Automated Fluorescence Microscopy Determination of Mycobacterium Tuberculosis Count via Vessel Filtering

Swazoo Claybon III

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science

In

Electrical Engineering

Alfred Wicks, Chair
Penelope Muelenaer
Aloysius Beex

April 19, 2017

Blacksburg Virginia

Keywords: Tuberculosis, Frangi Filter, Image Processing, Hessian

Copyright © 2017 by Swazoo Claybon III. All rights reserved
Automated Fluorescence Microscopy Determination of Mycobacterium Tuberculosis Count via Vessel Filtering

Swazoo Claybon III

ABSTRACT

Tuberculosis (TB), a deadly infectious disease caused by the bacillus Mycobacterium tuberculosis (MTB), is the leading infectious disease killer globally, ranking in the top 10 overall causes of death despite being curable with a timely diagnosis and the correct treatment [3]. As such, eradicating tuberculosis (TB) is one of the targets of the Sustainable Development Goals (SDGs) for global health as approved by the World Health Assembly (WHA) in 2014 [2,3].

This work describes an automated method of screening and determining the severity, or count, of the TB infection in patients via images of fluorescent TB on a sputum smear. Using images from a previously published dataset [9], the algorithm involves a vessel filter which uses the second derivative information in an image by looking at the eigenvalues of the Hessian matrix. Finally, filtering for size and by using background subtraction techniques, each bacillus is effectively isolated in the image.

The primary objective was to develop an image processing algorithm in Python that can accurately detect Mycobacteria bacilli in an image for a later deployment in an automated microscope that can improve the timeliness of accurate screenings for acid-fast bacilli (AFB) in a high-volume healthcare setting. Major findings include comparable average and overall object level precision, recall, and F1-score results as compared to the support vector machine (SVM) based algorithm from Chang et al. [9]. Furthermore, this work’s algorithm is more accurate on the field level infectiousness accuracy, based on F1-score results, and has a high visual semantic accuracy.
Automated Fluorescence Microscopy Determination of Mycobacterium Tuberculosis Count via Vessel Filtering

Swazoo Claybon III

GENERAL AUDIENCE ABSTRACT

Tuberculosis (TB), a deadly infectious disease caused by the bacillus Mycobacterium tuberculosis (MTB), is the leading infectious disease killer globally, ranking in the top 10 overall causes of death despite being curable with a timely diagnosis and correct treatment [3]. Furthermore, 3.2 billion are part of an at risk population for contracting tuberculosis, yet 90% of TB related deaths occur in countries across Africa and other Low and Middle Income Countries (LMICs) [2]. This occurrence is, at least in part, due to a lack of the skilled human resources in LMIC laboratories necessary to scan large numbers of patient specimens and properly screen for TB.

Sputum smear microscopy of acid-fast bacilli (AFB) is essential in the screening of TB in high-prevalence countries. With the high rates of TB found in LMICs, there is a need to develop affordable, time-efficient alternatives for lab technicians to effectively screen large volumes of patients. This work describes the development of an automated method of screening and determining the severity, or count, of the TB infection in patients via images of fluorescent TB on a sputum smear using images from a previously published dataset [9].

The primary objective of this study was to write a program that can accurately detect tuberculosis in an image for a later deployment in an automated microscope that can improve the timeliness of accurate screening for AFB in a high-volume healthcare setting. Major findings include improved accuracy compared to that of Chang et al.’s machine learning algorithm that was used on this dataset [9].
ACKNOWLEDGEMENTS

This work would not have been possible without my parents, without their love and support, without their patience in the past and in the present, or without all of the steps they have guided me on up to this day. And I cannot forget my younger brother who kept me happy and healthy with our jokes. I just want to thank them with all of my heart.

I want to thank my grandparents in North Carolina who were always thinking of me and praying for me during my stay in Virginia and who never let me go home without food whenever I visited.

I would like to thank my Church back home in Tennessee and all of my family members who were very supportive of my Graduate education and never let a chance go by to mention it when I went home.

My research lab, mentors, and advisors at Virginia Tech have been a solid backbone for me while here at Virginia Tech. My research advisor Dr. Al Wicks has given me excellent advice in technical writing, managing a research project, and in life. He has been very supportive, and I would definitely not have been able to do this without him. He and the Virginia Tech Mechatronics Lab have broadened my worldview with interesting and exciting conversations and have been very helpful to bounce ideas off of.

Then there are the mentors in the group that I am in as well as my research partner. Dr. Andre Muelenaer, Dr. Penny Muelenaer, and Dr. John Bird have been my mentors, supervisors, and friends at the Pediatric Medical Device Institute at Virginia Tech, and Sahal Thahir was my research partner and good friend. They have been helpful through each phase of the research process, have guided my research, have directed me to helpful medical technicians, have gotten me samples of bacteria, and have been very supportive during my time here at Virginia Tech.

I would also like to thank Dr. Louis Beex who was willing to be on my committee at the last minute and who, along with Dr. Al Wicks, heightened my perception of the rigor of mathematics and clear understanding, something I really started to enjoy as time went on.
TABLE OF CONTENTS

1. INTRODUCTION ........................................................................................................1
   1.1 History of Tuberculosis .......................................................................................2
   1.2 Problem Statement ............................................................................................3
   1.3 Thesis Structure ................................................................................................3

2. LITERATURE REVIEW ...............................................................................................5
   2.1. Microscopy .......................................................................................................5
      2.1.1. Field of View ............................................................................................6
      2.1.2. Fluorescence Microscopy .........................................................................7
   2.2. Diagnosis of Tuberculosis ..............................................................................8
      2.2.1. General Approaches ................................................................................8
      2.2.2. Reporting Level of Infectiousness .............................................................9
   2.3. Image Processing Approaches ........................................................................10
      2.3.1. Fluorescence ............................................................................................11
      2.3.2. Bright Field .............................................................................................18
   2.4. Background Overview ....................................................................................24

3. ALGORITHM ...............................................................................................................26
   3.1. Color Spaces ....................................................................................................26
   3.2. Convolution ......................................................................................................28
   3.3. Bilateral Filter ..................................................................................................35
   3.4. Vessel Filtering ................................................................................................40
   3.5. Size Threshold ................................................................................................56
3.6. Background Subtraction ........................................................................................................62
3.7. Algorithm Flowchart ..............................................................................................................66

4. METHODS ..................................................................................................................................67

4.1. Object Level Analysis ..............................................................................................................69
4.2. Slide Level Analysis ................................................................................................................70

5. CONCLUSION ..............................................................................................................................74

REFERENCES ..................................................................................................................................75

APPENDIX A – Python Code Libraries and Source Code .................................................................78

APPENDIX B – Example Images ......................................................................................................79
LIST OF FIGURES

Figure 2.1: Generalized Microscope ................................................................. 5
Figure 2.2: Field of View – Reticle ................................................................. 6
Figure 2.3: Epifluorescence Microscope ......................................................... 7
Figure 2.4: Optics Filter ................................................................................. 8
Figure 2.5: Figure depicting a ‘length’ of a sputum smear .............................. 10
Figure 2.6: Generalized Flowchart ................................................................. 11
Figure 2.7: (Left) Generalized Flowchart (Right) Veropoulos’ Flowchart .......... 12
Figure 2.8: (Left) Original TB image (Right) processed image ....................... 13
Figure 2.9: (Left) Generalized Flowchart (Right) Forero’s Flowchart .............. 14
Figure 2.10: Sputum smear without counterstain ............................................ 15
Figure 2.11: Fluorescing Tuberculosis Bacilli .................................................. 15
Figure 2.12: (Left) Generalized Flowchart (Right) Forero’s Flowchart .......... 16
Figure 2.13: (Left) Original (Right) Processed with Top Hat Transform ........... 17
Figure 2.14: (Left) Generalized Flowchart (Right) Costa’s Flowchart ............. 18
Figure 2.15: Circled bacilli of Costa’s final computer vision analysis .............. 19
Figure 2.16: (Left) Generalized Flowchart (Right) Khutlang’s Flowchart .......... 20
Figure 2.17: (Left) Generalized Flowchart (Right) Sadaphal’s Flowchart ........ 21
Figure 2.18: (Left) Original Image (Right) Computer labeled image ................ 22
Figure 2.19: (Left) Generalized Flowchart (Right) Rachna’s Flowchart .......... 23
Figure 2.20: (a) K-means clustering (b) Region growing after K-means .......... 24
clustering (c) Otsu thresholding (d) region growing after Otsu thresholding
Figure 3.1: First four sections of the algorithm flowchart ............................... 26
Figure 3.2: RGB color cube ......................................................................... 27
Figure 3.3: (Left) RGB (Right) Grayscale TB image ....................................... 28
Figure 3.4: (Left) f [3 4 5] (Right) h [2 1] ......................................................... 29
Figure 3.5: Convolution calculation example ............................................... 29
Figure 3.6: Lateral inhibition example ............................................................ 30
Figure 3.7: Mean filter convolution with an image ......................................... 31
Figure 3.8: Mean filter convolution with an image f ..................................... 33
Figure 3.9: (Left) 8-bit output (Right) (0-255) normalized output ............................................34
Figure 3.10: (Left) original image (Right) (3x3) mean filtered image .................................34
Figure 3.11: (Left) original image (Right) (9x9) mean filtered image ..................................35
Figure 3.12: First four sections of the algorithm flowchart (Vessel) .............................................35
Figure 3.13: Gaussian kernel ..................................................................................................36
Figure 3.14: Sharp boundary image ..........................................................................................37
Figure 3.15: Gaussian range kernel ..........................................................................................38
Figure 3.16: Bilateral filter kernel ............................................................................................38
Figure 3.17: (a) Original image (b) parameter change 1 (c) parameter change 2 ........39
Figure 3.18 (Left) Vessel Filter output without bilateral filtering, ........................................40
(Right) Vessel filter output with bilateral filtering
Figure 3.19: Vessel filter flow chart .........................................................................................40
Figure 3.20: First four sections of the algorithm flowchart (Vessel) ........................................41
Figure 3.21: Gaussian kernel ..................................................................................................42
Figure 3.22: (Left) x partial Gaussian derivative kernel ............................................................42
(Right) y partial Gaussian derivative kernel
Figure 3.23: Left) original TB image (Right) y (7x7) Gaussian derivative.................................43
Figure 3.24: (Left) x derivative kernel (Right) y derivative kernel ........................................43
Figure 3.25: original image (b) x gradient image (c) y gradient image ........................................44
Figure 3.26: (Left) original TB image (Right) y (3x3) Prewitt derivative output ..................44
Figure 3.27: (Left) original TB image (Right) y (7x7) Prewitt derivative output ..................45
Figure 3.28: Second Derivative Gaussian Kernels .....................................................................46
Figure 3.29: (a) Original image (b) Prewitt Vessel Filter (c) Gaussian Vessel Filter ............47
Figure 3.30: (a) Original image (line) (b) y derivative (c) second y derivative ..................48
Figure 3.31: zero second derivative example ..........................................................................48
Figure 3.32: (a) Original image (bacilli) (b) y derivative (c) second y derivative .................49
Figure 3.33: Local minima and maxima of a parabola ............................................................50
Figure 3.34: principle directions of curvature visual .................................................................52
Figure 3.35: image of a bacillus with a square patch cropped out ........................................52
Figure 3.36: Hessian matrix eigendecomposition example for a square patch .....................53
Figure 3.37: Hessian eigenvectors ..........................................................................................54
Figure 3.38: section of the tuberculosis image .................................................................55
Figure 3.39: Beta and c parameter analysis on the vessel filter with tuberculosis ...............55
Figure 3.40: Kernel size analysis on the vessel filter with tuberculosis .................................56
Figure 3.41: Last four sections of the algorithm flowchart (size thresholding) ....................57
Figure 3.42: Binary image of a TB bacillus ........................................................................57
Figure 3.43: TB bacillus with rotated bounding box ............................................................58
Figure 3.44: TB bacillus with a normal bounding box .........................................................59
Figure 3.45: Rotated Calipers method depiction .................................................................59
Figure 3.46: Rotated bounding box of a bacillus .................................................................60
Figure 3.47: Pixel to um scaled bacillus dimensions ............................................................61
Figure 3.48: Output of bilateral and vessel filtering .............................................................62
Figure 3.49: Last four sections of the algorithm flowchart (background subtraction) ..........63
Figure 3.50: Output TB image without background subtraction for foreground verification ....64
Figure 3.51: Output TB image with background subtraction for foreground verification .......65
Figure 3.52: Full algorithm outline .....................................................................................66
Figure 4.1: Precision and Recall graphics ............................................................................68
Figure 4.2: Object level Precision, Recall, and F1-score metrics ........................................67
Figure 4.3: Field level precision and recall infectiousness accuracy .....................................72
Figure 4.4: Field level F1-score .........................................................................................73
LIST OF TABLES

Table 2.1: Level of Infectiousness for Fluorescence Sputum Smears with AFB ......................... 10
Table 3.1: Eigenvalue structure table .......................................................................................... 51
Table 4.1: Level of Infectiousness for Fluorescence Sputum Smears with AFB ......................... 70
Table 4.2: Level of infectiousness confusion matrix for this work’s rule based algorithm .......... 71
Table 4.3: Level of infectiousness confusion matrix for Chang’s machine learning algorithm .. 72

LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDG</td>
<td>Sustainable Development Goals</td>
</tr>
<tr>
<td>AFB</td>
<td>Acid-Fast Bacilli</td>
</tr>
<tr>
<td>TB</td>
<td>Tuberculosis</td>
</tr>
<tr>
<td>MTB</td>
<td>Mycobacterium Tuberculosis</td>
</tr>
<tr>
<td>LMICs</td>
<td>Low and Middle Income Countries</td>
</tr>
<tr>
<td>PMDI</td>
<td>Pediatric Medical Device Institute</td>
</tr>
<tr>
<td>WHO</td>
<td>World Health Organization</td>
</tr>
<tr>
<td>WHA</td>
<td>World Health Assembly</td>
</tr>
<tr>
<td>ZN</td>
<td>Ziehl-Neelsen</td>
</tr>
<tr>
<td>TST</td>
<td>TB Skin Test</td>
</tr>
<tr>
<td>LTBI</td>
<td>Latent TB Infection</td>
</tr>
<tr>
<td>LED</td>
<td>Light Emitting Diode</td>
</tr>
<tr>
<td>GMM</td>
<td>Gaussian Mixture Model</td>
</tr>
<tr>
<td>HOG</td>
<td>Histogram of Oriented Gradients</td>
</tr>
<tr>
<td>LSVM</td>
<td>Linear Support Vector Machine</td>
</tr>
<tr>
<td>1D</td>
<td>One Dimensional</td>
</tr>
<tr>
<td>2D</td>
<td>Two Dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>Three Dimensional</td>
</tr>
<tr>
<td>MoG</td>
<td>Mixture of Gaussian</td>
</tr>
<tr>
<td>kNN</td>
<td>k-Nearest Neighbor</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

The current Sustainable Development Goal, as proposed by the World Health Organization (WHO) ‘End TB’ strategy, calls for a “90% reduction in tuberculosis (TB) deaths and an 80% reduction in the TB incidence rate by 2030, compared with 2015” [3]. It should be noted that, for many people, the tuberculosis infection is latent and does not cause death, but these people are still included in the incidence rate due to the risk they pose for transmitting the bacteria or having it turn into an active tuberculosis infection that causes serious illness. These goals involve a worldwide transformation, spearheaded by the United Nations, with tuberculosis at the forefront due to TB being the leading cause of death via infectious disease [3]. Furthermore, the reason why diseases like TB have not been eradicated, despite having a complete cure for most strains of TB bacteria, is simply due to the cost required to screen, diagnose and effectively test those in low-resource areas, mostly including areas within Low and Middle Income Countries (LMICs). Of course, if it were not because of the aid from the United Nations in acquiring the cure, that would also remain a dark issue.

This work aims to be a factor in the ‘End TB’ strategy by first focusing on Malawi, Africa, an accurate representation of LMICs. Although a notably friendly country, it can be said that it has not yet reached the industrial revolution in many ways. Due to this state, most of the work is directed towards physical labor and away from the medical field, which compounds the human resource issue due to the large number of people who need to be treated and screened. This work intends to ease the resulting time issue that arises from a lack of human resources with automated technology.

This work describes a Python based image processing algorithm that will be used in an automated microscope that can screen for tuberculosis by looking at coughed up material from the lungs and air passages, otherwise known as sputum or phlegm. Contrary to popular belief, as stated earlier, the main bottleneck in screening for TB and providing immediate and effective care is actually not the cost of the cure. Rather, it is the lack of the necessary human resources needed to screen a country such as Malawi.
1.1 History of Tuberculosis

Tuberculosis is a forgotten plague. TB has been called ‘the Captain of Death’, ‘The Wasting Plague’, and ‘Consumption’. TB has existed for millennia, and, according to written accounts, ‘had killed one in seven people that had ever lived by the dawn of the 19th century’ [4]. The influence of the tuberculosis infection has been woven into the fabric of society, especially North American. Furthermore, TB was a major cause in the settlement of western territories, played into the widespread development of city parks and porches, triggered the first mass public health campaign, influenced fashion and furniture design (rocking chairs and reclining chairs), crushed the publicly accepted notion of spitting on the sidewalk, created a new trend for shaved faces, sparked the ice cream cone, and reformed mindsets that a variety of major diseases were caused by airborne bacteria as opposed to being hereditary, which was the common consensus at the time [4].

Dr. Robert Koch is the German man who discovered that tuberculosis was caused by bacteria that transmits through the air via testing with guinea pigs in 1882 [5]. He was noted to have been influenced by his mentor and teacher who held a distinct belief against that fact that many diseases were hereditary [4]. However, his discovery was not spread to the United States of America until a man named Edward Trudeau became interested in his work.

Edward Trudeau, the founder of the first sanatorium, places of rest and healing for those infected with tuberculosis, was an American physician, but during these times, many physicians did not have proper training with microscopes and other tools that were necessary to culture bacteria [4]. After he took his wife and children to the mountains in an effort to placate his disease with fresh air, he found out about Dr. Robert Koch who had recently found the tuberculosis bacteria and described it in a journal article at the time [4]. From there, Edward Trudeau, who had a curious mind, decided to take a sample from his own lungs and see whether Dr. Robert Koch’s article had any merit [4]. After setting up his own rudimentary experiments by controlling temperature through candle flames and by opening and closing doors, he became the first person in America to prove the existence of the tuberculosis bacteria [4].

Almost 60 years after Dr. Robert Koch discovered Mycobacterium tuberculosis (MTB), in 1943, a graduate student named Albert Schatz discovered Streptomycin, the wonder drug that cured many tuberculosis strains at the time [4]. Unfortunately, he wasn’t given the recognition he deserved, and the Nobel Prize was completely given to his supervisor, Selman Waksman, instead.
Furthermore, this drug was not the end. Many strains of tuberculosis continue to develop, to this day, that require more varied combinations of antibiotic.

1.2 Problem Statement

The focus of this project is on developing a computer vision algorithm that can accurately determine the level of infectiousness, with regards to TB, via direct sputum smears, material from the air passages that comes from deep coughing. This application of image processing is feasible due to acid-fast bacilli (AFB) showing specific shape characteristics in sputum smears of people with active tuberculosis, the only form that is actually infectious. Note that TB is in the Mycobacterium family of bacteria, all of which are AFB. This work consists of an algorithm that is designed in Python to be able to work in tandem with an automated microscope to diagnose and determine the level of infectiousness based on a field of view, a subsection of the sputum smear, in fluorescence images of AFB stained sputum smears. Over the course of the past year, most first world health care settings have moved from bright field microscopy to fluorescence microscopy for sputum smear analysis as an alternative that is much more accurate both for humans and for machines. This project was undertaken in order to provide value to low resource environments in the World Health Organization’s (WHO) endeavor to effectively eradicate tuberculosis as an infectious disease.

Furthermore, the focus is on these low resource environments because of a dialogue between the PMDI (Pediatric Medical Device Institute) group at Virginia Tech and the Ministry of Health of Malawi in Africa revealing that there is a very noticeable lack of the human resources necessary to test the disproportionate number of people who are actually infected with TB in these areas. The underlying root cause is money, which breeds a human resource issue, which then results in a time issue. In the future, the software described will be used in an automated microscope to help solve this time issue.

1.3 Thesis Structure

This thesis is arranged with a literature review following directly after this introduction. First, microscopy is discussed, where the main two types of microscopes, bright field and fluorescent, as well as their uses are explained. General TB diagnostic paradigms are also discussed as well as the current WHO methods of diagnosis, which will be used in the final analysis of this algorithm.
The end of this literature review consists of an analysis of previous algorithms and comparisons to the algorithm described in this work via flowcharts.

The next section is the algorithm section, which is dedicated to explaining the complete algorithm implementation from the ground up. A sectioned flow chart is used for clarity during the explanation, and output images are used to exemplify each process in the algorithm for further understanding. A final example with circled TB bacilli is shown at the end as well.

After the algorithm section, the methods section discusses the methods of analysis, which include precision, recall, and an F1-score, which is the harmonic mean of precision and recall, for both TB bacilli level analysis and WHO field level analysis. The WHO field level analysis uses the same chart as from the Literature Review section which divides up the infectiousness level based on a count of bacilli in a field.

Finally, in the conclusion section, an analysis on the overall feasibility of the algorithm is given in order to shed light on the expectation of this algorithm to provide information regarding the level of infectiousness.

After the conclusion section is the appendix, which contains the import statements used for this algorithm, the underlying required packages, and a link to GitHub for the code. The first half of the name, Git, refers to the open source version control system that was started by Linus Torvalds. It is being used to store code for future use and to facilitate future development for use on the automated microscope when the device is taken to Malawi, Africa to capture samples. Following immediately after the Python code libraries and source code link are output example images containing computer circled bacilli alongside the original images for a visual perspective on the accuracy of the algorithm, sometimes known as visual semantic accuracy.
2. LITERATURE REVIEW

This literature review details the various concepts that were essential to completing the project starting with a review of microscopy, including techniques and hardware. This review then mathematically analyzes the number of pixels that make up a tuberculosis bacillus before moving on to the diagnosis of tuberculosis in high and low resource environments. Finally, this review discusses other potential approaches that have been used in the past before moving on to section three, the algorithm section.

2.1 Microscopy

Microscopy is a ubiquitous term in the medical industry. Specifically, there are two main forms of microscopy, namely Bright Field and Fluorescent. In the following sections, important points about microscopy are explained for better understanding and clarification. For future reference about the key components of a microscope, a diagram of a generalized microscope is shown in Figure 2.1.

![Generalized Microscope Diagram](image-url)

Figure 2.1: Generalized Microscope [6]
2.1.1. Field of View

The field of view (FOV) is the image through a microscope eyepiece of a section on the referenced plane, often a slide sample, seen at any given time. The image below in Figure 2.2 is of a reticle through a circular microscope eyepiece and is used to determine the size of objects when looking through a microscope because the distance between notches in the reticle are known. The equation to find the size of an unknown object is the length of the FOV divided by the length of the object as referenced by the notches in the reticle, commonly referred to as the fit.

\[
\text{size} = \frac{\text{FOV}}{\text{fit}}
\]

Figure 2.2: Field of View - Reticle

The average number of TB cases in North America can be counted on two hands, which means actual sputum samples are hard to come by. That compounded with the rapid decaying effect on fluorescent dye means that the TB that is used for this work would need to be alive and stained by Fluorescent dye within 24 hours of taking the image. The reticle method of determining size is imprecise, dangerous, and difficult to use due to the need for live TB as previously noted. Instead, the size is calculated mathematically. The average size of tuberculosis is 0.2 - 0.5 \( \mu m \) in width and 2-4 \( \mu m \) in length [7]. Now, given the size of the image, 5 Megapixels (MP), 1944x2592, as well as the FOV, 490x560 \( \mu m \), the calculation for the number of pixels that a bacillus takes up is simple. On one hand, there will be distortions from fluorescent halos, so it seems like we should calculate based on square \( \mu m \). Using this mindset, the square number of pixels is 2244, and the square \( \mu m \) FOV is 523. Now, dividing 2244 by 523 gives us about .43 \( \mu m \) per pixel. From this, it is shown that a bacillus, on average, ranges from around 1-2 pixels in width and around 5 to 9 pixels in length. Of course, this is the average, and the maximum possible size is more important.
so that none are missed. Thus, 2592 is used for the pixel count, and 490 μm is used for the actual size. Dividing 2592 by 490 μm, the outcome becomes 5 pixels per μm. This number means that the length of a bacillus could range all the way up to 20 pixels. Assuming distortion and fluorescent glow, an acceptable pixel-range for a TB bacillus that accounts for this variation is a 24x24 pixel square.

2.1.2. Fluorescence Microscopy

Although there are various types of fluorescence microscopes, the fluorescent microscope Chang et al. has used to collect the images for the dataset is called the Cell Scope, which has an epifluorescence design as shown in Figure 2.3 [8, 9].

![Diagram of Epifluorescence Microscope](image)

**Figure 2.3: Epifluorescence Microscope [10]**

A light source is focused through an excitation filter, which is normally a band pass optical filter that deflects light outside of a specified bandwidth. This light is then reflected onto a slide sample by a dichroic filter that contains a specimen that has been dyed with a dye specific to the bandwidth of the light being directed at it. Furthermore, the dye must be able to interact with the bacteria being analyzed. In the case of bacteria from the Mycobacterium family, the usual dye is called
Auramine O. Once the light of a specific wavelength hits the sample, anything that has absorbed the dye will fluoresce. The fluorescence is of a certain wavelength of light that is then directed through an emission filter, which is another band pass filter, that removes all light wavelengths except the specified one in order to clearly see the fluorescing sample. A visual graph showing the effects of these filters on the optical spectrum is shown in Figure 2.4.

2.2 Diagnosis of Tuberculosis

![Optics Filter](image)

**Figure 2.4: Optics Filter [11]**

2.2.1. General Approaches

There are two states that TB can be in, similar to many infectious bacteria. It can be either latent or active. Latent TB is not contagious, but these people have a 10\% risk of developing an active TB infection in their immediate future, specifically within the first two years of infection for the vast majority of people [12]. TB symptoms are mild in the beginning. This gradual creep of sensations causes many people to think that something else is wrong, leading to a delay in medical care. Even more unfortunate is that this delay increases the transmission of the bacteria to other people [13].

There are a variety of tests used to detect the TB bacteria. According to the CDC, TB skin tests (TST) and TB Blood Tests can tell whether a person has been infected with the TB bacteria but cannot differentiate between latent TB infection (LTBI) and active TB disease. This differentiation requires more thorough testing through x-rays or sputum samples, a mixture of
saliva and mucus coughed up from the respiratory tract sometimes called phlegm, which is analyzed through a microscope [14]. This thesis analyzes sputum sample images because the goal of this research project is to aid developing countries in testing for active TB through affordable technologies. Sputum sample testing is a non-specific test for acid-fast bacilli, and Mycobacterium tuberculosis fits within that category [15]. The affordable quality coupled with the high-throughput capable analysis makes this method a prime screening test that, when coupled with the right symptoms, can quickly prompt physicians to perform an actual test for TB or simply give the patient the medicine, just to be safe, if the appropriate materials are not on hand.

2.2.2. Reporting Level of Infectiousness

In most high income countries, TB is screened via fluorescence staining. However, in most LMICs, TB is screened via Ziehl-Neelsen staining, the bright field microscope staining dye, which is empirically less accurate than the fluorescent alternative due to an increase in false positives according to the World Health Organization (WHO). The more common use of light field microscopes is due to the high costs of fluorescence microscopes. However, due to recent advancements in the realm of optics, microcontrollers, and LEDs (light-emitting diode), the cost to develop an automated microscope has come down considerably. In the future, this algorithm will be used by a team of student engineers, doctors, and biologists within the PMDI group that are developing an affordable automated light and fluorescence microscope.

Reporting the level of infectiousness in patients is done by manually counting the number of TB bacilli in a number of fields, collecting the data, and fitting the sputum smear to a certain level of infectiousness based on an average level of infectiousness across a number of fields. Specifically, reporting for fluorescence is done via the chart in Table 2.1.
In reference to the above table, a length refers to a straight line, horizontally across the length of the smear, which can mean any number between 10 and 100 fields depending on lab specific protocol. An image depicting a length is shown in Figure 2.5.

![Figure 2.5: Figure depicting a ‘length’ of a sputum smear](image)

<table>
<thead>
<tr>
<th>What you see (200x)</th>
<th>What you see (400x)</th>
<th>What to report</th>
</tr>
</thead>
<tbody>
<tr>
<td>No AFB in one length</td>
<td>No AFB in one length</td>
<td>No AFB observed</td>
</tr>
<tr>
<td>1-4 AFB in one length</td>
<td>1-2 AFB in one length</td>
<td>Confirmation required*</td>
</tr>
<tr>
<td>5-49 AFB in one length</td>
<td>3-24 AFB in one length</td>
<td>Scanty</td>
</tr>
<tr>
<td>3-24 AFB in one field</td>
<td>1-6 AFB in one field</td>
<td>1+</td>
</tr>
<tr>
<td>25-250 AFB in one field</td>
<td>7-60 AFB in one field</td>
<td>2+</td>
</tr>
<tr>
<td>&gt;250 AFB in one field</td>
<td>&gt;60 AFB in one field</td>
<td>3+</td>
</tr>
</tbody>
</table>

Table 2.1: Level of Infectiousness for Fluorescence Sputum Smears with AFB [16]

2.3 Image Processing approaches

Chang et al. have described several algorithms previously employed in both Bright Field and Fluorescence microscopy for the analysis of sputum samples via image processing. This section consists of the general approaches from these algorithms as well as a few others, beginning with the algorithms tested on fluorescent images before moving on to the algorithms tested on bright field images. Each algorithm is shown in a flowchart and compared to the generalized flowchart of this work’s rule based algorithm that is shown in Figure 2.6. In all future flow charts, blue boxes are used as fillers and do not represent any process.
2.3.1. Fluorescence

Veropoulos’ Algorithm [17]

A flowchart comparison of this work’s general algorithm and Veropoulos et al.’s algorithm is shown in Figure 2.7.
Both algorithms begin by converting an image format such as TIFF into a matrix of numbers that can be analyzed. The following steps are considered pre-processing because they are done in order to filter out noise before the actual classification steps. In order to reduce the image to sections that have the potential to be bacilli, the rule based algorithm described in this thesis greyscale color conversion that converts the matrix from a 3D (3-Dimensional) representation to a 2D (2-Dimensional) representation followed by an edge preserving smoothing filter. Then, a filter that looks for vessels of a certain width is used just before a binary threshold that removes ambiguity in an eight bit 2D matrix by forcing each number to be either zero or 255. Note that vessels are bright elongated shapes in the image. At this point, every pixel, or location in the 2D matrix, that has a value of 255 has a high potential to correspond to a part of a TB bacillus. Finally,
the remaining pixels are grouped into potential bacilli by finding all external contours in the binary image. Veropoulos et al.’s algorithm begins the pre-processing step by using a common edge detector known as Canny, which has grayscale conversion built in. Next, the algorithm consists of a binary threshold. Then, using 8-connectivity, which implies a consideration of each of the surrounding 8 pixels, objects that could not possibly be TB bacilli are removed by thresholding for size based on the total number of pixels corresponding to an object. In the resulting edge image, objects that have holes are filled via morphological operators such as erosion and dilation in what is noted within Veropoulos et al.’s paper as pixel linking. Immediately following is a boundary tracing algorithm similar to external contour detection. An image of TB after edge detection, size filtering, and edge pixel linking, is shown in Figure 2.8 beside the original image.

![Original TB image](image1.png) ![Processed image](image2.png)

Figure 2.8: (Left) Original TB image (Right) processed image [17]

At this point, both algorithms have external boundaries, or contours, that are potential bacilli; however, Veropoulos et al. uses 16 different Fourier descriptors and a neural network based machine learning algorithm. The algorithm this thesis describes can simply threshold for size based on the ratio between height and width of the contour alongside some background subtraction techniques to confirm that the object is in the foreground and therefore an actual bacillus.
Forero’s Algorithm [18]

A flowchart comparison of this work’s general algorithm and Forero et al.’s algorithm is shown in Figure 2.9.

Forero’s algorithm had to deal with slides of tuberculosis that were only stained by Auramine O and not counterstained, which would remove most if not all of the background fluorescing colors, exemplified by the image in Figure 2.10.
The images seem to be overstained. Thus, the images that were analyzed by Forero contain a larger bandwidth of light, corresponding to multiple colors. Due to this, during the pre-processing phase, the first step was to threshold based on color by extracting the green channel from the RGB image due to empirical analysis in seeing that the bacilli were a green color as seen below in Figure 2.11.

After thresholding for a specific color range, the next step, similar to Veropoulos et al.’s algorithm, was edge detection, pixel linking, and filling in closed contours via morphological operators such as dilation and erosion. Next, information about shape via Hu moments were extracted after discarding other shape descriptors such as area, compactness, major and minor axis lengths,
eccentricity, perimeter, solidity, and Fourier descriptors due to underperformance according to Forero. At this point the Hu moments were clustered via a k-means algorithm due to the need for multiple Hu moments to describe the shape of tuberculosis bacilli. Finally, Forero implemented a Gaussian Mixture Model (GMM) to parametrize a 4-dimensional feature space for use with a Bayesian classifier. The 4-dimensional feature space consisted of four of the seven Hu moments that describe the shape of an object.

Chang’s Algorithm [9]

A flowchart comparison of this work’s general algorithm and Chang et al.’s algorithm is shown in Figure 2.12.

Figure 2.12 (Left) Generalized Flowchart (Right) Chang’s Flowchart
Chang et al. use an interesting preprocessing method called a Top Hat transformation. A before and after image is shown in Figure 2.13.

Chang et al.’s preprocessing method begins with a Top Hat transformation, which is a higher level combination of erosion, dilation, and background subtraction. The Top Hat transformation is more prone to noise than the vessel filter described in this thesis because the vessel filter specifically sets pixel values equal to zero and is therefore less noisy when converting to binary. In fact, the Top Hat transformation would have been used if not for the need for highly accurate shape information and a lower noise level to avoid the use of a larger number of feature descriptors. Chang et al. then use a connected component algorithm to find each remaining object in the binary image and extract a 24x24 pixel square around the center of the object, due to the size of an average TB bacteria, before continuing to the classification stage.

In the classification stage, the algorithm represents TB candidate objects using Hu moments, geometric and photometric features, as well as a histogram of oriented gradients (HOG), where the HOG proved to not be very useful as explained in the second half of Chang et al.’s paper. The geometric features were very diverse, including area, convex area, eccentricity, equivalent diameter, extent, filled area, major/minor axis length, max/min/mean intensity, perimeter, solidity, and Euler number. Note that Euler number is defined as the total number of
objects minus the number of holes in the objects. Finally, Chang et al. use and compare 3 different machine learning approaches, including logistic regression, linear support vector machines (LSVM), and nonlinear support vector machines to count identified bacilli.

2.3.2. Bright Field

Costa’s Algorithm [19]

A flowchart comparison of this work’s general algorithm and Costa et al.’s algorithm is shown in Figure 2.14.

Figure 2.14: (Left) Generalized Flowchart (Right) Costa’s Flowchart
Bright field algorithms are much more dependent on color due to the increase in background noise. The preprocessing section of Costa’s algorithm looks at color information in terms of a contrast maximized image by subtracting the G channel from the R channel, in an RGB sense, after empirically analyzing the hue histogram of the images. After segmentation using a 10-bin histogram of this R-G image via a simple threshold, the image is binarized, and an 8-connected component algorithm is used to find potential objects in the binary image.

From this point, the algorithm thresholds based on size using morphological filters in an effort to reduce false positive regions by removing objects with too many or too few pixels to be considered TB bacilli. Alternatively, this work’s algorithm uses a rotated bounding box to get more accurate shape information such as width and height because the number of pixels that make up an object can be deceiving in terms of size. For example, a circle could have the same number of pixels as a really long but thin rod. Images of circled bacilli is shown in Figure 2.15 in two separate example output images.

Figure 2.15: Circled bacilli of Costa’s final computer vision analysis [19]
Khutlang’s Algorithm [20]

A flowchart comparison of this work’s general algorithm and Khutlang et al.’s algorithm is shown in Figure 2.16.

![Flowchart](image)

Figure 2.16 (Left) Generalized Flowchart (Right) Khutlang’s Flowchart

The pre-processing section of this algorithm begins with a color threshold by looking at the Gaussian distribution of manually labeled masks of TB bacilli from the dataset and setting limits based on the center of the Gaussian distribution of color. After converting the image to binary, a connected component algorithm is used to identify potential bacilli.

In the classification stage, as in many of the other algorithms, a size threshold is used to remove objects that have a pixel number outside of the potential range of a bacilli. Finally, from various descriptors such as Fourier descriptors, generalized RGB moments, eccentricity, and pixel...
color values are analyzed for each object in the feature extraction stage. These features are then compared with known bacilli features in several supervised learning algorithms such as Mixture of Gaussian (MoG) and k-Nearest Neighbor (kNN) before using the output to count identified bacilli. The rule based algorithm this thesis describes drew inspiration from the way the color descriptor in Khutlang et al.’s algorithm was created for use in the foreground confirmation stage. This inspiration was with respect to taking a small subset of the pixels comprising a bacillus such as a sample of pixels from the center of a potential bacillus or the perimeter.

Sadaphal’s Algorithm [21]

A flowchart comparison of this work’s general algorithm and Sadaphal et al.’s algorithm is shown in Figure 2.17.

Figure 2.17 (Left) Generalized Flowchart (Right) Sadaphal’s Flowchart
In the pre-processing stage, this algorithm uses a Bayesian pixel classifier using prior masks of bacilli to identify potential bacilli locations similar to Khutlang et al. Then, an 8-connected component algorithm is employed after dilating the image to connect nearby disconnected pixels into a single object.

In the classification stage, a multi-stage size threshold analysis is done using rotation invariant shape descriptors, including axis ratio, eccentricity, and number of pixels. An example of the computer annotated identification is shown in Figure 2.18.

![Figure 2.18: (Left) Original image (Right) Computer labeled image [21]](image)

Note the red label on the computer labeled image that was rejected due to an eccentricity outside of the threshold specified and the green labels that signify correct computer annotations.

**Rachna’s Algorithm [22]**

A flowchart comparison of this work’s general algorithm and Rachna et al.’s algorithm is shown in Figure 2.19.
The pre-processing stage for Rachna et al.’s algorithm begins similar to the algorithm for this work, with a filter. Specifically, a median filter is used to remove noise, while this work uses a bilateral filter. The reasoning is that this work’s algorithm is more dependent on edge information and contrast, while Rachna et al.’s algorithm is heavily color based. In Rachna et al.’s algorithm, a global threshold is used to remove a large amount of the background, and a k-means clustering algorithm is utilized to collect the remaining colors into a predefined number of clusters in an adaptive thresholding sense. An alternative approach to the k-means algorithm is also used, called Otsu thresholding, which is simply another form of adaptive color thresholding. Immediately after, locally similar pixels are connected in a method Rachna et al. dub ‘Region Growing’, and the result is binarized. Finally, a connected component algorithm is used to identify the potential bacilli from the pre-processing stage.
In the classification stage, Rachna et al. use a pixel based size threshold to ensure that objects too large or too small are removed. Examples of color thresholding and region Growing are shown in Figure 2.20.

![Figure 2.20: (a) K-means clustering (b) Region growing after K-means clustering (c) Otsu thresholding (d) Region growing after Otsu thresholding [22]](image)

**2.4 Background Overview**

Although fluorescent microscopes are generally more expensive, they are also more accurate. Furthermore, several key technologies such as microcontroller, LED, and optics, including cameras, technology have had a significant reduction in price in the past few years, which makes the development of an affordable automated fluorescence microscope feasible. As such, the rule based image processing algorithm this thesis describes is written for fluorescent images using a dataset that Chang et al. has made public [9].

As seen from the review of previous methods, most of the computer vision methods for fluorescent images are machine learning based with a large number of features. Furthermore, none of the previous methods take size into account via the length and width of a rotated bounding box, which will be explained in the algorithm section. Instead, the other works use the area in number...
of pixels and other methods of taking size into account such as Hu moments, eccentricity, etc. The algorithm this thesis describes focuses on taking a more direct approach that is more intuitive by taking into account the width and height of the bacilli, which allows for a more specific size based threshold.
3. ALGORITHM

The algorithm section details this work’s associated algorithm by beginning with an understanding of the Red-Green-Blue (RGB) color space and how it is converted to Grayscale as well as their respective uses in the algorithm. Following this introduction to color spaces is a section that explains how convolution is used to quantitatively analyze differences between neighboring pixel values because of its importance to the next two sections. Then, a section on bilateral filtering, which is actually used to preprocess and smooth the image to remove noise before passing it through the vessel filter. Next, an explanation of vessel filtering is given along with how it works in the algorithm. The final two sections consist of an explanation of how size is estimated and about how background subtraction plays a role in removing false positives in the foreground confirmation step.

3.1 Color Spaces

Before anything happens, the image must be read into the algorithm and decomposed into an 8-bit 3D matrix. The first four steps to this algorithm are shown in Figure 3.1, where the decomposition to an 8-bit 3D matrix and the Grayscale conversion to a 2D matrix steps are exemplified by highlighting them blue in a snippet taken from the algorithm flowchart.

![Figure 3.1: First four sections of the algorithm flowchart](image)
An image file such as a JPEG, TIFF, or PNG is compressed and not readily readable by humans. Only when it is decompressed into a color space such as RGB, which is the most common space, can it be analyzed via computer vision algorithms. When creating datasets of images, it is important to use a lossless file format such as TIFF or PNG as opposed to a lossy format such as JPEG. The reasoning is that each time an image is read, there is some loss of information for JPEG, which is hugely detrimental to image processing. TIFF is the preferred choice for professionals in the photography industry and for people who work with computer vision when hard drive space is not an issue. Moving on, RGB space consists of an 8bit three dimensional matrix with three numbers ranging from 0-255 for each pixel as shown by Figure 3.2.

Another often used color space in image processing is called Grayscale, which, as its name implies is a monochrome depiction of the image that only carries intensity information. Grayscale images are represented as 2D matrices. An important point to mention is that an image can be considered an RGB image even if it looks Grayscale. Before saving the images, the people who created this dataset, Chang et al, converted the images to Grayscale [9]. This color space is the required intermediary step before converting an image to binary via an intensity threshold. As an example, if numbers are above that threshold, they will be changed to 255. Likewise, numbers that are below that threshold will be changed to 0. The RGB and Grayscale representations of a sputum smear with TB bacilli are shown in Figure 3.3, where the RGB image, which originally consisted mainly of green and black colors was converted to grayscale before being saved as a TIFF file by Chang et al.
Demystifying the differences between neighboring pixels and quantitatively analyzing them in reference to known values is the role of computer vision. Convolution is one widely used method in image processing applications because of this capability, which allows the computer to identify edges, match shapes, remove noise, etc. [24]. In particular, for this application, the algorithm uses it to find the concavity information in a TB smear image by using second derivative Gaussian kernels as well as the eigenvalue decomposition of the corresponding 2x2 Hessian matrix for a square vicinity around a pixel to find the shape of a tuberculosis bacillus. The Hessian matrix is discrete because an image is represented as a matrix of scaled delta functions, an impulse response, due to hardware limitations in distinguishing light via sensors and the mathematical practicality therein. In other words, an image is a matrix of numbers, as stated in the previous section. In digital signal processing, convolution acts on two 1-Dimensional (1D) signals. Moreover, in image processing, convolution acts on two images or, rather, one image and a filter, often called a kernel [24].

To best understand 2D convolution, a key component of this work, this section will first explain 1D convolution via the corresponding formula shown in equation 1, where f is the signal, h is the filter, and the integral becomes a summation when dealing with a discrete problem such as a sensor signal or image, which limits infinity to the size, or amplitude, of the signal. Finally, g is the result of the convolution of f with h.
\[
g(i) = f(i) \otimes h(i) = \int_{x=-\infty}^{\infty} f(x) h(i-x) \, dx
\] (1)

Assume a three-point signal \( f = [3 \ 4 \ 5] \) with positions 0 to 2 and a two-point binary filter \([2 \ 1]\) with positions 0 to 1. These signals are represented in Figure 3.4, where \( A \) is the magnitude and \( i \) is the signal location.

From equation 1, it is shown that each location in the signal \( f \) will be multiplied by a flipped filter \( h \), due to the negative sign, that is shifted by a number \( i \), where \( i \) encompasses the entire length of the signal. In Figure 3.5, the calculation for convolving \( f \) with \( h \) is shown. The border is padded with a zero at the end in order to completely multiply the filter with the signal.

\[
\begin{align*}
g(0) &= (3 \times 1) + (4 \times 2) = 11 \\
g(1) &= (4 \times 1) + (5 \times 2) = 14 \\
g(2) &= (5 \times 1) + (0 \times 1) = 5
\end{align*}
\]
Thus, the output becomes \( g = [11 14 5] \). From this, it is shown that convolution is simply a sliding multiplication problem. In the same way, a kernel will slide over the image for 2D convolution. The formula for a 2D convolution is shown by equation 2, where \( f \) is the image, \( h \) is the kernel or filter, and \( g \) is the resulting output, which is the convolution of \( f \) with \( h \), where \( h(i, j) \) is the center of the kernel, and \( f(x, y) \) is a pixel in the image.

\[
g(x,y) = f \otimes h = \int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} f(x,y) h(i-x,j-y) \, dx \, dy
\]  

(2)

Again, reiterating the importance of convolution, it is highly useful for image processing because it is similar to how the optic nerve processes information. Optic nerve cells will change their activation state based off of their neighbors. For example, through a process known as lateral inhibition, based on the intensity of the incoming photons, a nerve cell will decrease the neighboring nerve cells’ activations states [24]. To understand lateral inhibition, look at Figure 3.6.

![Figure 3.6: Lateral inhibition example [25]](image)

The squares A and B are actually the same color as seen by the cropped out squares on the right. This increase in perceived contrast is due to the more active cells diminishing the activation states of neighboring cells because the A square is surrounded by white squares and because the B square is brighter than the surrounding dark grey squares. Cropping the squares out and putting them on a white background shows that they are in fact the same color. This inhibition process happens
with other senses in the body as well. It is similar to how a gunshot wound would diminish the pain of a paper cut.

In order to further clarify how a filter works in convolution with an image, a mean filter will be used on a Snellen eye chart. Again, a filter and a kernel are interchangeable terms within the field of computer vision. The mean filter should do just as its name implies, take the mean of an image with respect to the size of the kernel applied, where the effect is that of blurring the image. The kernel is almost always a square matrix, and the one often used for a mean filter is a 3x3 matrix filled with ones. When convolving such a filter with an image, the center of the kernel is moved across each pixel in an image, summing up the multiplication of the kernel with the overlapping region in the image and replacing the value in the matrix by that value. An example is given in Figure 3.7. The kernel of ones is convolved with a 3x3 matrix, which represents the image.

\[
g = f \otimes h = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}
\]

Figure 3.7: Mean filter convolution with an image

Figure 3.8 works through the problem given in Figure 3.7, where a 3x3 grid is used to represent the kernel of ones. When convolving an image with a kernel, the center of the kernel is positioned on top of each pixel in the image. Furthermore, the borders of the image are padded with zeros similar to the 1D representation earlier in order to avoid null values when centering the kernel around a pixel at the border. The kernel has a one in every location, so flipping the kernel, as called for by convolution, does not change it. Also, a kernel in image processing is generally symmetrical, so it should be noted that flipping a kernel yields no qualitative change when the image is normalized in the end. Again, the 3x3 grid is the symbolic kernel of ones with its center overlaid on each pixel in the border padded image. The top left corner of the image f as well as in the output g will be called (0,0), where the first 0 is the y direction and the second is the x direction.
\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

\[g(0,0) = (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (1 \times 1) + (1 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) = 2\]

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

\[g(0,1) = (0 \times 1) + (0 \times 1) + (0 \times 1) + (1 \times 1) + (1 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) = 2\]

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

\[g(0,2) = (0 \times 1) + (0 \times 1) + (0 \times 1) + (1 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) = 1\]

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

\[g(1,0) = (0 \times 1) + (1 \times 1) + (1 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) = 2\]

\[
\begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
= \begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

\[g(1,1) = (1 \times 1) + (1 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) = 2\]
\[ g(1,2) = (1 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) = 1 \]

\[ g(2,0) = (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) = 0 \]

\[ g(2,1) = (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) = 0 \]

\[ g(2,2) = (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1) = 0 \]

\[ g = \begin{pmatrix} 2 & 2 & 1 \\ 2 & 2 & 1 \\ 0 & 0 & 0 \end{pmatrix} \]

Figure 3.8: Mean filter convolution with an image f
Performing the calculation for each location in the image will give the output $g$ as shown in Figure 3.8; however, an image has values within an 8-bit range, from 0 to 255. If those two ones in the original test image $f$ were instead 255, the output would be instead as shown in Figure 3.9. The left side gives the exact solution, and the right side gives the 0 to 255 normalized solution.

$$g = \begin{pmatrix} 510 & 510 & 255 \\ 510 & 510 & 255 \\ 0 & 0 & 0 \end{pmatrix} \quad g = \begin{pmatrix} 255 & 255 & 127 \\ 255 & 255 & 127 \\ 0 & 0 & 0 \end{pmatrix}$$

Figure 3.9: (Left) 8-bit output (Right) (0-255) normalized output

Finally, to give an idea of what convolving an actual image with a kernel looks like, the result of convolving the Snellen eye chart with the same 3x3 kernel of ones is shown in Figure 3.10.

![Figure 3.10: (Left) original image (Right) (3x3) mean filtered](image)

Not much of a change can be seen because the filter is too small. Now, a larger kernel will be used to give an intuition of what happens in this case. Figure 3.11 shows the same convolution with a kernel size of 9x9, having 81 ones in the kernel, instead of 3x3, having 9 ones in the kernel.
Now, the blurring effect can be seen because more pixels, or numbers, are taken into account. Note that a normalization step must be done to keep the value of the pixel from going over 255 to be viewable in a normal image format. The next section is the bilateral filter section, which also uses convolution to achieving a blurring or smoothing effect.

3.3 Bilateral Filter

The mean filter is a simple and useful example, but, in the process of smoothing the image out, it also blurs the edges in the image, thereby blurring the bacilli. An edge preserving filter such as a bilateral filter is necessary for this algorithm because the next step, the vessel filter, relies on gradient information based on those edges to find vessels in the image. This bilateral filtering step in the algorithm is shown in Figure 3.12.

Figure 3.11: (Left) original image (Right) (9x9) mean filtered image

![Figure 3.11: (Left) original image (Right) (9x9) mean filtered image](image)

Figure 3.12: First four sections of the algorithm flowchart (Bilateral Filter)
The bilateral filter has been traced back to the work of Aurich and Weule on nonlinear Gaussian filters in 1995 [27]. In truth, the bilateral filter works very similarly to the mean filter. It takes the mean of the surrounding pixels, determined by the size of the kernel, and replaces that value of the pixel in the image that the filter is centered around with that mean. The difference is that it also takes into account the variation of intensities within a certain distance when doing the mean. In short, it replaces the pixel value that the kernel is centered around with an average of nearby and similar valued pixels.

The equation for a bilateral filter BF for a 2D Image I at point p is defined below [26]:

\[
BF[I]_p = \frac{1}{N} \sum_{q \in S} G_{\sigma_s}(\|q - p\|)G_{\sigma_r}(\|I_p - I_q\|)I_q
\] (3)

\(G_{\sigma_s}\) is a spatial Gaussian where \(p\) is the pixel in consideration and \(q\) is a neighboring pixel. N is the normalization term that keeps the values of the image within an allowable range, i.e. 0-255 for a normal image. For the 2D case, the Gaussian function is defined by the bivariate normal distribution in equation 4 and as seen in Figure 3.13 with an 11 x 11 kernel size and a \(\sigma_s\) equal to 17.

\[
G_{\sigma_s}(\|p - q\|) = G_{xy} = \frac{1}{2\pi\sigma_s^2} e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma_s^2}}
\] (4)

Figure 3.13: Gaussian kernel
For the discrete case, in equation 5, the variables x and y refer to the coordinate location of the value in the kernel, assuming the center of this kernel is location (0,0). Furthermore, the coordinate (x0, y0) is the center location (0,0). Sigma is a threshold specified by the user that changes the spread of the Gaussian function.

\[ G_{\sigma_r}(||I_p - I_q||)_q = 1 - e^{-\frac{(||I_p - I_q||)^2}{2\sigma_r^2}} \]  

As an example of the bilateral filter, consider an image that has a sharp boundary as in Figure 3.14.

Figure 3.14: Sharp boundary image

If the bilateral filter, which is the Gaussian filter scaled by the values in the range filter, is centered on a pixel in the middle of the image on the first white pixel from the left, then an 11x11 range filter with a \( \sigma_r \) equal to 17 will be as seen in Figure 3.15.
Then the bilateral filter is the multiplication of the Gaussian range kernel and the Gaussian space kernel, resulting in Figure 3.16, which, in this case, essentially zeros the left 5 columns of the 11 x 11 Gaussian space kernel.
Many open source image processing modules such as OpenCV have this filter functionality readily available. The parameter specifications include the size of the kernel used, $d$, the value for the standard deviation of space, $\sigma_s$, and the value for the standard deviation of range/color, $\sigma_r$. Examples of how the bilateral filter works on an image are shown in Figure 3.17.

The first parameter change allows for easy visualization of the filter working. It becomes much more difficult to see the second set of parameters working because they are tuned for images of TB. The texture for the cat image is too strong, which is why the values are much higher in the first set of parameters. Vessel filtering is explained in the next section, and Figure 3.18 shows the output of the vessel filter with and without bilateral filtering when converted to binary so that the importance of the bilateral filter can be seen.

Figure 3.17: (a) Original image (b) parameter change 1 (c) parameter change 2

\[ d = 25 \]
\[ \sigma_r = 100 \]
\[ \sigma_s = 15 \]

\[ d = 25 \]
\[ \sigma_r = 100 \]
\[ \sigma_s = 15 \]
3.4 Vessel Filtering

The next step in the algorithm is the vessel filtering step. It consists of a convolution of the tuberculosis image with several second partial derivative Gaussian kernels to create an array of Hessian matrices. An eigendecomposition is done for each Hessian to determine whether the square vicinity of pixels associated with each Hessian is a part of a tubular structure. Finally, it employs several equations developed by a man named Frangi that uses the eigenvalues of each Hessian to highlight the vessels in the tuberculosis image. The flow in this vessel filtering section is shown in Figure 3.19.

Figure 3.18: (Left) Vessel Filter output without bilateral filtering  
(Right) Vessel filter output with bilateral filtering

Vessel Filter

- Construct second derivative Gaussian kernels
- Create array of Hessian matrices
- Determine structure from eigenvalues of each Hessian
- Create Vessel image using equations from Frangi

Figure 3.19: Vessel filter flow chart
The vessel filtering step is also highlighted in the overall flowchart in Figure 3.20.

![Algorithm Flowchart](image)

**Figure 3.20: First four sections of the algorithm flowchart (Vessel)**

First, this section will reconstruct the second derivative Gaussian kernels while providing examples along the way. Furthermore, a Prewitt kernel will be used to explain how a first derivative kernel works when convolved with an image because a Prewitt derivative kernel is the simplest version of a derivative kernel and will aid in understanding the first derivative Gaussian kernels. The vessel filter calls for Gaussian derivative kernels instead of Prewitt derivative kernels because they are much more robust, especially in the second derivative, by being less sensitive to noise. The 2D Gaussian function, or bivariate normal distribution, that defines the basic Gaussian kernel is defined by equation 6. Furthermore, Figure 3.21 shows an image of a 7x7 Gaussian kernel.

\[
G_{xy} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}}
\]  

(6)
For the discrete case, in equation 6, the variables $x$ and $y$ refer to the coordinate location of the value in the kernel, assuming the center of this kernel is location $(0,0)$. Furthermore, the coordinate $(x_0, y_0)$ is the center location $(0,0)$. Sigma is a threshold specified by the user. Leaving it as one is suitable for a small Gaussian kernel, otherwise a correction will have to be made for the sigma squared value by using it to normalize the Gaussian kernel after filling in the values into the correct locations. The $e$ simply references Euler’s number. Taking the partial derivative of this function with respect to $x$ gives us the equation for our $x$ derivative kernel as shown in equation 7 and as pictured in Figure 3.22.

\[
\frac{\partial}{\partial x} (G_{xy}) = \frac{-x}{2\pi\sigma^4} e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}}
\]  

(7)

Figure 3.21: Gaussian kernel

Figure 3.22: (Left) $x$ partial Gaussian derivative kernel
(Right) $y$ partial Gaussian derivative kernel
Filling in the coordinate locations according to equation 7 gives the first x derivative kernel, where a 7x7 kernel size was chosen due to the actual size of the second derivative kernel for the vessel filter used being 7x7. Then, transposing the x first derivative kernel gives the first y derivative kernel. Finally, convolving that first y derivative kernel with the TB image gives the following first derivative image output in Figure 3.23, where the brightness has been enhanced by 40%.

As shown, the convolution of these kernels with the image will take the derivative of the image, which is the same as saying that they will find the meaningful differences between pixels in the x and y directions. There are various kernels that will take the derivative of an image. For this application, a Gaussian function that defines what values the kernel will hold is used; however, a simpler derivative kernel called the Prewitt kernel makes it easier to explain how it works. The Prewitt Kernel is the simplest derivative kernel, based on the idea of a contrast difference. Both the x direction gradient and the y direction gradient kernels are shown in Figure 3.24 exemplified by an x and a y.

\[
x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}
\]

Figure 3.24: (Left) x derivative kernel (Right) y derivative kernel
These will give the edge information in the image. The greater the difference between pixels, the more pronounced the edge will seem in the resulting image. First, an output example when convolving an image with the x kernel, which is the x gradient information, will be given in Figure 3.25. This figure also includes the normalized output of convolving an image with the y kernel to show y gradient information.

![Figure 3.25](image)

(a) original image (b) x gradient image (c) y gradient image

Figure 3.25 (a) – original image (b) x gradient image (c) y gradient image

Figure 3.26 shows an image from a noisy TB positive sputum smear to see how the convolution of the y derivative Prewitt kernel works on an image of tuberculosis, where the brightness is enhanced by 40% for clarity.

![Figure 3.26](image)

Figure 3.26: (Left) original TB image (Right) y (3x3) Prewitt derivative output
As shown, the Prewitt Kernel does a pretty good job of detecting edges via gradient information; however, the vessel filter calls for a Gaussian derivative kernel because it is much more robust in the second derivative by being less sensitive to noise. Now, the previous example of a Prewitt kernel was 3x3. Using a first derivative 7x7 Prewitt kernel, the output would be shown as in Figure 3.27. The brightness has been enhanced by 40%.

![Figure 3.27: (Left) original TB image (Right) Y (7x7) Prewitt derivative output](image)

Now, taking the second partial derivative of the original Gaussian function to get the second derivative x kernel gives equation 8.

$$\frac{\partial^2}{\partial x \partial x}(G_{xy}) = \frac{x^2 - \sigma^2}{2\pi\sigma^6} e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}}$$  \hspace{1cm} (8)

Transposing the resulting x kernel gives the second derivative y kernel. Finally, the second partial derivative with respect to both x and y is given below in equation 9.

$$\frac{\partial^2}{\partial x \partial y}(G_{xy}) = \frac{x^2y^2}{2\pi\sigma^6} e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}}$$  \hspace{1cm} (9)

The 7x7 kernels that were calculated for this work and are correlated to equations 8 and 9 are given in Figure 3.28.
Now, the first goal of the vessel filter has been accomplished and the second derivative Gaussian kernels have been constructed. The next step is to create an array of Hessian matrices. For this step, each second derivative kernel must be convolved with the original tuberculosis image in order to create the n discrete Hessian matrices, where n is the number of pixels in the image. Each Hessian matrix is then constructed according to equation 10.

\[
H = \begin{bmatrix}
\Sigma_{i \in V} I(i) \frac{\partial^2 G_{xy}}{\partial x \partial x} (i - i0) & \Sigma_{i \in V} I(i) \frac{\partial^2 G_{xy}}{\partial x \partial y} (i - i0) \\
\Sigma_{i \in V} I(i) \frac{\partial^2 G_{xy}}{\partial x \partial y} (i - i0) & \Sigma_{i \in V} I(i) \frac{\partial^2 G_{xy}}{\partial y \partial y} (i - i0)
\end{bmatrix}
\] (10)
Equation 10 is defined where $G_{xy}$ is the Gaussian kernel centered at i0, V is some square vicinity of the same size of the Gaussian kernel, and I is the original 2D image. For the entire image, I, there is one 2x2 Hessian matrix for a square vicinity around each pixel, exemplified by equation 11, which is of length 2 x 2 x n. Again, n is the number of pixels in the image.

\[ H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \]  \hspace{1cm} (11)

Moving on, a comparison of TB bacilli using the vessel filtering method, which will be further explained later, is shown below in Figure 3.29. The second derivative Prewitt kernels are used for the second image, and the second derivative Gaussian kernels are used for the third image. This comparison is to show a preview of the final output of this vessel filter and to show that the Gaussian filter is much more robust than the Prewitt filter for this application, which is the reason it was chosen.

![Figure 3.29: (a) Original image (b) Prewitt Vessel Filter (c) Gaussian Vessel Filter](image)

The use of the second derivative will now be explained to give a deeper understanding of what it is in the context of image processing as well as what the output of convolving a second derivative kernel with an image looks like. The second derivative, as its name implies, is the derivative of a derivative or a rate of change of a rate of change in the same way that acceleration is the rate of change of speed. What is interesting here is that based off of the size of the kernel used for differentiation and the amount of gradient information in the original image, there can be an overlap in the output of the second derivative, which is useful for potentially obtaining information about the thickness of a blurry line. In more mathematical terms, the second derivative represents...
the concavity and convexity of the image in the direction of the derivative. Some examples will be given in the next figure.

In Figure 3.30, the original image, the first y derivative image, and the second y derivative image of a line are shown. The reason this image is used instead of the square in Figure 3.25 is because the lines on the square are too sharp, and there is very little gradient information. As shown in Figure 3.30, the image is somewhat blurred, resulting in much more gradient information.

![Figure 3.30: (a) Original image (line) (b) y derivative (c) second y derivative](image)

The concavity is shown in the center of the second y derivative output. Moreover, the third image is simply a derivative of the second image. If the second image was simply two single pixel thick lines due to the sharpness of the image, the third image would be completely black. The reason is that the if the kernel is centered along the white line, then the sum of the multiplication of the kernel and the pixels it is overlapping becomes zero. An example of this occurrence is shown in Figure 3.31.

![Figure 3.31: zero second derivative example](image)

\[-1 \times 0 - 1 \times 0 - 1 \times 0 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 1 \times 0 + 1 \times 0 + 1 \times 0 = 0\]
The pixel that this kernel is centered around will be set to zero, or black. Now, a TB bacillus from
the TB image using the first and second derivatives of a TB bacillus is shown in Figure 3.32.

![Figure 3.32: (a) Original image (bacilli) (b) y derivative (c) second y derivative](image)

The potential of the second derivative is shown. Instead of using a color threshold, if the thickness
of the object is known, the body of an object can be found using only second derivatives. This
solution to finding objects can eliminate a large amount of noise, which is everything but what is
being looked for.

Now the second goal of the vessel filter has been accomplished, which was creating a
Hessian matrix of length $2 \times 2 \times n$, where $n$ was the number of pixels in the original image. The
third goal was to determine structure from the eigenvalues of the Hessian. The Hessian matrix
gives the local maxima and minima of a multivariable function, which in this case, is the image.
This concept is similar to how the second derivative test gives the local maxima and minima of a
single variable function. First, a demonstration using the second derivative test with the single
variable case using the function $f(x) = x^2$ will be given, which is simply a parabola. The second
derivative test says that for the critical points of the first derivative, which are points that cause the
function to equal 0 when plugged into the second derivative, the output can be used to determine
if it is a local maximum or a local minimum. If the number is negative then it is a local maximum,
and if it is positive then it is a local minimum. The math is worked out in Figure 3.33.
It is shown that the critical point is at 0. Plugging that into the second derivative function gives a positive value, which means that the only saddle point is a local minimum at \( x = 0 \). The Hessian is simply an expansion of this reasoning into multiple dimensions, except that in this case, the behavior is determined by the eigenvalues of the matrix, which, for the single variable case, are analogous to the output of the second derivative when plugging in the critical points. When an eigenvalue is real and positive, it represents a local minimum. When an eigenvalue is real and negative, the output represents a local maximum. Alternatively, if an eigenvalue is 0, the output represents flatness. Furthermore, since a Hessian Matrix is always real and symmetric, the eigenvalues will always be real. A simple example of how the eigenvalues are used will be given later.

As noted in the ‘Multiscale Vessel Enhancement’ paper, the eigenvalues are analyzed to extract the ratio in curvature between the principle directions in which the local second order structure can be decomposed [28]. The following explanations assume the ordering for eigenvalues shown in equation 13, where the first eigenvalue is denoted as the smallest magnitude.

\[
(|\lambda_1| \leq |\lambda_2|) \tag{12}
\]

Once the above equation is held true, structures can be identified according to Table 3.1 as denoted in the ‘Multiscale Vessel Enhancement’ paper.
The notation $L$ stands for a low number near zero, while $H-$ stands for a large negative number and $H+$ stands for a large positive number. As seen from the above table, a tubular structure is associated with a low value for the first eigenvalue, where an associated large negative number for the second eigenvalue corresponds to a bright structure. This depiction accurately describes a TB bacillus, a bright tubular structure, which makes sense according to the previous analysis.

When the eigenvalue is positive, the output is a minimum, and when the eigenvalue is negative, the output is a maximum, as shown by $\lambda_2$ and its relationship to bright and dark objects. Intuitively, the largest eigenvalue is associated with the fastest curvature in the image, while the smallest eigenvalue is associated with the direction of slowest curvature. The associated eigenvectors give the principal directions of curvature. Again, without the knowledge that the eigenvectors are oriented along the directions of maximum curvature and the orthogonal direction, the associated eigenvalues hold no meaning. A curvature near zero, having very little second derivative information, would mean that the structure is extending off into space, while a large or fast curvature means that the structure changes quickly, denoting a thin width. These two eigenvalues together exemplify a 2D tubular structure. A visualization of a tubular structure in 2D space labeled with eigenvalues in the principal directions of curvature is shown in Figure 3.34.

<table>
<thead>
<tr>
<th>2D</th>
<th>Orientation pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_1$</td>
<td>$\lambda_2$</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>L</td>
<td>H-</td>
</tr>
<tr>
<td>L</td>
<td>H+</td>
</tr>
<tr>
<td>H-</td>
<td>H-</td>
</tr>
<tr>
<td>H+</td>
<td>H+</td>
</tr>
</tbody>
</table>

Table 3.1: Eigenvalue structure table [28]
A simple example in Figure 3.36 showing the math and relating it to Table 3.1 is shown as related to the Hessian matrix of some square vicinity that is part of the bacillus shown in Figure 3.35.

Figure 3.34: principle directions of curvature visual

Figure 3.35: image of a bacillus with a square patch cropped out
The eigenvectors showing the directions of greatest curvature, labeled with a red arrow, and of the orthogonal direction of curvature, labeled by the blue arrow, is shown in Figure 3.37, where \( v_1 \) is associated with \( \lambda_1 \) and where \( v_2 \) is associated with \( \lambda_2 \). The square patch being analyzed is overlaid onto the graph of the eigenvectors to see how they line up with the actual curvature in the image.

\[
H = \begin{bmatrix}
-16.75 & 18.84 \\
18.84 & -24.32
\end{bmatrix}
\]

\[
l_{xx} = -16.75 \\
l_{yy} = -24.32 \\
l_{xy} = 18.84
\]

\[
\text{Trace} = l_{xx} + l_{yy} \\
\text{Determinant} = l_{xx} l_{yy} - l_{xy}^2
\]

\[
\lambda_1 \lambda_2 = \text{Determinant} = 52.41 \\
\lambda_1 + \lambda_2 = \text{Trace} = -41.07
\]

\[
(|\lambda_1| \leq |\lambda_2|)
\]

\[
\lambda_1 = -1.318 \\
\lambda_2 = -39.752
\]

Meaning: Part of a bright Tubular Structure

Figure 3.36: Figure 3.31 refers to equation 10 and 11 for the patch
This example means that the point that the second derivative kernel is centered around is a high point since $\lambda_1$ and $\lambda_2$ are both negative, giving either a bright tubular or bright blob-like structure according to Table 3.1. Then, by looking at the difference, it is shown that one is much larger than the other, which means that the pixel the kernel is centered around is most likely part of an elongated object.

Now the third goal of the vessel filter has been accomplished, which was determining structure from a pixel patch corresponding to the size of the second derivative kernel used. The fourth and final goal was to use the equations from Frangi to identify and tune the vessel filter. Given this information about eigenvalues, Frangi proposes a vesselness measure in the Multiscale vessel enhancement paper. The 2D version is detailed in equations 13-16.

$$V_o(s) = \begin{cases} 
0 & \text{if } \lambda_2 > 0 \\
\exp\left(-\frac{R_\beta^2}{2\beta^2}\right) \left(1 - \exp\left(-\frac{S^2}{2c^2}\right)\right) & \text{otherwise} 
\end{cases} \quad (13)$$

$$R_\beta = \frac{\lambda_1}{\lambda_2} \quad (14)$$

$$S = ||H||_F = \sqrt{\sum_{j \in D} \lambda_j^2} \quad (15)$$
\[ S^2 = \lambda_1^2 + \lambda_2^2 \]  

The \( R_\beta \) variable is the dissimilarity measure, which accounts for the eccentricity of an ellipse. \( S \) is a measure of ‘second order structureness’. \( \beta \) and \( c \) are constants that control how much the various portions of the Vesselness measure are taken into account. Specifically, \( \beta \) is a correction constant that adjusts the filter’s sensitivity to the \( R_\beta \) variable, which accounts for the deviation from a blob-like structure. Furthermore, \( c \) is the correction constant that adjusts the filter’s sensitivity to the \( S \) variable, which accounts for area of high variance (structure). \( \beta = 0.5 \) and \( c = 15 \) were chosen for parameters in this algorithm via visual inspection. It is useful to see how the vessel filter works with different parameters on a section of the tuberculosis image. Figure 3.38 shows a section of a sputum smear or tuberculosis image.

Figure 3.38: section of the tuberculosis image

Figure 3.39 gives the beta and c analysis for a 7x7 kernel on the section of a tuberculosis image.

\[ \begin{align*}
\beta = 0.5 & & \beta = 0.1 & & \beta = 0.1 & & \beta = 0.5 \\
\beta = 0.1 & & \beta = 0.1 & & \beta = 0.1 & & \beta = 0.5 \\
n = 5 & & c = 5 & & c = 15 & & c = 15 \\
\end{align*} \]

Figure 3.39: Beta and c parameter analysis on the vessel filter with tuberculosis
From this, it can be seen that increasing $c$ increases the threshold for the size of the vessels found and that $\beta$ emphasizes connectivity between the vessels. Furthermore, increasing the size of the derivative kernel used from $7 \times 7$ to $25 \times 25$ tunes the filter to vessels of a certain size as seen in Figure 3.40, which gives the kernel size analysis of the vessel filter on the same section of the tuberculosis image by using a $25 \times 25$ size kernel.

$$
\begin{align*}
\beta &= 0.5 \\
& c = 5 \\
\beta &= 0.1 \\
& c = 5 \\
\beta &= 0.1 \\
& c = 15 \\
\beta &= 0.5 \\
& c = 15
\end{align*}
$$

Figure 3.40: Kernel size analysis on the vessel filter with tuberculosis

Now, all four goals of the vessel filter have been accomplished, and the vessel filter has been thoroughly analyzed. To recap, using a matrix of second derivatives called the Hessian, the eigenvalue decomposition is analyzed to find vessels of a certain size based off of the size of the kernel and the equations from Frangi. And when combined with the Bilateral filter, this vessel filter becomes a powerful tool for identifying vessels in an image.

### 3.5 Size Thresholding

Tuberculosis bacilli have been measured to be, on average, 0.2-0.5 um in width and 2-4 um in length. This figure is sometimes skewed by ambient fluorescence in the perceived length to width ratio; however, for the most part, TB bacilli lie in this range. Thus, classifying the bacilli candidates based on size is an important step in any tuberculosis detection algorithm. This step is shown in Figure 3.41, where a rotated bounding box is used to determine width and height before thresholding for size based on the ratio between the two.
Most methods for retrieving size information depend on a binary image. The binary image of the bacillus shown previously is shown in Figure 3.42.

Figure 3.41: Last four sections of the algorithm flowchart (size thresholding)

As stated previously, there are various ways to obtain size information from a binary image. One could simply count the number of white pixels programmatically using something called an 8-
connected component algorithm. Unfortunately, that would only give an estimated size. A circle could potentially have the same number of pixels as a bacillus. This work's algorithm uses a rotated bounding box of the bacilli to estimate the height and width. A rotated bounding box is the smallest rectangle that can completely enclose an object. Figure 3.43 shows the binary image of the bacillus with a rotated bounding box around it.

![Figure 3.43: TB bacillus with rotated bounding box](image)

Using this rotated bounding box, the smallest distance between edges is the width, and the largest distance between edges is the height. But it isn’t so simple to get this information. First, before finding the rotated bounding box, the external contour must be found, which is simply the outline of the object. Contour and bounding box functionality are readily available in many image processing modules such as Open CV, but this thesis still covers the background logic. The rotated bounding box is also often called the minimum bounding box. In order to find this bounding box, 3 pieces of information are needed, which consist of the center location, width, and height. To contrast the idea of a rotated bounding box, a normal bounding box is shown in Figure 3.44.
To find the center of the object, one can simply find the mean of the smallest and largest x values and the mean of the smallest and largest y values within the contour array of pixel locations, which is just the array of points that constitutes the outline of the object. Then, the most common algorithm to find the width and height is the ‘rotated calipers’ algorithm developed by Michael Shamos in 1978 [29]. Of a set of points, one would normally find the convex hull, the outermost points; however, the external contour of the object has already been found in this algorithm, which is essentially the same thing. The next step is to begin with a bounding rectangle around all of the points in the contour that has an edge coincident with the line between two points. A depiction is shown in Figure 3.45.

![Figure 3.44: TB bacillus with a normal bounding box](image)

![Figure 3.45: Rotated Calipers method depiction](image)
Then, an iteration through each rectangle that has an edge coincident with the line between two points is done, and the smallest rectangle becomes the rotated bounding box of the object. From that, the smallest distance between two points is the width, and the largest distance between two points is the length. A computer annotated bounding box on a bacillus is shown in Figure 3.46.

![Figure 3.46: Rotated bounding box of a bacillus](image1)

This width and height is in number of pixels, which is not intuitive when trying to threshold for size. So, given the average size of a bacilli, noted earlier, the system is tuned by dividing by a scale factor to convert to micro meters as shown in Figure 3.47.

![Figure 3.47: Pixel to um scaled bacillus dimensions](image2)
The size threshold parameters were chosen based off of the actual size of tuberculosis bacilli, which range between 0.2-0.5 um in width and between 2-4 um in length. The parameters include a limit to the maximum width, a limit to the minim length, and a ratio range between the width and the height. The process taken is as follows:

1. Identify the noisiest image.
2. Run it through the bilateral and Vessel Filter
3. Using a large range for width and height, find all potential bacilli in the image, with respect to a hand labeled dataset, and programmatically calculate the width and height via a rotated bounding box for each bacillus.
4. Use a ‘pixels per metric’ variable to convert the pixel size of the smallest bacillus width in the image to 0.2.
5. Limit the width to less than 0.5.
6. Limit the length to more than 0.1 less than the average length of all 0.2 width bacilli in the image.
7. Find the ratio of the average length of each 0.2 um bacillus to 0.2. This number is the center from which the ratio range will extend.
8. Determine the best range by increasing the deviation on each side of the number found in step seven by a multiple of one.

The best thresholding values by following the above steps consist of a deviation of 2, according to the eighth step, and are as follows:

1. Width < 0.5
2. Length > 0.8
3. Length > 2.75*Width
4. Length < 6.75*Width
At this point there will be quite a few false positives because the vessel filter is not perfect. Some of the irregular objects have vessel-like structures of the right size on top of them even with the bilateral filter, so there is a need to separate these from the actual tuberculosis bacilli as seen in the binary output of the vessel filter with the bilateral filter in Figure 3.48.

![Figure 3.48: Output of bilateral and vessel filtering](image)

### 3.6 Background Subtraction

Background subtraction can have many meanings. For this work, the foreground is the object, the bacillus in question. The background is everything else. This work uses background subtraction techniques to ensure that a detected bacillus is not actually part of a larger structure. In this situation, the bacilli should be brighter than the background, which means that the greyscale pixel values for the bacilli, ranging from 0 to 255, should be higher than the greyscale pixel values for the background. This step is shown in Figure 3.49.
The most comprehensive way to do this foreground confirmation is to test whether or not the mean of the greyscale pixel values for the bacilli is higher than the greyscale pixel values of the background, and if so, by how much. Programmatically, however, it is much simpler to take a square sample of pixels around the center of the bacilli, since the center location is already known. Then, the mean of all of the greyscale pixel values of a 24x24 pixel patch around the center of the bacilli can be subtracted, since the average bacillus would fit within that. Then by varying the threshold for how much that difference is, a threshold for bacilli in the foreground can be done. Figure 3.50 shows the output of the bilateral and vessel filter without size thresholding, where the identified bacilli have a green circle around them.

Figure 3.49: Last four sections of the algorithm flowchart (background subtraction)
As shown in the above image, there are a number of false positives. Now, implementing the background subtraction method mentioned, the output is shown in Figure 3.51.
Figure 3.51: output TB image with background subtraction for foreground verification
3.7 Flow Chart

Read in image

Convert to greyscale

Bilateral Filter

Vessel Filter

Find width and height of rotated bounding box using contour points

Threshold for size using width and height

Background Subtraction for bacilli confirmation

Circle bacilli on original image

Figure 3.52: Full algorithm outline
4. METHODS

Statistical analysis, in general, of data is highly varied; however, analysis within the machine learning and computer vision community tend to center around precision and recall as the default methods of drawing conclusions about accuracy as opposed to the generic definition for accuracy as seen below.

\[
\text{accuracy} = \frac{(\text{True Positive} + \text{True Negative})}{(\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative})}
\]  \hspace{1cm} (18)

Individually, they are both important, but having a high precision and a low recall is not always a highly valued result, which is why most people draw a precision/recall (PR) curve with varied parameter changes, or in the case of a set parameter, an F-score. These metrics are defined below, where true positives are the number of correctly identified bacilli (bacilli found that are actually bacilli), false positives are the number of incorrectly identified bacilli (fragments thought to be bacilli), and false negatives are the number of bacilli incorrectly labeled as fragments (bacilli thought to be fragments). True negatives, which would be fragments correctly identified as fragments, are not taken into account because there are simply too many of them with respect to the number of actual bacilli, and there is no meaningful reason to analyze this number. A graphical explanation is shown in Figure 4.1.
Figure 4.1: Precision and Recall graphics [31]

**Precision**: the number of true positives divided by the sum of true positives and false positives

**Recall**: the number of true positives divided by the sum of true positives and false negatives

The F1-score gives the relationship between precision and recall via a weighted harmonic mean and is favored over accuracy when given an unbalanced dataset, as explained earlier, in an attempt to measure the overall effectiveness of the algorithm, especially in one-class systems.

\[
F_\beta = \frac{(1 + \beta^2)(true \ positive)}{(1 + \beta^2)(true \ positive) + (\beta^2)(false \ negative) + (false \ positive)}
\]

(16)

\[
F_1 = 2 \left( \frac{1}{recall} + \frac{1}{precision} \right) = \frac{2(precision)(recall)}{precision + recall}
\]

(17)
4.1 Object Level Analysis

For object level analysis, accuracy was calculated in two different ways. First, the precision, recall, and F1 score with respect to all annotated bacilli were calculated. Second, the precision, recall, and F1 score with respect to bacilli in each of the 92 images were calculated, giving an average across each image. True positives were computationally analyzed by checking to see if a manual annotation was within a 17-pixel radius of the center of a bacillus. Once a point is associated with a bacillus, it is removed from the array of bacilli points so that it is not double counted. The data comparison of this algorithm with Chang’s algorithm is shown in Figure 4.2 because both algorithms use the same dataset. Chang’s algorithm can be tuned to be better or worse with recall or precision, but for this test, the default algorithm from the code made available was used. Only the red label outputs in that default algorithm were taken into account because the other labels, which were labeled green were below the threshold level of 0.2 that was already set.

![Figure 4.2: Object level Precision, Recall, and F1-score metrics](chart.png)

In Figure 4.2, it should be noted that the overall F1 score is slightly works than Chang et al.’s algorithm, while the average F1 score is slightly better. The reason behind this occurrence is that
some images contain more than 500 bacilli, while some slides contain less than 5 bacilli. Furthermore, in those images with more than 500 bacilli, it seems like many of the bacilli were not counted by the technicians that labeled these images. Since those images with large amounts of bacilli are skewing the accuracy reading, the average F1 score seems like a better analysis as is confirmed by the slide level analysis section.

4.2 Slide Level Analysis

Slide level analysis as noted in the literature review sections requires ten to one hundred fields, or images, per slide. The dataset this work uses only has 92 annotated images. Note that it does include over 300 other positive and negative slide images; however, none of them have been annotated. Assuming that the negative slide samples have absolutely no bacilli is a fallacy, and they should be labeled. The process of labeling was not undertaken for this work because this sort of counting procedure should be done by a professional, which is why the creators of this dataset should be applauded for collecting the data, for finding the necessary experts, and for making it all public.

Due to the fact that there are not enough annotated images for slide level analysis, the data is analyzed on a field by field, image by image, basis. This method relies on the way that actual technicians count and report bacilli in an image. The method of reporting, which can also be found in the literature review section, is shown in Table 4.1.

<table>
<thead>
<tr>
<th>What you see (200x)</th>
<th>What you see (400x)</th>
<th>What to report</th>
</tr>
</thead>
<tbody>
<tr>
<td>No AFB in one length</td>
<td>No AFB in one length</td>
<td>No AFB observed</td>
</tr>
<tr>
<td>1-4 AFB in one length</td>
<td>1-2 AFB in one length</td>
<td>Confirmation required*</td>
</tr>
<tr>
<td>5-49 AFB in one length</td>
<td>3-24 AFB in one length</td>
<td>Scanty</td>
</tr>
<tr>
<td>3-24 AFB in one field</td>
<td>1-6 AFB in one field</td>
<td>1+</td>
</tr>
<tr>
<td>25-250 AFB in one field</td>
<td>7-60 AFB in one field</td>
<td>2+</td>
</tr>
<tr>
<td>&gt;250 AFB in one field</td>
<td>&gt;60 AFB in one field</td>
<td>3+</td>
</tr>
</tbody>
</table>

Table 4.1: Level of Infectiousness for Fluorescence Sputum Smears with AFB [16]
As stated earlier, a length normally denotes a straight line, horizontally across the length of the smear, which can mean any number between 10 and 100 fields. The top three rows of the infectiousness chart specify what happens when there are too few bacilli in the slide. When this situation occurs, the technician can only add up all of the bacilli seen in one length of the smear to make a statement about the level of infectiousness. These three sections are considered 0+ for the purpose of this analysis. Otherwise, if there are at least 1-5 bacilli per field of view seen on the slide, the level of infectiousness is then determined based on the average number of bacilli seen per field. This work uses this line of thought to determine what the performance of the system might be in determining the overall level of infectiousness for a slide because the output infectiousness level is the number average number of bacilli in one field according to Table 4.1. In this section, a 4 label confusion table is presented as well as the associated precision, recall, and F1 score computations for the level of infectiousness estimate per field as shown in Table 4.2 and Table 4.3.

<table>
<thead>
<tr>
<th>Actual</th>
<th>0+</th>
<th>1+</th>
<th>2+</th>
<th>3+</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>18</td>
<td>12</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1+</td>
<td>7</td>
<td>30</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2+</td>
<td>1</td>
<td>0</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>3+</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 4.2: Level of infectiousness confusion matrix for this work’s algorithm
A confusion matrix is a clear way to visualize data. It tells us information such as where the prediction went wrong and by how much under the different categories. In this case, summing a column shows the total number predicted for that class, and summing a row shows the total actual number of that class. The precision and recall of field level infectiousness accuracy is shown in Figure 4.2.

<table>
<thead>
<tr>
<th></th>
<th>0+</th>
<th>1+</th>
<th>2+</th>
<th>3+</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>13</td>
<td>17</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1+</td>
<td>2</td>
<td>27</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>2+</td>
<td>0</td>
<td>1</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>3+</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.3: Level of infectiousness confusion matrix for Chang’s algorithm

The F1-score for field level infectiousness accuracy is shown in Figure 4.3.
Figure 4.4: Field level F1-score
5. CONCLUSION

The algorithm this thesis describes is an image processing algorithm in Python that can accurately determine level of infectiousness using the field level classifications dictated by the World Health Organization (WHO). This algorithm is comparable to the previous algorithm that was tested on this dataset in some ways and more effective in others, considering various performance metrics, but it is centered more on the actual depiction of Mycobacterium tuberculosis (MTB), in terms of size and shape, as opposed to the much more common machine learning approach with a bevy of descriptors. As shown from the methods section, the algorithm this work describes has a lower recall but a higher precision than the test program Chang et al. have made available in terms of object level analysis; however, since the F1-score takes into account both precision and recall, this paper is within 5% of the Chang paper with this work, being higher in terms of average F1-score results and lower in terms of overall F1-score results. This difference can be explained due to the large number of bacilli in only a few images that skews the overall F1-score. Furthermore, when analyzing level of infection at the field level, this algorithm is much more effective in the 1+, 2+, and 3+ categories in terms of both precision and recall, except for the recall in the 3+ category. The rule based algorithm described in this work seems much lower in terms of the 3+ recall category even though it only missed one due to the low number of 3+ images, of which there were five. Furthermore, this work retains an 88% effectiveness rating as judged by the F1-score in the 3+ category, a 92% rating in the 2+ category, a 75% rating in the 1+ category, and a 64% rating in the 0+ category. The 0+ category consists of three categories, and normally needs a second opinion by several trained technicians. Overall, the algorithm described in this work is more effective at field level infectiousness determination; however, it is possible that Chang et al.’s algorithm could become more effective by changing the threshold from the default program to something lower than 0.2 to be more precise in what is detected even though it would lower the object level recall.
6. REFERENCES


[10] Derivative work, InkScape image by Henry Mühlpfordt licensed with Cc-by-sa-3.0


[23] image by Horst Frank licensed with Cc-by-sa-3.0


[25] created by Adrian Pingstone, based on the original created by Edward H. Adelson


[31] image by Walber licensed with Cc-by-sa-4.0
Appendix A – Python Code Libraries and Source Code

Imported Statements from Python Libraries:

import cv2
import numpy as np
import time
from skimage.feature import hessian_matrix_eigvals
from matplotlib import pyplot as plt
from scipy.ndimage.filters import convolve
import scipy
import scipy.misc
from scipy import ndimage
from imutils import perspective
from imutils import contours
import numpy as np
import imutils
import cv2
from scipy.spatial import distance as dist
import scipy.misc
import os
from tempfile import TemporaryFile
import re
from natsort import natsort
import pandas as pd

Link to Code:

https://github.com/sclaybon/Tuberculosis-Image-Processing-/blob/master/TB_Analyzer_README
Appendix B - Example Images

Example 1 Original Image
Example 1 Output Image
Example 2 Original Image
Example 2 Output Image
Example 3 Original Image
Example 3 Output Image
Example 4 Original Image
Example 4 Output Image