Re-Training the Algorithm

Given a fundamental change in the data characteristics (e.g. new imaging device), it may be necessary to re-train the algorithm and generate a new model using representative training data (generate new "model_out_whog.mat" or "model_out_wohog.mat" file). To do so, one must provide the algorithm with object-level ground truth annotations (at least several hundred bacilli objects is preferable). See “Obtaining Object-Level Human Annotations” and “Image Directory Structure” Sections for more information.

To re-train the algorithm, one must run two functions: “runall.m” and “trainalg.m.” “runall.m” only needs to be run once for a given data set, after which “trainalg.m” may be run multiple times with various input sets. Note: If you would like to re-train the algorithm with our data set (i.e., just try different parameter settings), you may run “trainalg.m” directly (without first running “runall.m”).

The MATLAB *function* “runall.m” extracts candidate objects and features from all images and generates various MAT-files corresponding to subsets of images. These MAT-files contain the object and feature information required to run “trainalg.m.”

The MATLAB *function* “trainalg.m” uses the MAT-files generated by the “runall.m” to train a new object-level classifier. This classifier is optimized over some slide-level performance metric. In particular, we allow the user to choose from the following three slide-level performance metrics: (1) F_β-measure, (2) maximizing specificity for a particular value of sensitivity, and (3) maximizing sensitivity for a particular value of specificity.

Using “runall.m”

USAGE: Run “runall.m” function.

INPUTS: None. As long as the image directory structure is correct, the function should run automatically.

OUTPUTS: The following MAT-files are generated (first four pairs of MAT-files correspond to different image subsets):

- **allposobjs.mat and posfeats.mat:** Object and features extracted from culture-positive images with object-level human annotations. Only objects that coincide with human annotations are considered here.
- **allposobjs_igntag.mat and posfeats_igntag.mat:** Candidate objects and features extracted from culture-positive images with object-level human annotations. Here, all objects are considered regardless of whether or not they coincide with human annotations (i.e., ignore the object-level human annotations).
- **allposobjs_wotag.mat and posfeats_wotag.mat:** Candidate objects and features extracted from culture-positive images that never had object-level human annotations.
- **allnegobjs.mat and negfeats.mat:** Candidate objects and features extracted from culture-negative images.
- **nummissed.mat:** Records number of human-annotated objects missed (in each image) during candidate-object identification step.

Using “trainalg.m”

USAGE: trainalg(imdir, sel, param, dohog)

INPUTS:

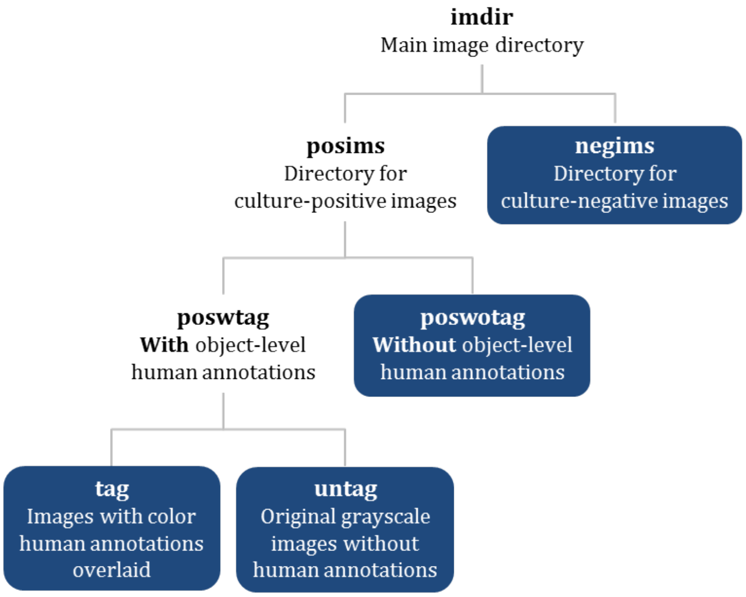


Figure 6.1. Image directory structure required to use code to re-train algorithm. Shaded directories contain image files while unshaded directories contain subdirectories.

- **imdir**: Directory containing TIF images. IMPORTANT: See "Image Directory Structure" Section for required directory structure.
- **sel**: Specifies optimization method. All these metrics refer to slide-level performance. Default value = 3.
  - "sel" = 1: Maximizes F_β-measure over sensitivity/specificity curve.
  - "sel" = 2: Maximizes specificity at a preset sensitivity value.
  - "sel" = 3: Maximizes sensitivitiy at a preset specificity value.
- **param**: parameter for optimization method specified by "sel." Default = 0.96.
  - If "sel" = 1, "param" corresponds to β. Ex: "param" = 1 or 0.5. See also Section 3.4.
  - If "sel" = 2, "param" corresponds to preset sensitivity value. Ex: "param" = 0.5.
  - If "sel" = 3, "param" corresponds to preset specificity value. Ex: "param" = 0.96.
- **dohog**: 1 to include HoG features (0 to exclude HoG features)

OUTPUT: Creates "model_out_whog.mat" or "model_out_wohog.mat," which contains the LibSVM model structure and parameter settings. The output model file should then be put in the same directory as "runsingim_libsvm.m" and "runonim_libsvm.m," which will call it when running the algorithm on new test images.

Obtaining Object-Level Human Annotations

The object-level human annotations in this study were generated in the MS Paint environment, but other annotations tools could certainly be used. The re-train algorithm code assumes the following annotation scheme: red, yellow, and blue annotations corresponding to high-confidence single-bacillus TB-object, borderline single-bacillus TB-object, and multi-bacilli TB-object, respectively. In our study, both high-confidence and borderline TB-objects were considered positive TB-objects. Multi-bacilli TB-objects were excluded in the training process, as they happen infrequently compared to single-bacillus TB-objects and exhibit varied shape characteristics.

Image Directory Structure

The re-train algorithm code assumes a particular image directory structure, as outlined in Figure 6.1. "imdir" specifies the main image directory and contains two subdirectories: "posims" and "negims," corresponding to culture-positive and culture-negative images. The "negims" directory contains the TIF files of culture-negative sputum smears. The "posims" directory further contains two subdirectories: "poswtag" and "poswotag," corresponding to images with and without object-level human annotations. TIF files of culture-positive sputum smears that have no object-level human annotations are located in the "poswotag" directory. Finally, the "poswtag" directory contains another two subdirectories: "tag" and "untag," corresponding to the images that have object-level tags from human readers. TIF files in the "tag" directory are images with the object-level human annotations (red, yellow, and blue labels), while the corresponding original, untagged TIF files are located in the "untag" directory.

Applying Code to Images Taken By Other Devices

To apply this code to images taken by other devices, a few parameters need to be changed prior to re-training the algorithm. In particular, the following need to be adjusted appropriately:

- **Patch Size**: Size of the square patch (in pixels). This value may be modified in the "System Parameters" section of the "trainalg.m" file. Patch should be large enough to accommodate a single TB-bacillus object (ideally without including neighboring bright objects). All the code developed in this study used a patch size of 24x24 pixels.
- **Gaussian Filter Parameters**: Size and standard deviation of Gaussian filter. These parameters may be set in the "xcorrwgauss.m" file, which is located in the "objlevel" folder. This Gaussian filter is used in the object candidate identification step and must be adjusted depending on the size of the bacilli in the image. For this study, the Gaussian filter standard deviation was chosen so that the Gaussian kernel was approximately the width of a typical bacillus object, and the size of the filter was taken to be 16x16 pixels.

Other than adjusting these parameters, the rest of the algorithm should be able to adapt automatically during re-training. Please note that the algorithm assumes the input images are monochromatic.

6.3 Status of Code Deployment

We plan to test the current version of our algorithm in Vietnam during Summer 2012. This effort is in collaboration with the University of California, San Francisco, and the Vietnamese National TB Program.

An initial graphical user interface (GUI) has been developed by Mike D'Ambrosio (Fletcher Lab at UC Berkeley). In the GUI, the user is presented with a global view of the image as well as the candidate objects sorted by their corresponding confidence scores from the algorithm. Hovering over the objects in the global view will highlight the bounding box around the candidate object and its score. The user is also free to navigate around the image and view zoomed/interpolated regions of interest.

Chapter 7

Conclusion

We propose an accurate and robust automated TB detection algorithm that may be used with low-cost, portable digital microscopes such as CellScope. Applying modern computer vision techniques to images captured by mobile microscopy could save lives in low-resource communities burdened by TB and suffering poor access to high-quality TB diagnostics. The sputum smears used in our study were collected in Uganda and provide a realistic dataset for algorithm training and evaluation.

Our algorithm first identifies potential TB objects and characterizes each candidate object using Hu moment, geometric, photometric, and oriented gradient features. We then classify each of the candidate objects using support vector machines. At the object level, we achieve Average Precision of 89.2% ± 2.1%. At the slide level, our algorithm performs as well as human readers, showing promise for making a tremendous impact on global TB diagnosis and care. Though this paper is largely focused on applying our algorithm to CellScope images, our method may be readily transferred to digital images taken by other FM devices. We plan to release our dataset, annotations, and algorithm/evaluation code, which we hope will provide helpful insights for future approaches to quantitative TB diagnosis.

References

[1] D. N. Breslauer, R. N. Maamari, N. A. Switz, W. A. Lam, and D. A. Fletcher, “Mobile phone based clinical microscopy for global health applications,” *PLoS ONE*, vol. 4, no. 7, p. e6320, 07 2009. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0006320

[2] A. Tapley, *Mobile Digital Imaging-based Microscopy for Tuberculosis Diagnosis, M.S. Thesis.* University of California, Berkeley, 2012.

[3] World Health Organization, “Global tuberculosis control 2011,” 2011. [Online]. Available: http://whqlibdoc.who.int/publications/2011/9789241564380_eng.pdf

[4] M. G. Forero, G. Cristóbal., and D. M., “Automatic identification of Mycobacterium tuberculosis by Gaussian mixture models,” *J of Microscopy*, vol. 223, no. 2, pp. 120–132, 2006. [Online]. Available: http://dx.doi.org/10.1111/j.1365-2818.2006.01610.x

[5] S. Dorman, “New diagnostic tests for tuberculosis: Bench, bedside, and beyond,” *Clin Infect Dis*, vol. 50, pp. Suppl 3:S173–7, 2010.

[6] C. Boehme, P. Nabeta, D. Hillemann, M. Nicol, S. Shenai, F. Krapp, J. Allen, R. Tahirli, R. Blakemore, R. Rustomjee, A. Milovic, M. Jones, S. O’Brien, D. Persing, S. Ruesch-Gerdes, E. Gotuzzo, C. Rodrigues, D. Alland, and M. Perkins, “Rapid molecular detection of tuberculosis and rifampin resistance,” *New England J of Med*, vol. 363, no. 11, pp. 1005–1015, 2010.

[7] C. Evans, “Genexpert-a game-changer for tuberculosis control?” *PLoS Med.*, vol. 8, p. e1001064, 2011.

[8] A. Cattamanchi, J. L. Davis, W. Worodria, S. den Boon, S. Yoo, J. Matovu, J. Kiidha, F. Nankya, R. Kyeyune, P. Byanyima, A. Andama, M. Joloba, D. H. Osmond, P. C. Hopewell, and L. Huang, “Sensitivity and specificity of fluorescence microscopy for diagnosing pulmonary tuberculosis in a high HIV prevalence setting,” *Int J Tuberc Lung Dis*, vol. 13, no. 9, pp. 1130–1136, 2010.

[9] L. Kivihya-Ndugga, M. van Cleeff, W. Githui, L. Nganga, D. Kibuga, J. Odhiambo, and P. Klatser, “A comprehensive comparison of Ziehl-Neelsen and fluorescence microscopy for the diagnosis of tuberculosis in a resource-poor urban setting.” *Int J Tuberc Lung Dis*, vol. 7, no. 12, pp. 1163–71, 2003.

[10] X. Heng, D. Erickson, L. R. Baugh, Z. Yaqoob, P. W. Sternberg, D. Psaltis, and C. Yang, “Optofluidic microscopy-a method for implementing a high resolution optical microscope on a chip,” *Lab Chip*, vol. 6, pp. 1274–1276, 2006. [Online]. Available: http://dx.doi.org/10.1039/B604676B

[11] D. Tseng, O. Mudanyali, C. Oztoprak, S. O. Isikman, I. Sencan, O. Yaglidere, and A. Ozcan, “Lensfree microscopy on a cellphone,” pp. 1787–1792, 2010. [Online]. Available: http://dx.doi.org/10.1039/C003477K

[12] Z. J. Smith, K. Chu, A. R. Espenson, M. Rahimzadeh, A. Gryshuk, M. Molinaro, D. M. Dwyre, S. Lane, D. Matthews, and S. Wachsmann-Hogiu, “Cell-phone-based platform for biomedical device development and education applications,” *PLoS ONE 6(3)*, p. e17150, 03 2011.

[13] K. Veropoulos, G. Learmonth, C. Campbell, B. Knight, and J. Simpson, “Automated identification of tubercle bacilli in sputum. a preliminary investigation.” pp. 277–282, 1999.

[14] K. Veropoulos, *Machine learning approaches to medical decision making.* U of Bristol, 2001. [Online]. Available: http://books.google.com/books?id=taWyGwAACAAJ

[15] M. G. Forero, F. Sroubek, and G. Cristóbal, “Identification of tuberculosis bacteria based on shape and color,” *Real-Time Imaging*, vol. 10, no. 4, pp. 251–262, 2004.

[16] M. Costa, C. Costa Filho, J. Sena, J. Salem, and M. de Lima, “Automatic identificatioin of mycobacterium tuberculosis with conventional light microscopy,” *30th Ann Int IEEE EMBS Conf.*, August 2008.

[17] R. Khutlang, S. Krishnan, R. Dendere, A. Whitelaw, K. Veropoulos, G. Learmonth, and T. S. Douglas, “Classification of mycobacterium tuberculosis in images of ZN-stained sputum smears,” *IEEE Trans. on Information Technology in Biomedicine*, vol. 14, no. 4, pp. 949–957, 2010.

[18] P. Sadaphal, J. Rao, G. Comstock, and M. Beg, “Image processing techniques for identifying Mycobacterium tuberculosis in Ziehl-Neelsen stains,” *Int J Tuberc Lung Dis*, vol. 12, pp. 579–582, 2008.

[19] G. Kubica, “Mycobacterium tuberculosis bacteria using acid-fast ziehl-neelsen stain; magnified 1000x.” 1979.

[20] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. San Diego, CA: Academic Press Inc., 1990.

[21] P. Rousseeuw, “Sillhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational Applied Mathematics*, vol. 20, pp. 53–85, 1987.

[22] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach.* Prentice Hall, 2011.

[23] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, “Objects in context,” in *Proc. ICCV*, 2007, pp. 1–8.

[24] T. Malisiewicz and A. Efros, “Recognition by association via learning per-exemplar distances,” in *Proc. CVPR*, 2008, pp. 1–8.

[25] C. Gu, J. Lim, P. Arbelaez, and J. Malik, “Recognition using regions,” in *Proc. CVPR*, 2009.

[26] J. Carreira, F. Li, and C. Sminchisescu, “Object Recognition by Sequential Figure-Ground Ranking,” *Int J of Computer Vision*, 2012, to appear.

[27] M.-K. Hu, “Visual pattern recognition by moment invariants,” *Information Theory, IRE Trans. on*, vol. 8, no. 2, pp. 179–187, February 1962.

[28] R. Gonzales and R. Woods, *Digital Image Processing*, 3rd ed. Pearson Education, 2008.

[29] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Trans on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[30] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *CVPR (1)*. IEEE Computer Society, 2005, pp. 886–893.

[31] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[32] C. Bishop, *Pattern Recognition and Machine Learning*. Springer Science and Business Media, 2006.

[33] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995, 10.1007/BF00994018. [Online]. Available: http://dx.doi.org/10.1007/BF00994018

[34] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “LIBLINEAR: A library for large linear classification,” *J of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[35] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.

[36] S. Maji, A. Berg, and J. Malik, “Classification using intersection kernel support vector machines is efficient,” *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, June 2008.

[37] J. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” in *Advances in Large Margin Classifiers*, 1999, pp. 61–74.

[38] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[39] P. Dollár, “Piotr’s Image and Video Matlab Toolbox (PMT),” http://vision.ucsd.edu/ pdollar/toolbox/doc/index.html.

Appendix A

Shape Descriptors

- **Generalized Color Moments:** The RGB Color moments of an image are defined as

  $$M_{pq}^{abc} = \int\int_{\Omega} x^p y^q [R(x,y)]^a [G(x,y)]^b [B(x,y)]^c dxdy \tag{A.1}$$

  where p+q is defined to be the order and a+b+c is defined to be the degree of the transform, $R(x,y)$, $G(x,y)$ and $B(x,y)$ represent the Red, Green and Blue channels of the color image, respectively [17].

- **Compactness:** Describes how "compact" the shape is, with the circle being the most compact shape. Defined as $\frac{Perimeter^2}{Area}$, where the perimeter may refer to inner, outer, or extended boundary. Using the outer boundary provides rotation, translation, and scale invariance [4].

- **Eccentricity:** Based on the best-fitting ellipse of the object. The ratio of the distance between the focal lengths and the major axis length. Circles have eccentricity of zero, and thus low eccentricity may be interpreted as the object being round. [4]

- **Fourier descriptors:** Assuming a K-point digital boundary in a 2-D image and starting at an arbitrary pixel belonging to the boundary, coordinate pairs are encountered in traversing the boundary in an arbitrary direction (counter-clockwise or clockwise). We can represent each pixel belonging to the boundary as

  $$s(k) = x(k) + jy(k) \tag{A.2}$$

  where $x(k)$ is the x coordinate and $y(k)$ is the y-coordinate of the k-th boundary pixel, which is equivalent to considering the x-axis and y-axis to be the real and imaginary axes respectively. In such a way, the 2-D boundary of the object is represented as a 1-D sequence. Once this dimension reduction has been carried out, the K-point Discrete Fourier Transform of $s(k)$ can be computed, which is given by:

  $$a(u) = \sum_{k=1}^{K-1} s(k) e^{\frac{-j\pi 2uk}{K}} \tag{A.3}$$

  The Fourier descriptors of the boundary are defined to be the DFT coefficients, $a(u)$, in this case (with a possible scaling by K) [28]. These descriptors are relevant to the bacilli-detection problem, since they carry shape information, i.e., in the inverse transform, they can reconstruct the shape of the object. To ensure invariance of the descriptors under affine transforms, the absolute value of the coefficients($|a(u)|$) were used instead, since $|a(u)|$ remains unchanged under translation and rotation of the boundary contour. The number of coefficients to be kept is determined by the classification accuracy of the nearest neighbor classifier [13], [16].

Appendix B

Feature Subset Selection Schemes

Often, a subset of the full feature set is selected to reduce the computational complexity of the classification task. Subset selection methods mentioned in Chapter 2:

- **Population-Based Incremental Learning (PBIL):** Chooses features in a probabilistic manner, using weights that favor a high evaluation figure-of-merit.
- **Correlation-Based Feature Selection (CFS):** Calculates the merit of using a subset of k features and tries to select features that are highly correlated to the current data vector itself but simultaneously uncorrelated with other features in the subset.
- **Sequential Floating Forward/Backward Selection (SFFS/SFBS):** Starts with a base feature and uses a number of forward or backward steps to find the next best feature. This is done until the number of desired best features has been reached.
- **Branch and Bound (B&B):** Uses an evaluation function to select the best subset of features by using a tree structure. It is described as an optimal feature selection algorithm since it cannot miss the best combination of features.

Appendix C

Classification Techniques

Below are more detailed descriptions of various classification and related methods mentioned in Chapter 2:

- **Bayes' Classifier:** Khutlang's segmentation approach relies on a Bayes' classifier, which assign an object the class whose probability density function dominates at the pixel position. For the Bayes' classifier, the probability density functions of the bacilli/non-bacilli classes were implicitly assumed to be Gaussian. The mean and covariance of the two classes in the training dataset were approximated, and the Bayes' classifier was then used to classify the query pixel as bacillus/non-bacillus [17].
- **Gaussian Mixture Models (GMMs):** A linear combination of multivariate Gaussian variables whose coefficients sum up to one. Intuitively, we can think of the coefficients as the a priori probability of each individual multivariate Gaussian in the mixture [13], [32]. Below, we see how a GMM was applied in Forero's work:

  Here, the probability of observing a sample feature vector x given that it belongs to cluster j of the bacillus class is defined as a 4-variate Gaussian distribution.

$$\begin{aligned} p(x) &= \textstyle\sum_{j=1}^{M} p(x|\Omega_j)p(\Omega_j) \\ p(x|\Omega_j) &= \tfrac{1}{\sqrt{2\pi|\Sigma_j|}} e^{-\left(\frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)\right)} \end{aligned} \tag{C.1}$$

- **K-Means Clustering:** A simple clustering algorithm. Number of clusters K is input parameter. Begin with K arbitrary initial cluster centers. Calculate the distance between each sample point and the initial cluster centers and assign each point to the closest cluster center. For each cluster, calculate the average and set that as the new cluster center. Again calculate the distance between each sample point and the new cluster centers. Reassign sample points if necessary (such that each point belongs to the cluster whose center it is closest to). Repeat distance calculations and cluster reassignments until no changes result from these steps [32].
- **K Nearest Neighbors Classifier (kNN):** Potential object-level classifier mentioned in Khutlang *et al.* . Predicts the classification of each query object by considering its k closest training example neighbors (bacillus/nonbacillus labels for training set are known). Proximity of neighbors is determined in terms of Euclidean distance [17].

- **Linear Regression Classifier:** Explored by Khutlang *et al.* as a potential pixel-level classifier for segmentation. Determines a linear mapping between training examples and their ground truth labels, which is in turn used to predict labels of new queries. The mapping is aims to minimize the error between classes (bacillus/non-bacillus) in the least-squares, Euclidean distance sense [17].
- **Minimum Error Bayesian Classifier:** We describe the method in which Forero used a minimum error Bayesian classifier [15]. After modeling the 4-dimensional feature space, Forero's group defined discriminant functions g (one associated with each cluster) and identified the cluster with the maximal discriminant function as given in equation (C.2).

  $$\begin{gathered} g_i(x) = p(x|\Omega_i)p(\Omega_i) \\ \underset{i=1\ldots M}{max}\, g_i(x) \end{gathered} \tag{C.2}$$

  They then invoked maximal discriminant cluster c's Bayesian threshold decision rule for the final classification. Each cluster's decision rule may be equivalently expressed more intuitively, in terms of the Mahalanobis distance between sample feature vector x and cluster c.

  $$\xi(x) = \begin{cases} \Omega_c & if \quad g_c(x) > t_c \\ \Omega_0 & if \quad g_c(x) < t_c \end{cases} \tag{C.3}$$

  $$(x-\mu_c)^T \Sigma_c^{-1} (x-\mu_c) < -2t_c - log\,|\Sigma_c| + 2log(p(\Omega_c)) \tag{C.4}$$

- **Neural Networks:** Modeled after the way neurons in the brain function, neural networks provide a method of classification that does not require density estimation (such as in the case Forero's GMM and Bayesian classifier). Multilayer feed-forward neural networks like those used in Veropoulos' work consist of layers of nodes, with each node in a given layer outputting and feeding into every neuron in the following layer. The layers between the input and output layers are often called the hidden layers. Each node acts like a perceptron model, which calculates a weighted sum of all the incoming signals and passes the resulting sum through a threshold or activation function (e.g., a sigmoid function). This activation function is represents synaptic behavior in the brain, and is typically the same for all the nodes in a given network. When an input pattern is fed into the network, it is matched with class i if the i-th output is high and other outputs are low. Training a neural network consists of finding the appropriate weights used in calculating the sums. Various learning rules for neural networks have been proposed, with error back-propagation and scaled conjugate gradient being among the most common [13], [28].
- **Quadratic Discriminant Classifier:** Another alternative for a pixel-level segmentation classifier explored by Khutlang *et al.* . Similar to linear regression, except that the mapping between training examples and their ground truth labels is quadratic. The class mean and covariance matrices (mean and covariance of the training data set for bacilli and non-bacilli classes separately) are used to determine the classification boundaries [17].
- **Support Vector Machines:** See Chapter 3.

Appendix D

Performance Metrics

Common performance metrics used in the computer vision and medical communities:

- **Recall:** Number of positive objects correctly classified/Number of actual positive objects. Range: $[0, 1]$, with 1 being perfect performance in terms of recall.
- **Precision:** Number of positive objects correctly classified/Number of objects classified as positive. Range: $[f, 1]$, where f the fraction of total objects that are positive, and 1 still corresponds to perfect performance in terms of precision.
- **Sensitivity:** Same as recall. Also known as true positive rate or detection rate. Range: $[0, 1]$, with 1 being perfect performance in terms of sensitivity. [4], [13]
- **Specificity:** Number of negative objects correctly classified/Number of actual negative objects. Also known as true negative rate. Same as 1-(false alarm rate). Range: $[0, 1]$, with 1 being perfect performance in terms of specificity. [4], [13].
- **Overall Accuracy:** Correctly classified objects/Total objects. An acceptable metric for overall performance of a binary classifier if (1) class distributions are relatively constant and balanced and (2) the error costs of false alarms and missed detections are equal. Range: $[0, 1]$, with 1 being perfect performance. [13].

A few other performance metrics used by Khutlang *et al.* for evaluating segmentation:

- **Common Rate:** Number of correctly classified pixels in the reference object. May be converted to percentage of correctly classified pixels by normalizing with the number of total object pixels in the reference image [17].
- **Difference Rate:** Number of pixels that (1) belong to a bacillus object in the ground truth image but are not segmented as bacillus pixels or (2) belong to the background in the ground truth image but are labeled as bacillus pixels in the segmentation process. May be converted to percentage of incorrectly classified pixels by normalizing with the union of (1) ground truth bacillus pixels and (2) segmented bacillus pixels [17].