

https://dspace.org/

Application of Computer Vision and Machine Learning for classification of mycobacterium Tuberculosis (TB) cultures to assist disease diagnostics

Venkataraman, Karthikeyan

2019-10

https://hdl.handle.net/10539/29627

Venkataraman, Karthikeyan, (2019) Application of computer vision and machine learning for classification of Mycobacterium Tuberculosis (TB) cultures to assist disease diagnostics, University of the Witwatersrand, https://hdl.handle.net/10539/29627

Downloaded from a DSpace repository. https://dspace.org/

Application of Computer Vision and Machine Learning for classification of *Mycobacterium Tuberculosis (TB)* cultures to assist disease diagnostics

School of Computer Science and Applied Mathematics
University of the Witwatersrand

Karthikeyan Venkataraman
1817218

Supervised by
Dr. Richard Klein
Dr. Pravesh Ranchod

October 13, 2019



A Research Report submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Master of Computer Science by Coursework and Research Report

Abstract

Diagnosis of Tuberculosis (TB) by the analysis of sputum samples is a manual activity that requires the sputum culture samples to be analysed after a gestation period between 6 to 12 weeks following the collection of the sample from the patient. The current method of TB diagnosis at the National Heath Laboratory Service (NHLS) is a manual process that involves photographing the culture samples by the laboratory technician using a camera phone, among other steps. It is proposed to use these photographs to automate the process of diagnosis by using Computer Vision and Machine Learning technologies to detect the presence of Tuberculosis bacteria.

A photograph of a culture sample or plate, consists of 48 unique circular regions or wells, each of which is classified by the machine learning program as Positive, Negative, Contaminated or Condensation (indicating presence of water bubbles).

This classification of each well is done using a combination of Image Processing, Computer Vision and Machine Learning (ML) techniques. Two ML programs, in particular, were used for this task - Gaussian Mixture Models (GMM) and Random Forest (RF). At the time of writing of this report, a classification accuracy of 77.6% using 25 training images was obtained. The results from each ML technique are compared and approaches to further improve the classsification accuracy are discussed.

A data flow and machine learning pipleline was developed in Python 3 and the code has been open-sourced on GitHub at https://github.com/karthik111/TB-culture-classification-ML allowing future work to be undertaken. The Image Processing and Computer Vision techniques implemented here are agnostic of the downstream ML technique allowing other ML approaches to be explored.

This work establishes the feasibility of an automated approach for classification of sputum samples to aid TB diagnosis and describes approaches that if taken, can further improve the classification accuracy.

Declaration

I declare that this thesis is my own, unaided work. It is being submitted for the Degree of Master of Science at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.



Karthikeyan Venkataraman
October 13, 2019

Dedication

To my dear wife Priya, children Arhant, Himani, and my mother, Sarala, a survivor of Tuberculoid Meningitis, whose love, support and patience, and the blessings of the Almighty that allows me to pursue my passion for lifelong learning and exploration.

To also my younger brother, Srihari, whose dedication to his profession and service to his country, has been a continuing source of inspiration to me in my own endeavors.

Acknowledgements

I would like to place on record my sincere thanks to the following individuals who made this possible:

1. Dr. Pravesh Ranchod, Senior Lecturer, Wits University who suggested the idea of this project and put me in touch with the National Health Laboratory Service (NHLS). Pravesh's guidance, helpful and ever-friendly spirit in fielding so many of my questions around the program during the beginning of 2017 was a key factor in my decision to embark on this Master's program while working full time at IBM and later now at Accenture.

2. Dr. Julian Peters and the rest of the team at the NHLS for letting me have a glimpse into their fascinating their world of TB diagostics and research and explaining their TB culture analysis processes to me. As I worked through the system implementation, I continued to reach out to Julian for clarifications and validation of the early results as I tried multiple approaches.

3. Dr. Richard Klein, Lecturer, Wits University was my primary guide for this project. Richard's excellent lectures on Computer Vision first sparked my interest in Computer Vision and helped build a solid base to apply these techniques on a practical problem. A number of sessions with Richard helped sharpen the approach and analysis of the techniques used. Richard was an invaluable sounding board and guide as I went about this task.

4. My colleagues and managers first at IBM and later at Accenture were invaluable in their support and encouragement during this project.

Contents

Preface
Abstract i
Declaration ii
Acknowledgements iv
Table of Contents v
List of Figures vii
List of Tables ix
Nomenclature x

1 Introduction **1**
1.1 Tuberculosis 1
1.2 The TB Culture Classification Process 3
1.3 An automation approach for TB diagnosis 7

2 Background and Related Work **9**
2.1 Diagnostic methods for TB 9
2.2 Machine Learning approaches for TB diagnosis 10
2.2.1 Use of chest radiographs 10
2.2.2 Photographic images of sputum samples taken under a microscope . . . 10
2.2.3 Photographic images of sputum samples as seen through the naked eye 11

3 Methodology **12**
3.1 System Design and Expected Benefits 12
3.1.1 Key System Components 12
3.1.2 Expected benefits of the system 12
3.2 Data Preparation and Feature Extraction 13
3.3 Each well as a training sample 14
3.4 Obtaining, Processing and Loading Training Data 15
3.5 Identifying regions of interest and Feature Extraction 16
3.5.1 Well detection with Hough Circles 17
3.5.2 Extracting Pixels within the Well using Masking 17
3.5.3 Feature Extraction and Reduction using Histogram based approaches . 18
3.5.4 Feature Engineering using LBP to analyze textures of each well 21
3.5.5 Limitations in the use of contours to extract region of interest for Texture analysis 25
3.5.6 Feature Engineering Using HSV values of each well 27

3.6 Learning, Inferencing and Validation with Machine Learning Models 28
3.7 Gaussian Mixture Model (GMM) Classifier 28
3.7.1 Training the Gaussian Mixture Models 29
3.7.2 Using the Akaike information criterion and Bayesian information criterion 30
3.7.3 Training or Fitting the GMM model for each category of well 31
3.7.4 Inferencing from the GMM model for each category of well 31
3.8 Random Forest Classifier 31

4 Results and Discussion **34**
4.1 Classification results on a Test data set 34
4.2 Results with GMM 34
4.3 Results with Random Forest Classifier 36

5 Conclusions and Future Work **39**
5.1 Conclusions 39
5.2 Future Work 40

References **46**

A MPN Calculator **47**

B Relevant Code Listings **48**
B.1 Feature extraction from well regions within an image of a sputum culture assay plate 48
B.2 Extraction of image texture information within a well 48
B.3 Use of contours to create a rectangular region around a well 48
B.4 Training the GMM classifier 49
B.5 Inferencing using the results returned by the GMM classifier 49

List of Figures

1.1 Deaths from TB, by world region . . . 2
1.2 Positive well samples within a plate . . . 4
1.3 Negative well samples within a plate . . . 5
1.4 Condensation or Water samples . . . 5
1.5 Contaminated samples . . . 6
1.6 Patient Label on samples . . . 6

3.1 Proposed process flow . . . 12
3.2 Data flow pipeline . . . 13
3.3 Two adjacent positive wells . . . 14
3.4 Two adjacent negative wells . . . 14
3.5 Two adjacent water or condensation wells . . . 15
3.6 Two adjacent contaminated wells . . . 15
3.7 A culture assay plate or tray with 48 wells and their class labels for training . . . 16
3.8 Another training pair with Positive, Negative and Contaminated well samples . . . 17
3.9 Use of Hough Circles algorithm to identify wells within a plate . . . 18
3.10 Use of Masks to obtain grayscale pixel values within each well . . . 18
3.11 25 Bin Histogram of Grayscale Values for a Positive well . . . 19
3.12 25 Bin Histogram of Grayscale Values for another Positive well . . . 19
3.13 25 Bin Histogram of Grayscale Values for a Negative well . . . 20
3.14 25 Bin Histogram of Grayscale Values for another Negative well . . . 20
3.15 25 Bin Histogram of Grayscale Values for a Water well . . . 21
3.16 25 Bin Histogram of Grayscale Values for a Contaminated well . . . 21
3.17 LBP for a Positive well sample . . . 22
3.18 LBP Histogram for a Positive well sample . . . 23
3.19 LBP for another Positive well sample . . . 23
3.20 LBP for a Negative well sample . . . 24
3.21 LBP for another Negative well sample . . . 24
3.22 LBP for a Condensation or Water well sample . . . 25
3.23 Extracted region of interest using masking for texture analysis . . . 26
3.24 Rectangular region of interest for texture analysis . . . 26
3.25 All contours drawn within the region of interest . . . 27
3.26 100 Bin Histogram of HSV Values for a Positive sample . . . 28
3.27 100 Bin Histogram of HSV Values for a Negative sample . . . 28
3.28 AIC and BIC curves for Positive and Negative classes . . . 30
3.29 Validation Curve for Random Forest classifier . . . 32

4.1 Confusion Matrix for GMMs with equal values of K (number of Gaussians) . . 35
4.2 Confusion Matrix for GMMs with different values of K (number of Gaussians) 36
4.3 Confusion Matrix for Random Forest classifier with a smaller sized training data set 37
4.4 Confusion Matrix for Random Forest classifier with a larger sized training data set 38

5.1 Learning Curves for Random Forest classifier 41
5.2 Confusion Matrix for Random Forest classifier with two different sets of hyper-parameter values shown in Table 5.1 43

A.1 MPN Calculator 47

List of Tables

3.1 Accuracy levels observed for each fold within a 5-fold cross validation 32
3.2 Number of training samples 33

4.1 Equal number of Gaussians per category 34
4.2 Modified number of Gaussians per category 35

5.1 Changes to select hyper-parameter values in the Random Forest Classifier . . . 43

Chapter 1

Introduction

1.1 Tuberculosis

TUBERCULOSIS, also known as TB is one of three major diseases, the other two being Malaria and HIV/AIDS, that causes the most number of fatalities among human beings. In 2016 alone, 1.2 million deaths were attributed to TB. This disease attacks not just human beings, but also many species of birds and mammals.

TB is caused by a bacterium from the *Mycobacterium tuberculosis* complex (WHO 2018). Only 5 to 15% of people who are infected with this bacteria, eventually exhibit symptoms of the TB disease. It is estimated by the World Health Organization (WHO) and the Institute for Health Metric and Evaluation (IHME) that approximately 25% of the world's population is infected with the bacterium but does not suffer from the disease. This is known as 'Latent TB' (Roser *et al.* 2018).

Being an airborne disease, TB spreads by exposure to the cough, sneeze or sputum of a person suffering from the disease. A person with latent TB cannot infect other people. However, the existing large human population with latent TB is a potential source of TB infection, if their condition changes from latent to active in the future.

People with latent TB and with weak immune systems, smokers or HIV-positive individuals are at a greater risk of eventually developing 'Active TB'. Latent TB infections can become active later in life as the immune system weakens due to advancing age or lifestyle related causes.

There are two kinds of tests to detect latent TB or the presence of TB bacteria in the body - the TB skin test (TST) and TB blood tests. When a person with latent TB infection (LTBI) develops active TB, other kinds of tests such as chest x-rays or sputum culture tests are used to detect and diagnose the presence of the TB disease.

A person with Active TB exhibits symptoms such as coughing, often with blood in the sputum, unexplained loss of weight, loss of appetite, night sweats, fever, fatigue and chills. While TB infects the lungs in the majority of the cases, it can also affect other parts of the body such as the brain. Left untreated, a person with active TB has a 70% chance of dying, which is caused by the TB bacteria destroying the patient's lungs, eventually leading to suffocation.

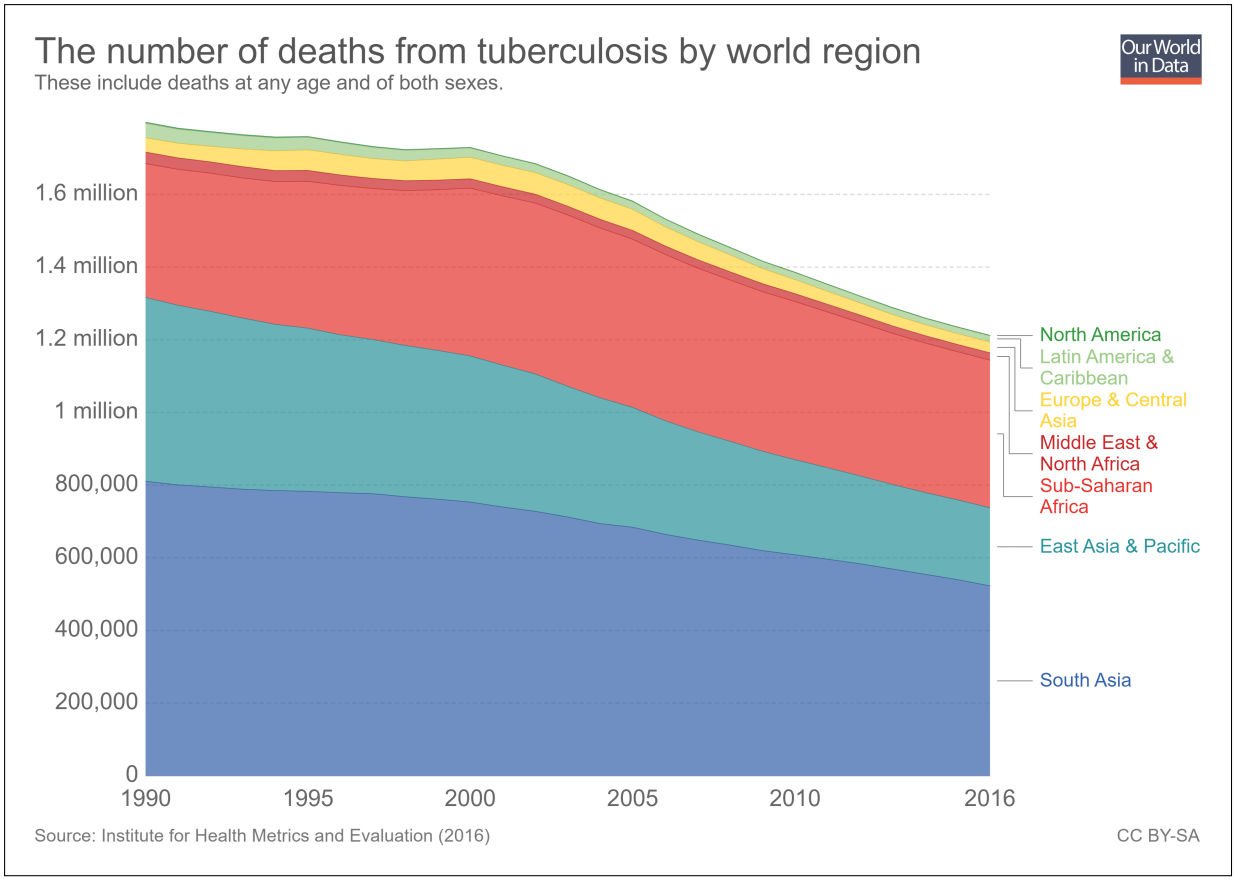


Figure 1.1: The number of deaths from TB by world region

While the number of people with latent or active TB has been increasing with growth of the human population, the number of deaths caused by the disease has declined since 1990 when 1.8 million people were killed by this disease.

Figure 1.1 shows the number of deaths from TB in different parts of the world between 1990 and 2016. While the overall incidence of deaths worldwide has decreased, it is concerning to note that in Sub-Saharan Africa, the number of deaths by TB has marginally increased to 406,554 deaths in 2016 from 368,376 deaths in 1990.

TB Meningitis, another form of TB, is caused by the late detection or treatment of the disease. It can be particularly devastating for children, leading to dealth or permanent disability. According to Dr. Regan Solomons, paediatric neurologist at the Tygerberg Children's Hospital in Cape Town, "It only takes a few droplets of a sneeze or cough for TB to spread from an adult to a child. TB will first go to the lungs and then spread to the brain, causing TB meningitis."(Cloete 2016)

Whilst the currently available means against the disease (with sufficient investment) can feasibly reduce the number of TB cases and deaths to very low numbers (a 95% reduction from 2015 case numbers is targeted by 2035), the WHO and Stop TB Partnership point out that adequate means are not yet available to completely eradicate the disease. For complete eradication, research investment and development would be required for new tools to be developed to combat the disease.

Therefore, the early, cost-effective and reliable detection of TB is a critical endeavour for the benefit of humanity at large.

1.2 The TB Culture Classification Process

Mycobacteria is the name of a genus that consists of over 190 species of pathogens known to cause serious diseases in mammals including Tuberculosis (*Mycobacterium tuberculosis*) and Leprosy (*Mycobacterium leprae*) in humans. The Greek prefix *myco-* means "fungus", alluding to the way mycobacteria have been observed to grow in a mold-like fashion on the surface of cultures.

Mycobacteria Growth Indicator Tube (MGIT) is the name of a clinical method used in the detection and recovery of Mycobacteria. The MGIT Mycobacteria Growth Indicator Tube contains of 4 mL (milliliters) of a chemical compound called modified Middlebrook 7H9 Broth base to which the sputum sample of the patient is added. This medium is one of the most commonly used liquid media for the cultivation of Mycobacteria.

At the National Health Laboratory Service (NHLS) in Johannesburg, TB culture samples are prepared and analysed using a method called Most Probable Number (MPN) Limiting Dilution Assays. Earlier, pictures of the culture samples were taken weekly, however due to high sample numbers, pictures are now taken once in six weeks which is equivalent to a widely used clinical method called MGIT Culture Incubation.

Approximately 50 samples per week are analysed for results with sample collections numbering around 10 samples per day.

All types of clinical specimens, pulmonary as well as extra-pulmonary (except blood and urine), can be processed for primary isolation in the MGIT tube using conventional methods. After the processed specimen is inoculated, the MGIT tube must be continuously monitored either manually or by automated instruments until positive or the end of the testing protocol.

Two plates are prepared for each sample as described below:

1. An RG / BG plate: Certain resuscitation factors (catalyst compounds) are added to the sample to help the growth of differentially culturable bacteria, if any.
2. A Media plate: No resuscitation factors are added to this plate and this serves as a control group to assess growth of bacteria without the presence of any resuscitation factors. In the field (outside the labs), only the media plate is used to assess the growth of Mycobacteria Tuberculosis in the MGIT culture. This perhaps explains the lower detection rate and the longer time to MGIT positivity in clinical settings.

Figure 1.2 shows a sample, also called a plate or assay tray. The plate consists of 48 wells (6 rows by 8 columns) with each well containing an MGIT tube. The six row of wells on a plate are labeled A through F and the eight columns are labeled 1 through 8.

In the RG/BG plate, the top 3 rows (A, B, C) are given the RV catalyst factor and the bottom 3 rows (D, E, F) are given the BG factor. A volume of 450 micro-litre of each respective liquid experiment (RV, BG or media) is dispensed into the appropriate wells. A volume of 50 micro-litre of decontaminated patient sputum is added to the first column (1) and this is adequately

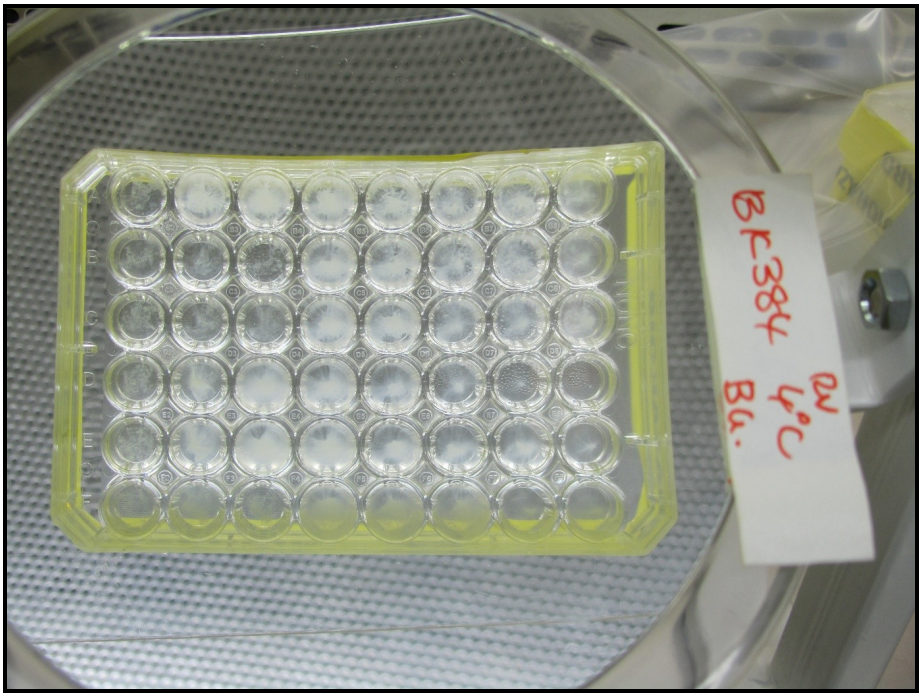


Figure 1.2: Positive well samples within a plate – notice cloudy formation on many of the wells

mixed by using a multi-channel pipette. A 10-fold serial dilution of the mixed sample is carried out in subsequent columns by transferring 50 micro-litre of each mixed column to the next column resulting in one-tenth of the concentration of bacteria from the previous column.

Therefore, when a well flags positive as in Figure 1.2, it gives an indication of the presence of replicating bacteria in under varying conditions (high to low degrees of presence of resuscitating factors)

At the end of the 6 week period, the presence of cloudy (Positive) or clear (Negative) wells are counted within the plate and are entered into a spreadsheet based tool called an MPN calculator [1] which provides an estimate (probabilistic) of the presence or absence of TB in the samples including confidence intervals. A larger number of cloudy wells reduce the boundaries of the confidence intervals. Both RG/BG and Media plate observations are used by the MPN formula in its inference model.

Black or Orange coloration in a well indicate contamination (Figure 1.5) and these are counted as well. Water bubbles may also form in some wells (Figure 1.4) and lastly wells that exhibit no change are are considered to be Negative (Figure 1.3)

Identification of each plate or assay tray is done by means of a label that is affixed with a serial number as shown in Figure 1.6. This serial number links the sample to the patient in the NHLS' records.

[1] depicted in A.1 in the Appendix

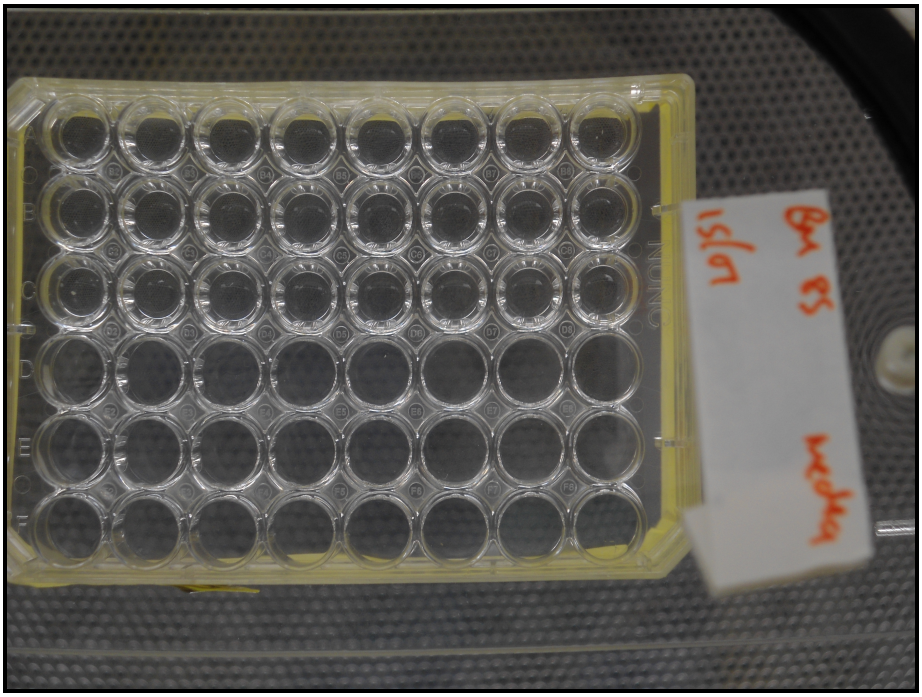


Figure 1.3: Negative well samples within a plate – no coloration within any of the wells after 6 weeks

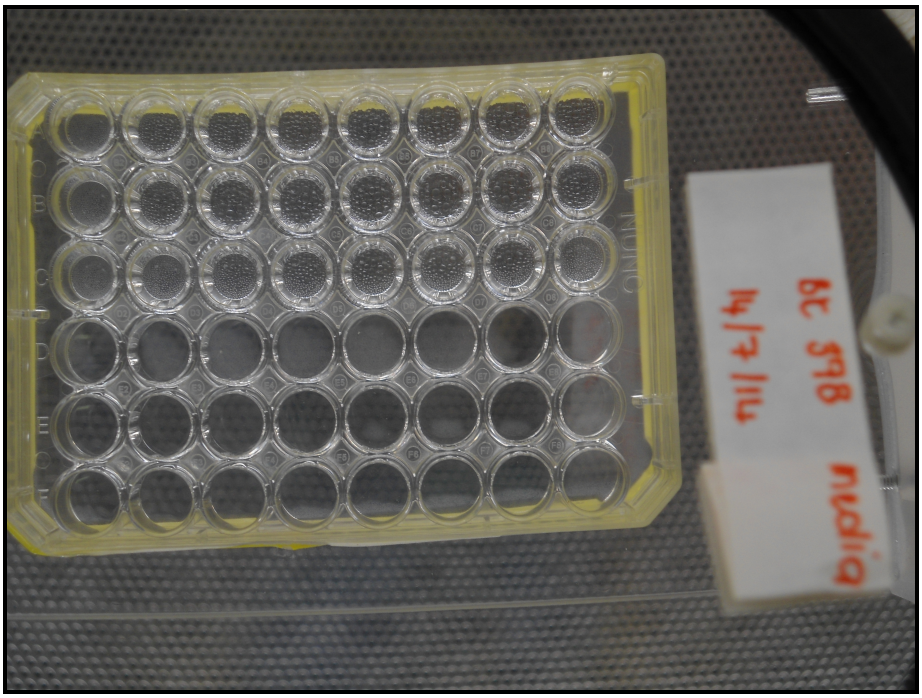


Figure 1.4: Condensation or formation of water bubbles on the wells can sometimes closely resemble Positive samples. Only upon close examination can a human analyst distinguish them from a Positive sample.

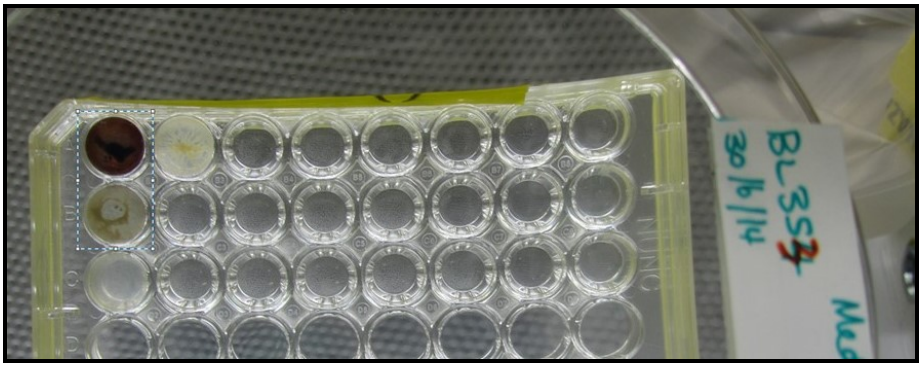


Figure 1.5: Contaminated samples – Black and Orange colorations within any well indicate contamination of that well



Figure 1.6: Labels that are affixed to each sample to identify patient and sample number. Any one of R# or #CP may be used to identify the patient

1.3 An automation approach for TB diagnosis

Automated image classification of *Mycobacterium tuberculosis* culture samples using machine learning algorithms can potentially complement the current manual diagnostic processes that are prevalent today. Currently, the process of recording and analysing images of TB samples is done manually at the NHLS in Johannesburg. The introduction of machine learning and artificial intelligence techniques can help alleviate the current human workload on the researchers that work there.

This work provides a quantitative assessment of whether this is indeed possible and whether machine learning techniques are at a state of readiness where they can be included within a human diagnostic process such as the one used at the NHLS.

The discussion, analysis and outcome of this work is structured along the following lines:

1. A description of the current process of manual data capture and classification of TB culture samples into their respective categories.
2. An outline of a proposed mobile application that can be developed using the Computer Vision and Machine Learning (ML) programs that has been developed to further automate this process at scale. This proposed mobile application is expected to provide specific benefits that are also outlined.
3. The Data Gathering method adopted and the associated Feature Extraction techniques applied on the image of TB cultures. A description of the different Image Processing and Computer Vision techniques used for Feature Extraction is described.
4. Development of the ML programs that performed the Learning and Classification tasks using the data gathered in the above step. Two specific ML algorithms were chosen, namely
   (a) Probabilistic classification using Gaussian Mixture Models (GMM)
   (b) Random Forest classification
5. The behaviour of these algorithms on the chosen data set provides an in-depth understanding of the issues involved and the performance on the TB culture classification task.
6. Full source code, documentation and execution steps have been uploaded to Github to encourage further research and application development using the programs provided. The system is designed in a modular manner allowing any other ML program to be plugged in. The design of the Machine Learning application that was developed using these algorithms is also described in terms of it's key system components.

   The accompanying implementation in Python 3 provides a complete Data Processing and Classification Pipeline with the software components listed below.
   (a) Image Processing on pictures of TB culture samples taken with a Samsung mobile phone camera.
   (b) Feature Extraction
   (c) Model Training on training data set

(d) Hyperparameter tuning using a validation data set

(e) Inferencing on a test data set

(f) Confusion Matrix for classification accuracy analysis

Github: https://github.com/karthik111/TB-culture-classification-ML

7. Analysis of the results obtained by each of the ML learning approaches in terms of classification accuracy and execution timings for both training and testing along with reasons behind the same.

8. Recommendations on the way forward in terms of usage, further projected improvement in classification accuracy anticipated by other techniques beyond the ones implemented are also discussed.

Chapter 2

Background and Related Work

2.1 Diagnostic methods for TB

Since the advent of human civilization, the conquest of disease has been a priority across millenia. While some dreaded diseases such as Smallbox and Rinderpest have been eradicated or permanently eliminated at a global level, there are others like Tuberculosis, HIV/AIDS and Malaria that still exist.

The complete eradication of TB is an extremely challenging goal due to the very high prevalence of latent TB and growth of human population on the two continents that are most impacted by this disease (Asia and Africa). In order to effectively control TB and make progress on the path to eventually eradicating this disease in the future, the following steps are all important:

1. Developing more effective vaccines to prevent TB infection.
2. Developing better drugs to combat multi-drug resistant tuberculosis.
3. Improving diagnostics.

Therefore, the early, cost-effective and reliable diagnosis of TB is vital in the combat of this dreaded disease.

Chest radiographs (X-rays) are a widely used type of medical imaging and a commonly used approach for TB detection. Another approach prevalent is the bacteriological diagnosis of TB or by detecting the presence of the mycobacteria. The following types of pathological tests are conducted for the diagnosis of TB bacteriologically:

1. Sputum Smear Microscopy: Sputum samples are tested for the presence of bacteria under a microscope. The images taken under the microscope can then be used for ML based approaches. A ML approach discussed in Section 2.2.2 used these images.
2. Rapid Molecular Test: This is a molecular test, based on polymerase chain reaction, which detects the DNA of *Mycobacterium tuberculosis* within 2 hours.
3. Culture methods: The WHO defines this test as the current reference standard. The disadvantage of this test lies in duration of evaluation which takes 8 to 12 weeks for the presence of the bacteria to manifest itself in the sample. In practice, the 48 wells

in the MPN plate are evaluated or examined upto a maximum of 8 weeks. Thereafter, the wells start drying up and are not usable for evaluation purposes. This work uses the photographic images of these culture samples as the data set for training machine learning algorithms.

2.2 Machine Learning approaches for TB diagnosis

Increased availability of General Processing Unit (GPU) based processing capability at affordable price points and the availability of patient data sets has made it possible to use machine learning for the diagnosis of TB. Detection or diagnosis of Tuberculosis from patient data has been attempted using a variety of techniques, ranging from conventional machine learing approaches to Deep Learning methods such as Convolutional Neural Networks (CNN). This section describes some of the work that has been done in this space.

The data sets used for an ML based approach ultimately depend on the type of TB detection test that is being conducted.

The most commonly used ML approaches explored in this area appear to be based on use of images of chest radiographs or images of body fluid samples taken under a microscope. In these techniques, a CNN is the most common approach for both feature extraction and culture classification. The related work described below are a representative view of different kinds of data sets in use by ML based approaches to aid TB diagnosis.

2.2.1 Use of chest radiographs

In the work referred to in (Unger *et al.* 2017), a CNN was used to extract features from the radiograph and an SVM is used to perform the classification task. This was based on the idea that CNNs are good at learning features, while SVMs are good at finding decision surfaces. Also, lesser data is needed to train a SVM than to train or fine-tune a whole CNN.

2.2.2 Photographic images of sputum samples taken under a microscope

In this approach, the sputum sample is colorized, placed under a microscope and a photograph taken through the microscope. The feature extraction technique here uses a patch of the image (e.g. 32*32 pixels) with a stride length of 16. Each patch is classified as Positive or Negative. Data augmentation techniques used include rotation, mirroring, and position shifting to eventually obtain a training set of over 3 million samples. Classification is done using a CNN model based on Google's CIFAR-10 model which is designed to handle micro-images. Results indicate a high level of classification accuracy into two categories, Positive and Negative. With the diagnosis made by pathologists jointly via microscope and digital slides as the reference standard for classification accuracy, the sensitivity and specificity of TB detection using this particular ML application (called TB-AI) was 97.94% and 83.65% respectively. These results were obtained after increasing the precision of the digital slides resulting in an improvment of the image quality of the microscopic samples used as test data for the classification. Contaminated samples were found to introduce errors in the classification, although no separate class

for Contamimation was labelled which could have allowed the CNN to learn the features of contaminated patches (Xiong *et al.* 2018).

2.2.3 Photographic images of sputum samples as seen through the naked eye

In comparision to the above methods, there appears to be far less work done, if any, on using ML approaches to classify cultures of sputum samples. A preliminary literature review done during the course of this work pointed to ML based classification on other types of diagnostic images but none for sputum samples.

This highlights a significant opportunity for ML to be used for this particular diagnostic work, given that the duration of evaluation of sputum samples (8 to 12 weeks), thereby requiring a human analyst to visually inspect the culture growth in the sample at periodic intervals each week. Clearly, a significant amount of human labour can be saved by developing ML based techniques to replace or augment this currently manual task of visual inspection and classification.

Chapter 3

Methodology

3.1 System Design and Expected Benefits

3.1.1 Key System Components

The three primary components of the proposed system are shown in Figure 3.1. This work focused on the second and perhaps the most significant aspect of the system which is to use machine learning techniques to process and classify the image accurately.

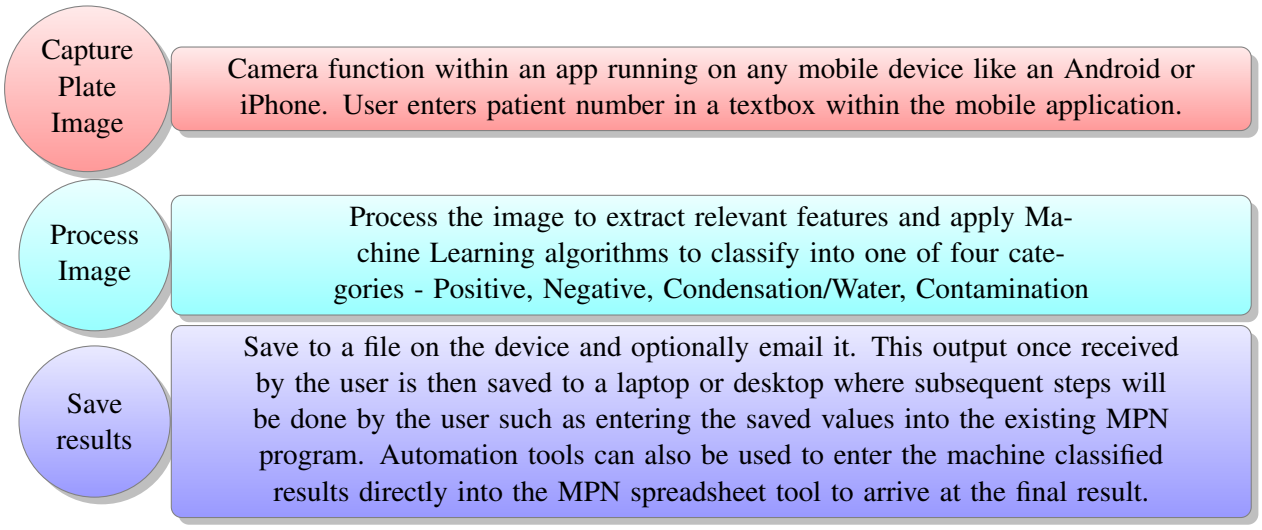


Figure 3.1: Proposed process flow

3.1.2 Expected benefits of the system

1. Avoid variance and errors in human measurements. According to NHLS researchers, different people analyzing the same plate have been seen to arrive at different results.
2. The task of image classification and counting is mundane. The scarce time of researchers can be used productively elsewhere. Based on interviews with NHLS staff, approximately 50 patient sputum samples are received per week. At 2 plate images per patient

sputum sample and 5 minutes to process each patient sputum sample (scoring, writing down) and to populate a Microsoft Excel result table, the time taken is 50 * 5 = 250 minutes per week = 4 hours per week as a representative measure of effort saved as a result of machine learning based automation.

3. Design a scalable process that does not increase human effort linearly as the number of patient sputum samples increases.

3.2 Data Preparation and Feature Extraction

A modular and repeatable data flow pipeline has been developed that allows the training data features to be used across different types of machine learning models. This allows for significant reuse as the data collection and feature extraction steps which form the bulk of a machine learning project can be easily substituted into a new type of ML model that is different from the one used in this particular project.

The data flow and machine learning model pipeline developed for this project is shown in Figure 3.2. Subsequent sections will describe the specific strategies adopted for each stage of the pipeline, key observations and results in each stage and the interfacing mechanism between consecutive stages:

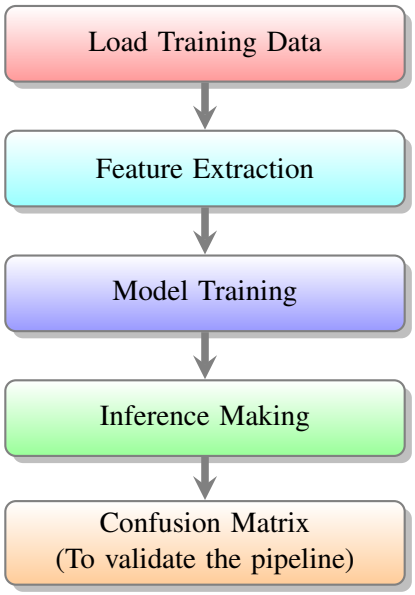


Figure 3.2: Data flow pipeline developed

The Load Training Data and Feature Extraction steps are a one-time activity. The features extracted and stored in this step can be used in any of the machine learning models that are available. For the purposes of this project, the Gaussian Mixture Model (GMM) and Random Forest classifiers were used as will be outlined in the following sections.

3.3 Each well as a training sample

A closer look at each well is helpful to discern image characteristics of the well that can be used for the classification. Figure 3.3 shows two positive wells. Note the cloudy coloration in different parts of each well.

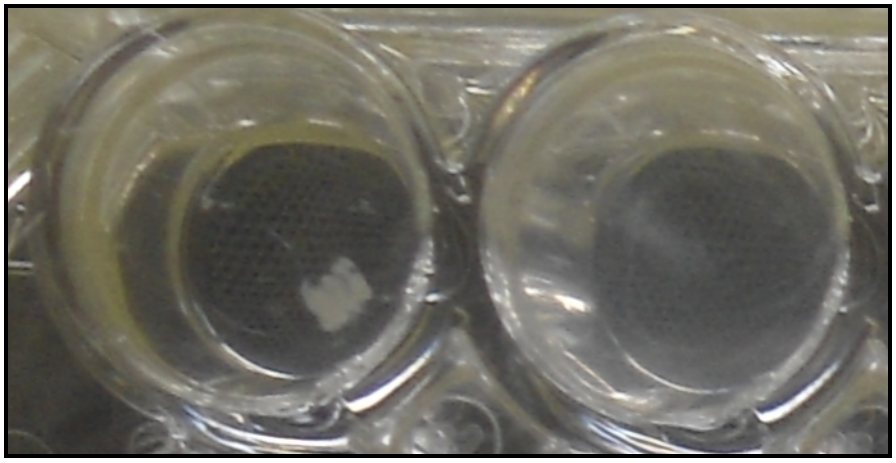


Figure 3.3: An enlarged view of two adjacent positive wells. Note the cloudy coloration to varying degrees within each well

Figure 3.4 shows two negative wells. Note the clear liquid within both wells with no sign of cloudy formation or coloration. There are two small regions within each well where light is reflected from the surface of the liquid. This often leads to the negative wells being misclassified as will be discussed in the results section.



Figure 3.4: An enlarged view of two adjacent negative wells. No cloudy formation or coloration is visible. Notice the near symmetrical reflection of light visible in the left and right parts of the upper region in both wells

Figure 3.5 shows two water wells. Note the presence of water bubbles in both wells. While the water bubbles in both wells are easily visible, often water bubbles are very small in size making it difficult to discern their presence. Also note the well labels D4 and D5 which are visible on the upper left of both wells. These labels are used to identify the well within the

assay plate as well as within the spreadsheet containing the class labels for each well, as will be discussed in the next section.

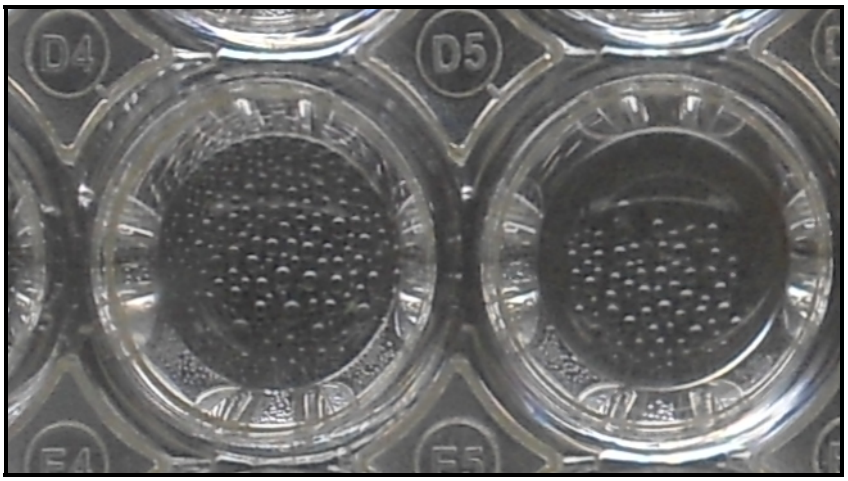


Figure 3.5: An enlarged view of two adjacent wells that have water bubbles or condensation in them. Note the varying sizes of the water bubbles in each well

Figure 3.6 shows two contaminated wells. Note the different kinds of coloration that indicates contamination within a well. The well on the left has a light orange coloration and the one on the right is much darker, almost brown in color, indicating severe contamination of the well.

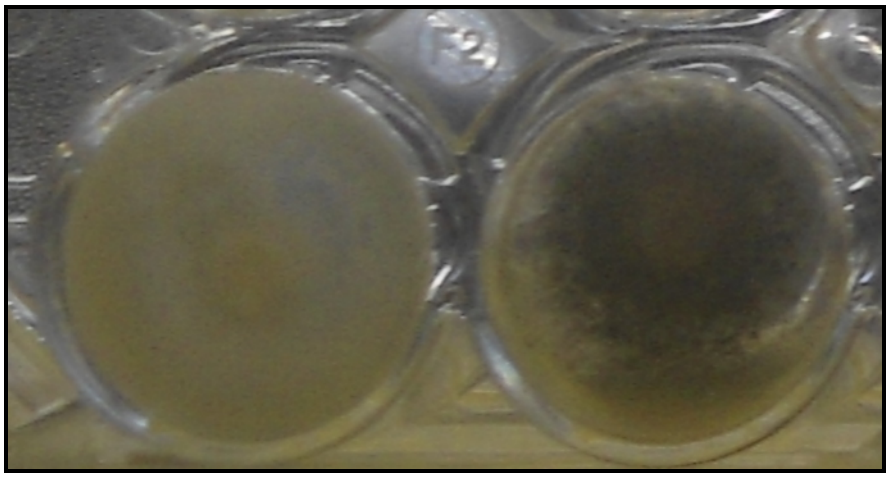


Figure 3.6: An enlarged view of two adjacent contaminated wells. Note the different kinds of coloration that indicate contamination

3.4 Obtaining, Processing and Loading Training Data

A total of 659 images of TB sputum assay plates or trays were obtained from the NHLS. Of these, a small sample of images (25 images) were chosen as Training Data. Emphasis was laid on selecting a representative set of training images that provided samples for each of the 4 classes of interest described earlier. It must be noted that each image provided 48 training

samples as each image has 6 rows and 8 columns of samples (or wells) therefore 6 rows * 8 columns contains 48 well samples. So the total training data size was 25 images * 48 wells per image = 1200 labelled well samples. In practice though, a much smaller sized training set was used in order to speed up the training time on a developer laptop. The tradeoffs as a result of this are discussed in this work.

The training label of each well sample within an image was defined in an accompanying spreadsheet as shown in Figure 3.7. The first row and column of the spreadsheet indicates the row (labelled A though F) and column (labelled 1 through 8) respectively, which is then used to identify the well within the assay tray.



	A	B	C	D	E	F	G	H	I
1		1	2	3	4	5	6	7	8
2	A	N	N	N	N	N	N	N	N
3	B	N	N	N	N	N	N	N	N
4	C	N	N	N	N	N	N	N	N
5	D	W	W	N	N	N	N	N	N
6	E	W	W	W	W	N	N	N	N
7	F	N	W	W	W	N	N	N	N
8									
9									
10									
11									
12									

(a) Training image: DSC05817.JPG

(b) Corresponding Training labels in an xls

Figure 3.7: The 48 wells within a training image are shown on the left. Class labels (training) for each well are provided in the spreadsheet shown on the right

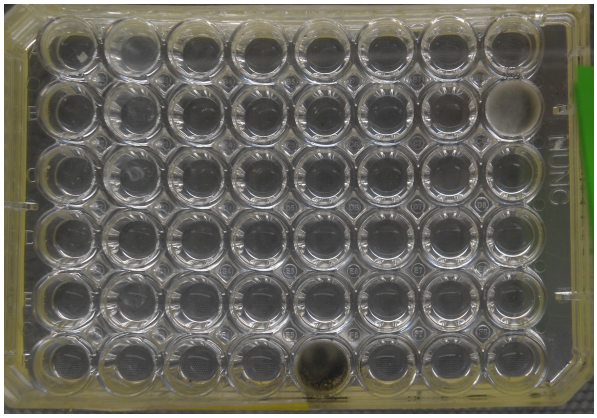
The labels for each well sample within the spreadsheet are one of - 'P' for Positive wells, 'N' for Negative wells, 'W' for wells which have water bubbles in them (indicating condensation) and 'C' for wells that have been contaminated (as indicated by an orange or black coloration within the well).

Notice the degree of similarity between the 'W' labelled wells (Water or Condensation wells) and 'N ' labelled wells (Negative wells). This close similarity will be an interesting source of classification errors as we will discuss further in the results section. For the purposes of illustration, in Figure 3.7, on the plate image shown on the left of the figure, well (E, 1) is labelled 'W' due to water bubbles within the well and the well directly below it, (F, 1) is labelled 'N' as it is Negative (no evidence of bacterial growth, water bubbles or contamination).

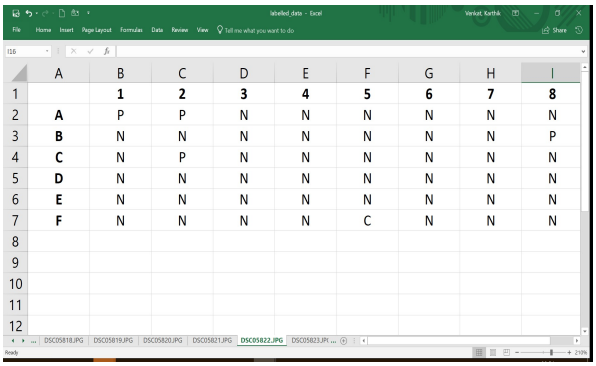
Figure 3.8 shows an assay tray with Positive, Negative and Contaminated wells. The Positive samples although identifiable by a small degree of cloudy formations, can be mistaken for Negative unless closely examined. Well number (B, 8), on row B and column 8, is labelled Positive, but can also be mistaken for Water or Condensation, if not looked at closely.

3.5 Identifying regions of interest and Feature Extraction

The first task in Feature Extraction was to identify regions of interest within each image. Clearly, only the wells themselves within each culture image are of interest in the classifi-



(a) Training image: DSC05822.JPG



	A	B	C	D	E	F	G	H	I
1		1	2	3	4	5	6	7	8
2	A	P	P	N	N	N	N	N	N
3	B	N	N	N	N	N	N	N	P
4	C	N	P	N	N	N	N	N	N
5	D	N	N	N	N	N	N	N	N
6	E	N	N	N	N	N	N	N	N
7	F	N	N	N	N	C	N	N	N
8									
9									
10									
11									
12									

(b) Corresponding sheet of labelled data for this training image

Figure 3.8: Another training pair with Positive, Negative and Contaminated samples - Look for any visible distinguishing features between Positive and Negative wells

cation task. Each well is a circular or near circular region within the image of the assay tray plate containing 48 wells. That part of the assay tray plate image which falls outside of any well is not of interest for the learning and classification task. Therefore, the first task in the Feature Extraction task was to find a way by which all the 48 wells within each image could be identified.

3.5.1 Well detection with Hough Circles

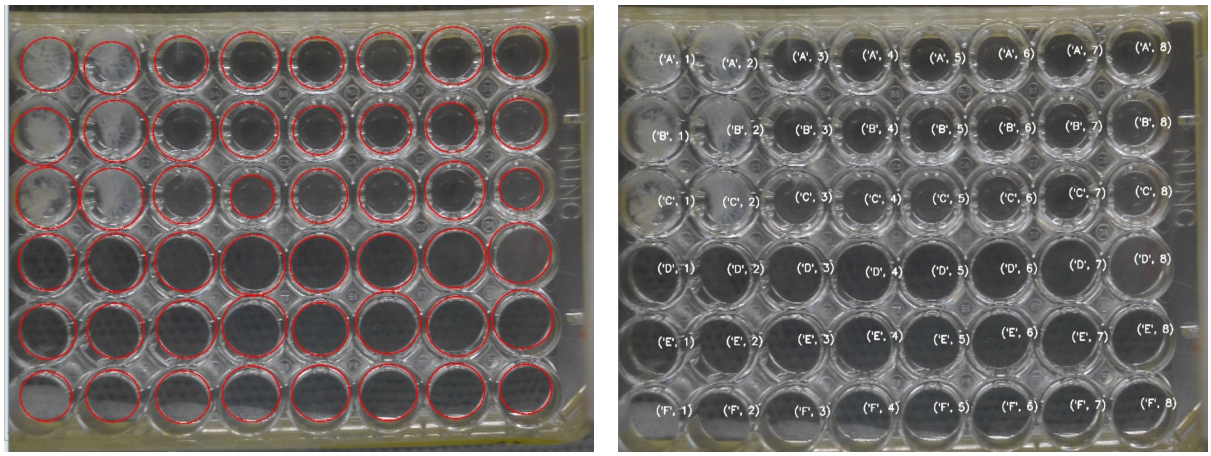
The Hough Circle Detection algorithm was used to detect the shapes of the wells. A heuristic approach was used to find the right combination of parameters (e.g. number of circles representing the wells and radius of each circle) for the OpenCV Hough Circle function to detect the circles representing the wells. Each of the 48 wells were then programmatically provided a name in a (row, column) format representing the row and the column within the training image that the well belonged to. This identifier for each well was then used to uniquely identify each well within all the training images.

Figure 3.9 shows an assay tray plate with each of the 48 wells within the image having first been identified with an enclosing red circle around each well. Figure 3.9a shows the enclosing circles around each well and Figure 3.9b depicts an identifying label in (row, column) format overlaid on top of each well. These are two important steps in identifying all the wells within an image.

3.5.2 Extracting Pixels within the Well using Masking

A masking appoach was used to extract the pixels within the well. After identifying each of the wells using Hough Circles, the wells were drawn in White over a Black background representing the plate image. This resulted in 48 white wells drawn over a black background. This is shown in the Figure 3.10a below.

The mask was then used to read the pixel values within each of the outlined circles from the original image. Once a mask was used, the pixels within the well could be identified.



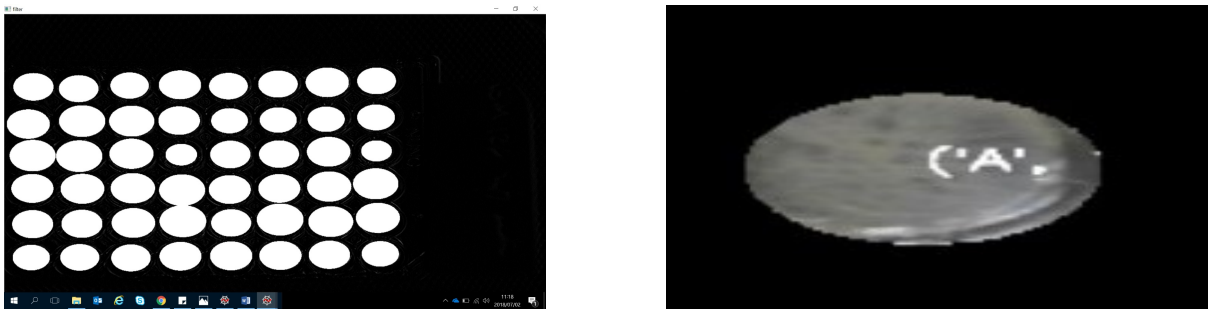
(a) Enclosing circles on each of the 48 wells

(b) Each well identified with a (row, column) format

Figure 3.9: The OpenCV HoughCircles algorithm was used to identify and name the wells within each training image

Obviously, separate masks are created for each plate image. Using a mask then allowed the easy reading of each pixel value within the well using the code shown in Listing B.1 resulting in the pixel values being obtained for each well.

See Figure 3.10b where the pixel values for the well ('A', 1) has been isolated.



(a) Masks for each well in the image

(b) Using the mask to extract pixel data for well (A,1)

Figure 3.10: Use of Masks to obtain pixel values from grayscale representations of a well in an image of an assay tray or plate

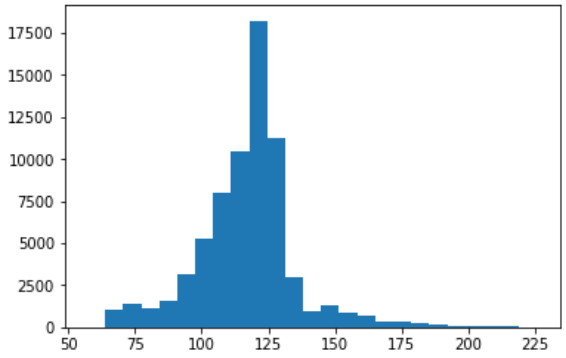
3.5.3 Feature Extraction and Reduction using Histogram based approaches

There are approximately 70,000 pixels within the circular region of each well. Each pixel contains the grayscale value (between 0 and 255) and each such pixel represents a unique feature. Initial attempts in using the Expectation Maximization (EM) algorithm to model the underlying stochastic process in the distribution of each individual pixel value using a GMM model took an inordinately long time to execute on a personal computing device. Clearly this approach was untenable for off the shelf hardware devices. Hence, a feature reduction technique was adopted by using a histogram of pixel values within each well, instead of each

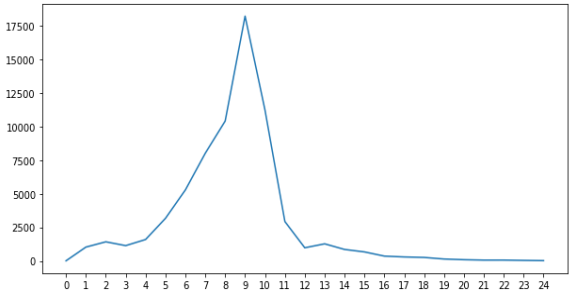
individual pixel value. This reduced the number of features from 70,000 (approximate number of pixels in each well) to 25 (the number of bins in the histogram). Each pixel was allocated to one of the bins based on it's grayscale pixel value which ranged from 0 to 255. The bins counts (number of pixels in each bin) were then used as features.

Histograms of the pixel values were then analysed and the following trends in the grayscale values of the pixels were observed. The hypothesis here was that Positive classes will generally exhibit a higher proportion of lighter colour pixels, due to the cloudy formations in the wells, as compared to the other classes. The histograms were initially designed with 25 bins and are shown from Figures 3.11 to 3.16.

Figures 3.11 and 3.12 show the histograms of pixel grayscale values for two different positive wells. Note the high proportion of grayscale values between the pixel values of 100 and 150 (the nominal range of grayscale pixel is between 0 and 255) in both these figures

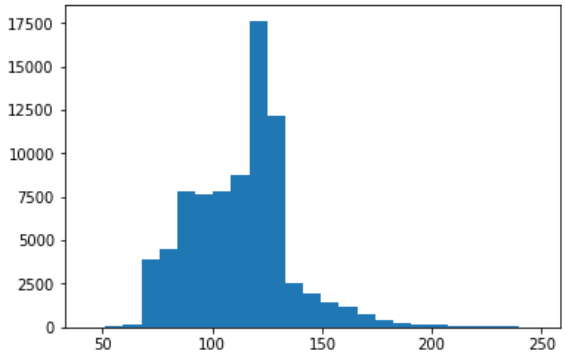


(a) Histogram representation

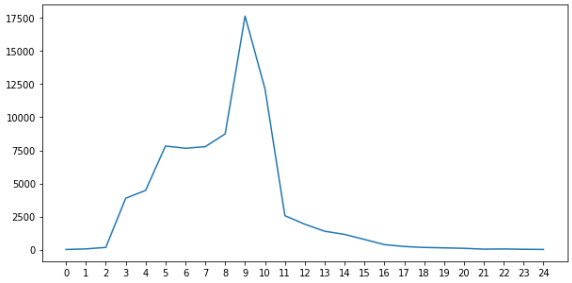


(b) Line graph representation

Figure 3.11: 25 Bin Histogram of Grayscale Values for a Positive well



(a) Histogram representation

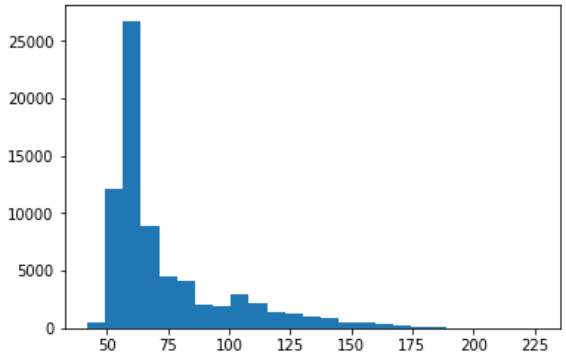


(b) Line graph representation

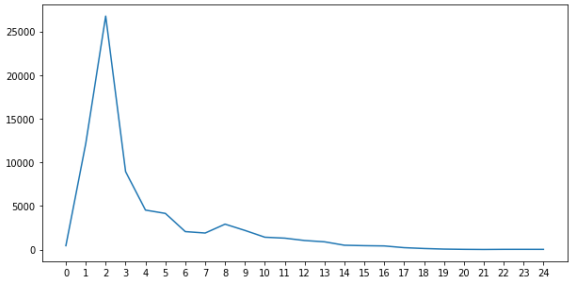
Figure 3.12: 25 Bin Histogram of Grayscale Values for another Positive well

Figures 3.13 and 3.14 show the histograms of pixel grayscale values for two different negative wells. Note the high proportion of grayscale values between the pixel values of 50 and 70 in both these figures

Figure 3.15 shows the histogram of pixel grayscale values for a well with water bubbles or signs of condensation. Note the close similarity of it's pixel value distribution with that

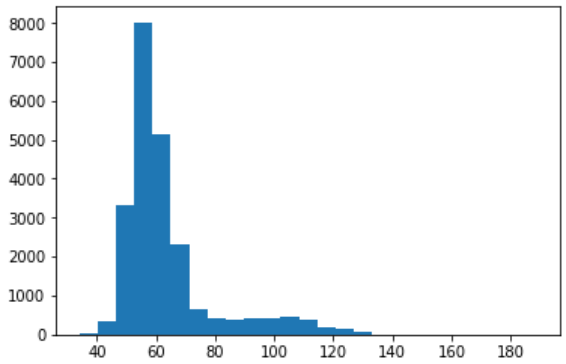


(a) Histogram representation

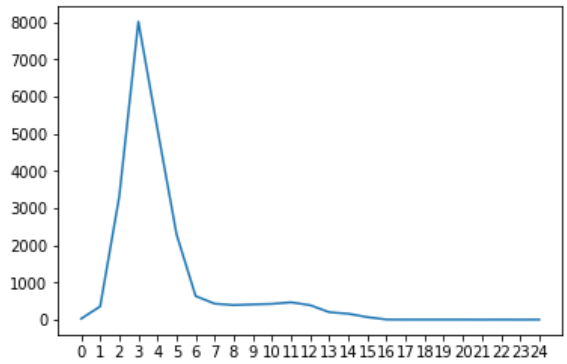


(b) Line graph representation

Figure 3.13: 25 Bin Histogram of Grayscale Values for a Negative well



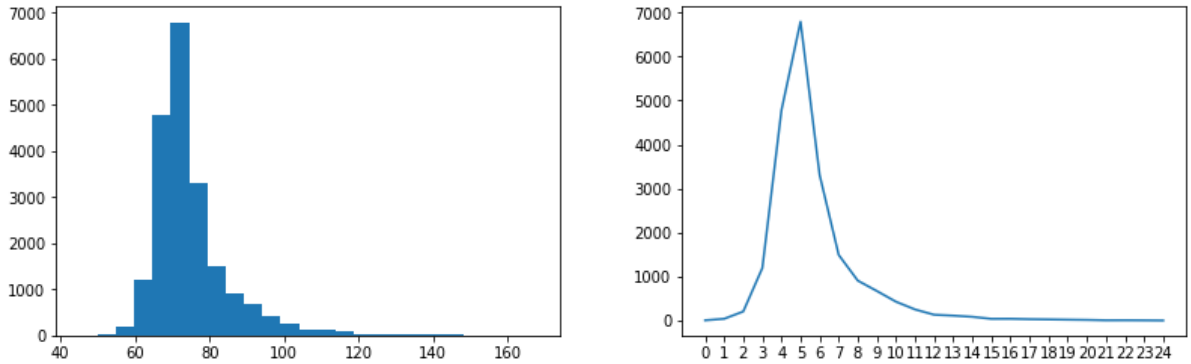
(a) Histogram representation



(b) Line graph representation

Figure 3.14: 25 Bin Histogram of Grayscale Values for another Negative well

of a negative well, which make the class of water wells particularly difficult to separate from negative wells as will be discussed in the results section. Even so, it can be seen that the water well has most of it's pixel grayscale values above a threshold value of 60 while both the negative wells in Figures 3.13 and 3.14 have a number of pixels with a grayscale value between 50 and 60. This difference, if observed, across several images of both classes could provide a useful means to discriminate between the Water and Negative class.

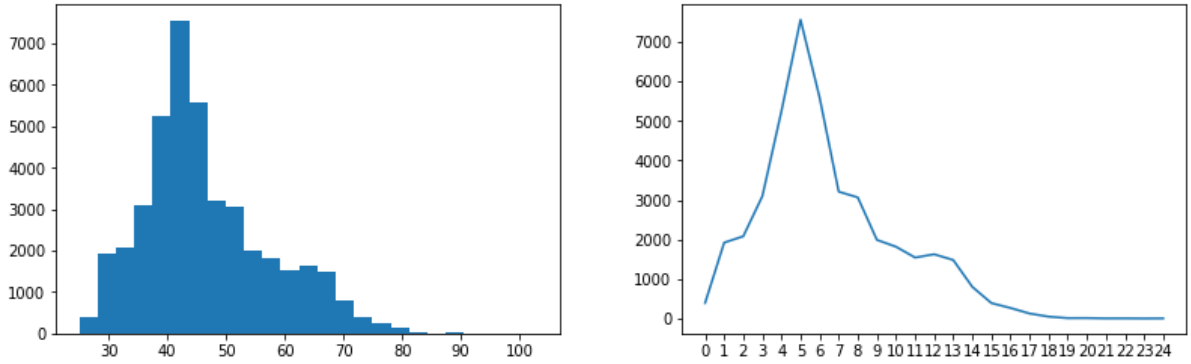


(a) Histogram representation

(b) Line graph representation

Figure 3.15: 25 Bin Histogram of Grayscale Values for a Water well

Figure 3.16 show the histogram of pixel grayscale values for a well with signs of contamination. Note the large number pixel values between 30 and 60 indicating a number of dark or nearly black pixels in the contaminated well.



(a) Histogram representation

(b) Line graph representation

Figure 3.16: 25 Bin Histogram of Grayscale Values for a Contaminated well

3.5.4 Feature Engineering using LBP to analyze textures of each well

Image textures provide information about the spatial arrangement of color or pixel intensities in an image or selected areas of an image. In this project, it was hypothesized that image textures could provide a useful feature to use in the learning and classification process.

A Local Binary Pattern (LBP) descriptor was used to analyze the texture within the different class types. The LBP operator returns a discrete value at each pixel that characterizes the local texture in a way that is partially invariant to luminance changes.

The basic LBP operator compares the eight neighboring pixel intensities to the center pixel intensity, assigning a 0 or a 1 to each neighbor depending on whether they are less than or greater than the center value. The neighboring 24 pixels around every pixel were examined around each point within a well as shown in the Code Listing B.2

Any analysis of the results of the LBP histograms was done to highlight the Edges, Flat surfaces and Corners of the images, with the hypothesis that a Positive well will have more edges and corners due to the cloudy formations while Negative images will have a higher proportion of flat surfaces.

Figures 3.17 and 3.19 indicate the texture characteristics of positive wells and Figure 3.18 shows a line graph representation of the LBP histogram that is used as one of the features to train an ML program.

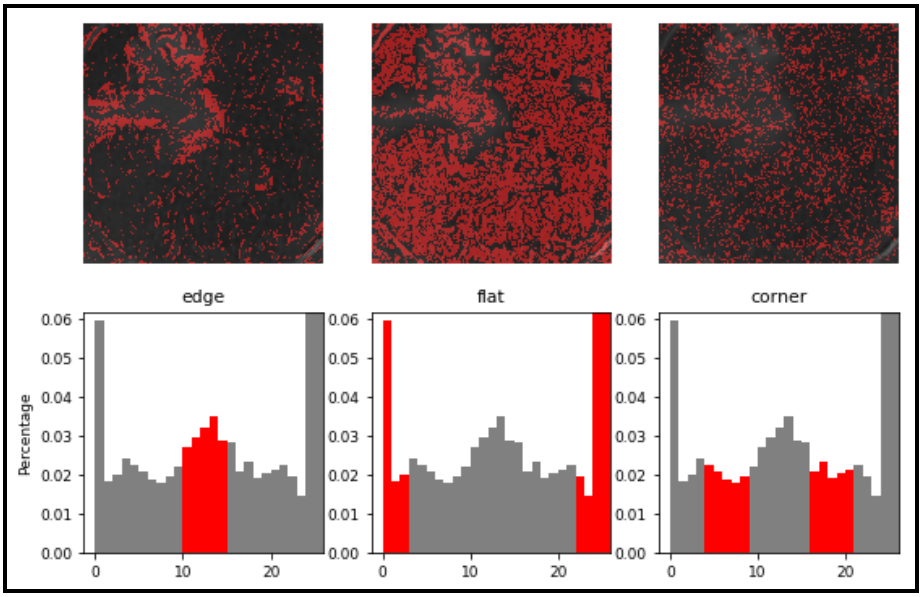


Figure 3.17: LBP for a Positive well sample: Proportion of Edges and Corners are expected to be higher than a Negative sample. Notice the presence of edges due to cloudy formations on a positive sample.

Figures 3.20 and 3.21 indicate the texture characteristics of negative well samples.

Figure 3.22 shows the texture characteristics of a water well sample. The analysis of each well texture is provided with the description for each figure.

While the LBP does seem to provide useful features that can be used to train the model, it appears to have the following limitations:

1. The Crop region for each well was a rectangle and not a circle, therefore the outline of the well boundaries or borders were also included in the LBP region. This can be seen in Figures 3.24 and 3.25 where the outline of the well border is visible. This introduces avoidable noise in the LBP analysis as well boundaries do not provide any additional

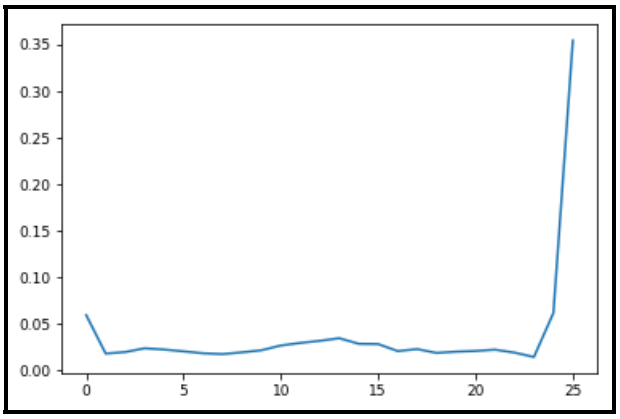


Figure 3.18: LBP Histogram for a Positive well sample: Used as a Feature for training the model

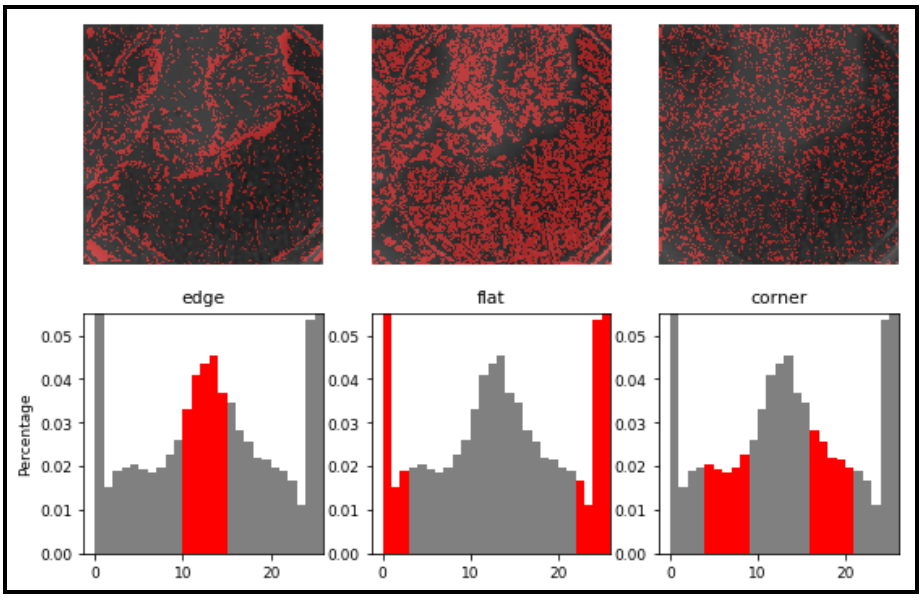


Figure 3.19: LBP for another Positive well sample: Proportion of Edges and Corners are expected to be higher than a negative sample. Notice the presence of edges due to cloudy formations on a positive sample.

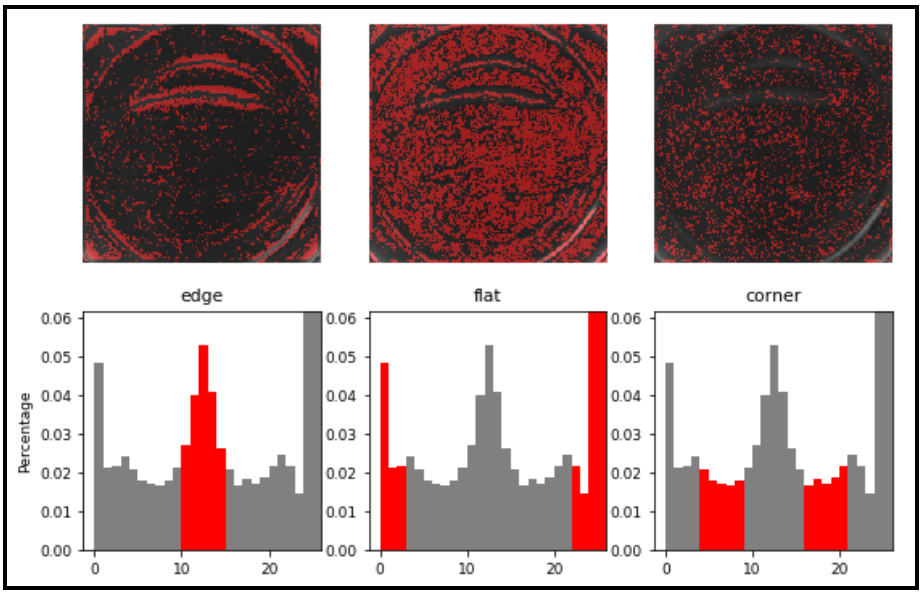


Figure 3.20: LBP for a Negative well sample: Proportion of Flat regions are expected to be higher than a positive sample. Notice the lack of edges. The crescent shaped edges are due to reflection of light on this particular negative image sample. Reflection of light can be mistaken for a texture difference between wells

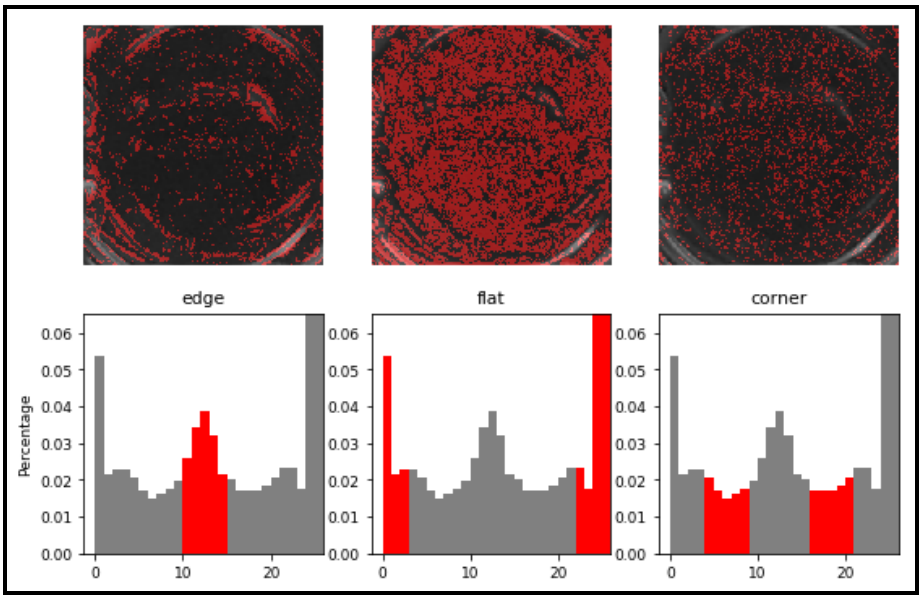


Figure 3.21: LBP for another Negative well sample: Proportion of Flat regions are expected to be higher than a positive sample. Notice the absence of significant edge features

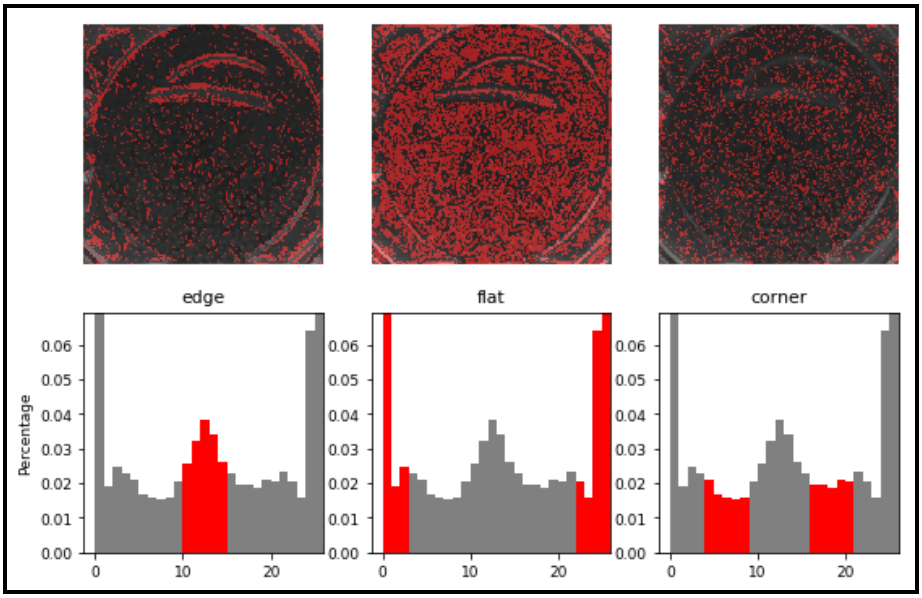


Figure 3.22: LBP for a Condensation or Water well sample. Notice the close similarity with the texture of Negative samples. A factor which we will discuss in the results

information to aid the classification task as they occur across all the 4 classes of wells. While this is due to the choice of the image processing approach used to extract the well region, it is a not a limitation of the LBP approach itself. Future image processing techniques could focus on LBP analysis of the bounding circle region itself.

2. There does not seem to be any noticeable texture difference between Negative and Condensation samples. The presence of a water bubbles on what is inherently a negative well does not seen to yield any noticeable difference in textures. This could explain some of the eventual results seen in classifying a well as a Negative or Water (Condensation)

3.5.5 Limitations in the use of contours to extract region of interest for Texture analysis

This section describes the use of contours to extract the region of interest, specifically the area within a well for subsequent LBP analysis. While the Hough Circle approach described earlier allowed for all pixels within the area of interest of a well to be extracted, texture analysis requires not just the pixel values but the location information to be preserved, in order for LBP to infer textures based on difference in intensities of neighboring pixel for every pixel.

The approach used for this for to identify the enclosing rectangular contours for every well image as can be seen in Code Listing B.3. As can be seen in Figures 3.24 and 3.25, this causes additional, non-significant areas of the image to be included for LBP analysis as well, such as the rim or well borders. This is expected to affect the classification accuracy. As can be seen in Figure 3.25, both enclosing rectangles and enclosing circles were attempted which includes all the contour points within the image. This resulted in the inclusion of extraneous regions within the region of interest such as well borders that

adversely impacts the classification accuracy.

To overcome this, it is suggested to use an approach that does not depend on contour analysis, but also preserves the shape and location information of every pixel. An image that does not include well borders is shown in Figure 3.23. This technique is already used for the histogram feature extraction discussed earlier and the same cropped region can be used for LBP as well, without necessitating a bounding rectangle using contour analysis. This is identified as an area for further work.

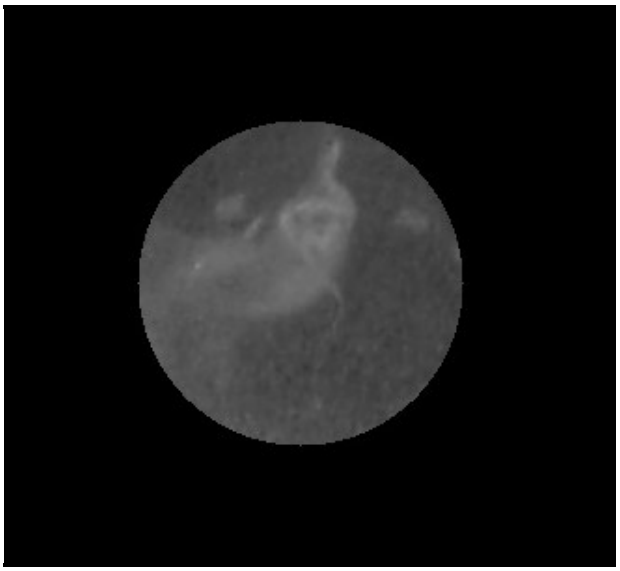


Figure 3.23: Circular region of a well extracted using masking and Hough Circle Technique for shape detection

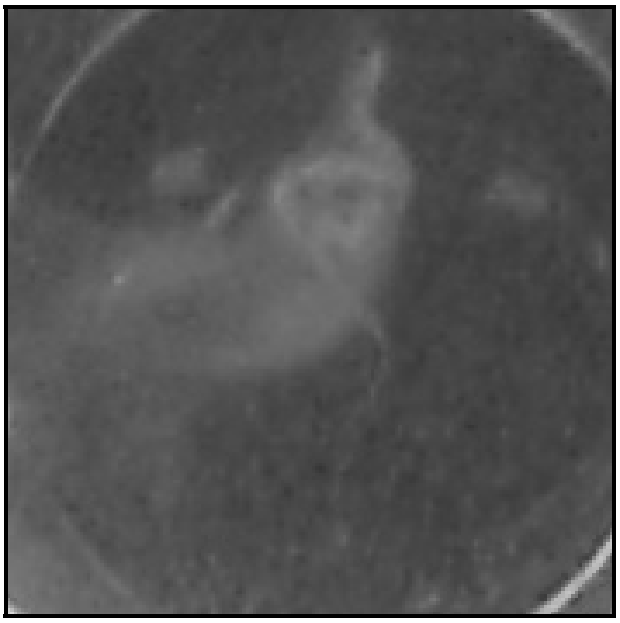


Figure 3.24: Bounding rectangular region extracted with contour analysis and used for LBP. Notice presence of well borders which can affect the LBP analysis

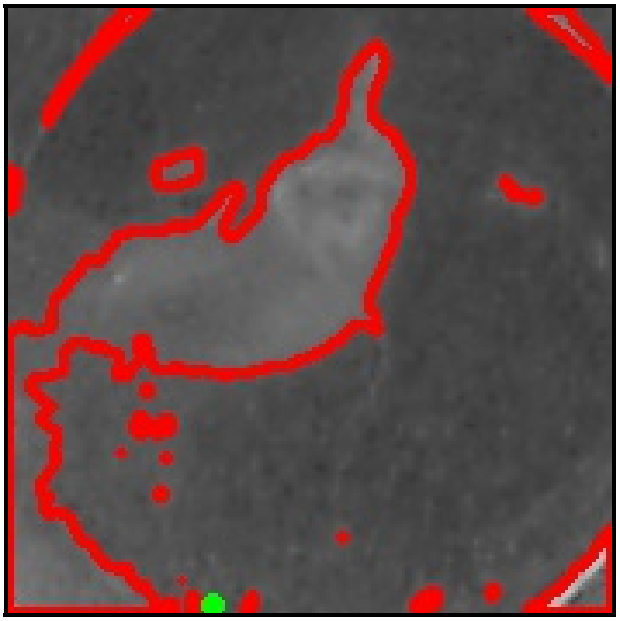


Figure 3.25: Other contours are visible as well. The image was thresholded using a pixel value of 90 to identify contours of light coloured areas or cloudy sections of the pixels. Notice presence of well borders

3.5.6 Feature Engineering Using HSV values of each well

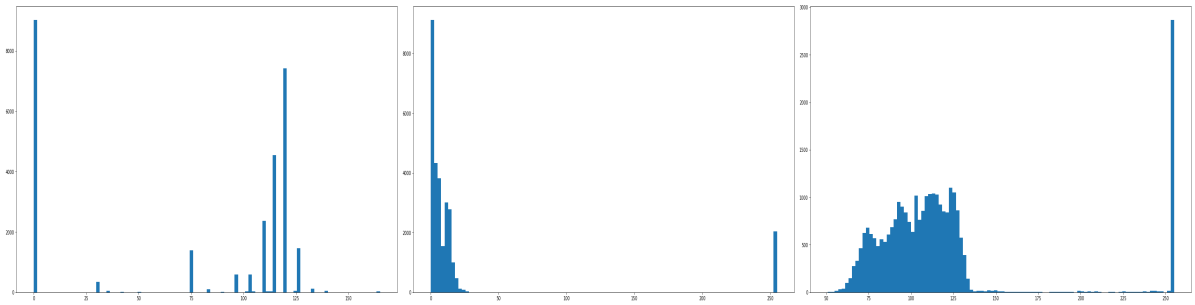
In this approach, the HSV (hue, saturation, value) color model was used to analyse the different image classes and extract useful features that could be used in the image classification task.

It is often more effective to represent a color in HSV than in BGR color space. The characteristics generally used to distinguish one color from another are Brightness, Hue and Saturation.

Hue is an attribute associated with the dominant wavelength in a mixture of light waves. Hue represents dominant color as perceived by an observer. Saturation refers to the relative purity or the amount of white light mixed with the hue. The degree of saturation is inversely proportional to the amount of white light added (Gonzalez and Woods 2016).

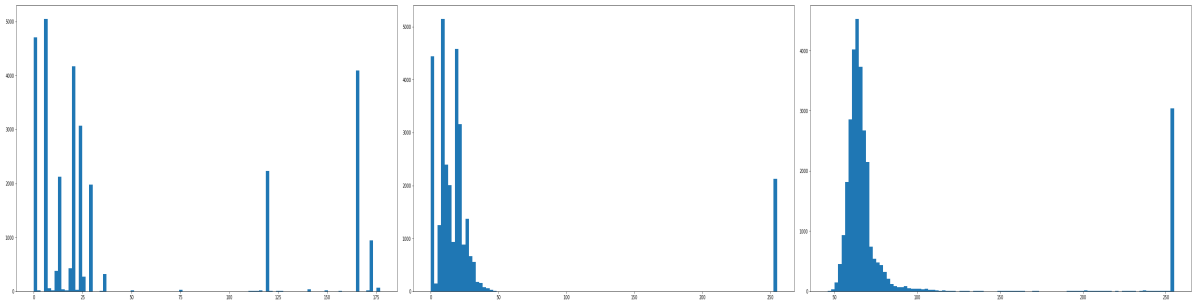
As an example, the histogram of HSV values for a Positive and Negative class are shown in Figures 3.26 and 3.27. As can be seen in a visual comparison of the HSV histograms of both classes, there does appear to be distinct differences in HSV across both class types. The Hue histogram for positive class is shifted to the range of higher Hue values (to the right of the x-axis) than that of the Negative class which has higher occurrences in the lower range of Hue values (to the left of the x-axis). The histograms of Saturations values, on the other hand, exhibit no signficant differences. The histograms of the Values (V) seem to mirror the differences observed between the grayscale histograms of the Positive and Negative classes.

For the purposes of model training, the V (Values) were selected and the testing results discussed later in this document are based on the V values. However, the Hue and Saturation histograms could also be used to train the model. This comparative analysis and contribution of H, S and V values to the model effectiveness is an area identified for future work.



(a) Histogram of Hues (b) Histogram of Saturations (c) Histogram of Values

Figure 3.26: 100 Bin Histogram of HSV Values for a Positive sample



(a) [Histogram of Hues (b) Histogram of Saturations (c) Histogram of Values

Figure 3.27: 100 Bin Histogram of HSV Values for a Negative sample

3.6 Learning, Inferencing and Validation with Machine Learning Models

Two supervised learning algorithms for the classification task were chosen for the classification task:

1. Gaussian Mixture Model (GMM) Probabilistic Classifier - A parametric classification method
2. Random Forest Classifier - A non-parametric classification method

The use of these classification models are described in the following sections

3.7 Gaussian Mixture Model (GMM) Classifier

A Gaussian mixture model (GMM), also referred to as a Mixture of Gaussian (MoG) attempts to find a mixture of multi-variate Gaussian probability distributions that best model any input dataset. For the problem at hand, separate GMM models were constructed using the training data for each of the desired classification types, namely:

1. Positive Class
2. Negative Class

3. Condensation Class
4. Contamination Class

The data within each of these classes are described as a weighted sum of K class conditional probability distributions

$$Pr(x|c = \text{'P'}, \hat{\theta}) = \sum_{k=1}^{K_P} \lambda_k Norm_x[\mu_k, \Sigma_k] \text{ for the Positive (P) class of wells.}$$

$$Pr(x|c = \text{'N'}, \hat{\theta}) = \sum_{k=1}^{K_N} \lambda_k Norm_x[\mu_k, \Sigma_k] \text{ for the Negative (N) class of wells.}$$

$$Pr(x|c = \text{'W'}, \hat{\theta}) = \sum_{k=1}^{K_W} \lambda_k Norm_x[\mu_k, \Sigma_k] \text{ for the Water (W) class of wells.}$$

$$Pr(x|c = \text{'C'}, \hat{\theta}) = \sum_{k=1}^{K_C} \lambda_k Norm_x[\mu_k, \Sigma_k] \text{ for the Contaminated (C) class of wells.}$$

where $\mu_{1..K}$ and $\Sigma_{1..K}$ are the means and covariances of the normal distributions representing the features for each of the four class of wells and $\lambda_{1..K}$ are positive valued weights that sum to one. The GMM describes complex multi-modal probability distributions densities by combining simpler constituent distributions.

The GMM function within the scikit-learn package learns the parameters $\hat{\theta} = \{\mu_k, \Sigma_k, \lambda_k\}$ for each value of k from 1 to K from the training data $\{x_i\}$ for each class of well.

K_P, K_N, K_W, K_C denote the value of K for the Positive, Negative, Water and Contamination classes of wells.

3.7.1 Training the Gaussian Mixture Models

A separate GMM model was trained for each of the four different classification types. For example, a GMM model for the Positive class was trained on the well samples of positive cultures only (those with cloudy formation). Similarly the GMM model for the Negative class was trained on well samples of negative cultures. This allowed each GMM model to model the overall distribution of training data in that class.

Different values of K (number of gaussian distributions) for each Mixture of Gaussian per sample were tested to determine the optimum value for K. Determining the optimum value of K per class was key to avoid over-fitting.

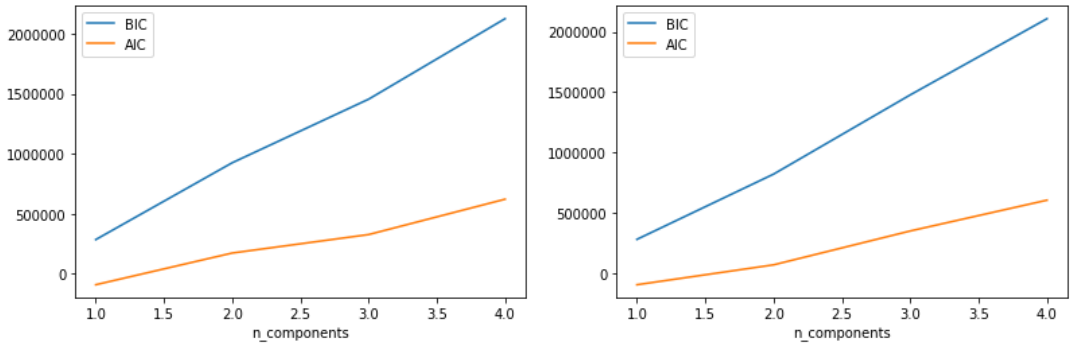
Analytic criteria such as Akaike Information criterion (AIC) or the Bayesian Information Criterion (BIC) was also used to determine the value optimum value of K for each class of well. Scikit-Learn's GMM estimator includes built-in methods that compute both of AIC and BIC.

3.7.2 Using the Akaike information criterion and Bayesian information criterion

The Akaike information criterion (AIC) and Bayesian information criterion (BIC) are two estimators of the relative quality of different statistical models for a given set of data. AIC and BIC are based on information theory and provide a useful criteria for model selection. The AIC and BIC scores take into consideration the trade-offs between the goodness of fit of the model on a given data set, and the simplicity of the model in terms of the number of its parameters, or in terms of the structure of the model. The goodness of fit refers to the ability of the statistical model (GMM in this case) to represent the underlying process that generated the data. The model parameter is the value of K (number of Gaussians in the GMM model for each class).

While AIC rewards the goodness of fit of the model, it also imposes a penalty for increasing the number of model parameters. Therefore, it discourages model overfitting. The BIC works similarly but imposes a higher penalty based on the number of samples used to train the model (i.e. the BIC score is willing to consider more complicated structures if there is 'enough' evidence for one)

Given a set of models that fit the data set, the preferred model is the one that has the minimum AIC or BIC value.



(a) Positive Class

(b) Negative Class

Figure 3.28: AIC and BIC curves for Positive and Negative classes

The optimal value of K is the one that minimizes the AIC or BIC. The AIC and BIC curves in Figure 3.28 indicate that a low value of K (number of gaussian components per class or clusters) is preferable.

It is important to note that this choice of number of components measures how well GMM works as a density estimator, not how well it works as a clustering algorithm. Literature recommends the use of GMM primarily as a density estimator, and to use it for clustering only when warranted within simple datasets (VanderPlas 2016).

This could explain why the AIC and BIC curves above are not particularly useful in this problem which involves complex data sets.

A feature minimization technique like PCA (Principle Components Analysis) could be used to simplify or reduce the data sets before fitting a GMM model to it. This is identified as a future area of exploration.

3.7.3 Training or Fitting the GMM model for each category of well

The fitting of a GMM model to the training data samples for each of the four classes was done using the Scikit-Learn's GMM function as shown in Code Listing B.4. Here you see the GMM model, called clf for classifier, being trained for the positive class training data.

The hyperparameter *covariance_type* controls the degrees of freedom in the shape of each cluster or class. The setting *covariance_type='full'*, allows each cluster or class to be modelled as an ellipse with arbitrary orientation, providing maximum flexibility in the fit. Although computationally expensive, it was possible to train the models on an Intel i7, 1.9 GHz, 16GB RAM Windows machine.

3.7.4 Inferencing from the GMM model for each category of well

After training or fitting each of the 4 GMM models to their respective training data sets, the inferencing process then scores the new sample (from the validation or test data set) to be classified, with each of the 4 models. The model score represents the Log probability of the new sample. This is done with the *score_samples* method as shown in code listing B.5. The model with the maximum log probability for the new data is selected as the class label.

The GMM model also provides an average log-likelihood of the new sample. Given that this is a likelihood measure and not a probability measure, an identified future area of work is to explore how this likelihood measure can be used to improve the classification accuracy.

The code snippet in B.5 shows the scoring process for the new sample using each of the 4 GMM models.

3.8 Random Forest Classifier

Random forests are an example of an ensemble learner built on decision trees. Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

In Python's Scikit-Learn package, such an optimized ensemble of randomized decision trees is implemented in the RandomForestClassifier estimator, which takes care of all the randomization automatically.

A Random Forest with 4 decision trees was used for initial testing. Within the sample data for training, a cross validation set of 80:20 was created i.e. 80% of the sample data was used for training and 20% for validation.

The results with the validation set were encouraging at 76.19%.

K-fold cross validation was used to analyse the hyper-parameters while retaining generalization accuracy. Here, the training set is split into k smaller sets. The following procedure is followed for each of the k "folds":

1. A model is trained using k–1 of the folds as training data

2. The resulting model is validated on the remaining part of the data (i.e., it is used as a validation set to compute a performance measure such as accuracy). With a K-fold value of K=5, Table 3.1 shows the validation accuracy levels obtained for each of the five folds. The results indicate a mean accuracy of 73.87% with a standard deviation of 12.55%

5-Fold Accuracy Score	
Cross Fold No.	Accuracy
1	62.50%
2	77.78%
3	90.32%
4	82.26%
5	58.06%

Table 3.1: Accuracy levels observed for each fold within a 5-fold cross validation

A Validation Curve shown in Figure 3.29 was used to plot the influence of a single hyperparameter on the training score and the validation score to find out whether the estimator is overfitting or underfitting for some hyperparameter values. The hyperparameter in this case is the number of trees in the random forest.

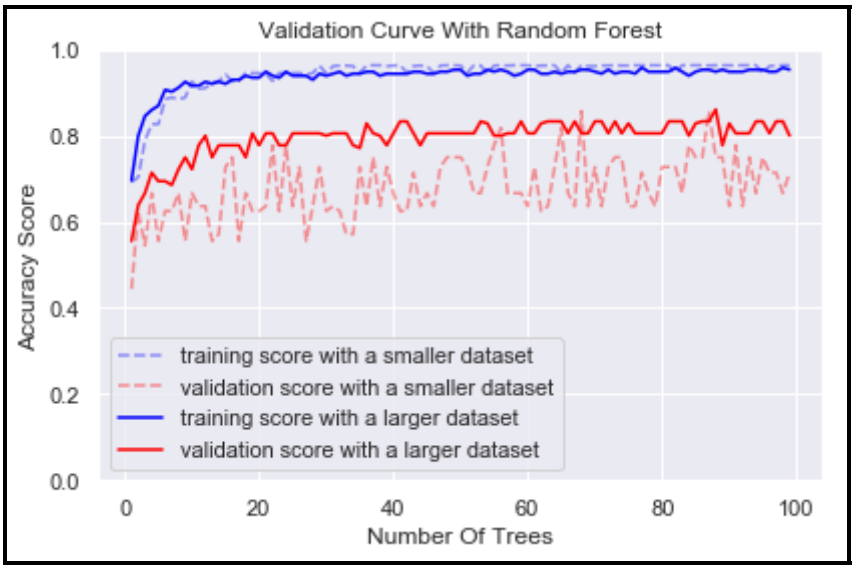


Figure 3.29: A learning curve with training and validation accuracy scores on two datasets. The solid lines show the results with a larger data set (80% of sample data), while the fainter dashed lines show the results with a smaller data set (20% of sample data).

It must be noted that although the well samples are different between the Training and Validation sets, they all come from the same set of images. This could possibly explain the high validation accuracy which was not necessarily the case when the model was run on the test data which came from a completely different set of images.

Table 3.2 lists the number of training samples within each category that were available and the numbers that was actually utilized for training purposes.

Number of training samples used for the different classes		
Class Label	Number of total training samples available	Number of training samples utilized
Positive	158	100
Negative	929	100
Condensation	97	97
Contamination	16	16
Total	1200	313

Table 3.2: Number of training samples

Unless other specified in the results section that follows, the number of samples used for training purposes are 100 or less per sample, as shown in the rightmost column of Table 3.2. This was motivated by the need to keep training times short (typically within 2-3 minutes) given the constraints of human time available. The implications of this constraint and future work possible is discussed in the following sections.

Chapter 4

Results and Discussion

4.1 Classification results on a Test data set

The results discussed in this section were obtained in a test data set comprising of 4 test images which were neither present in the training data set nor the validation data set. Therefore, these were completely new or unseen images for the trained classifiers. These 4 test images provided a total of 4 * 48 = 192 wells, for the trained models predicting a probability of the label for each well. These predicted labels are discussed in this section.

Results are discussed with the aid of a Confusion Matrix as well as the corresponding Cohen Kappa score. Cohen's kappa coefficient is a statistic which measures inter-rater agreement for qualitative (categorical) items. It is generally thought to be a more robust measure than simple percent agreement calculation, as it takes into account the possibility of the agreement occurring by chance as well as changes due to imbalanced data sets.

4.2 Results with GMM

Initial results with the GMM gave the following results:

1. **Equal values of K (number of Gaussians) for each of the 4 well classes.**

   In this test, the K value was set to be the same for each well class as shown in Table 4.1

K Values	
GMM for Positive class of wells	K_P = 8
GMM for Negative class of wells	K_N = 8
GMM for Water/Condensation class of wells	K_W = 8
GMM for Contamination class of wells	K_C = 8

Table 4.1: Equal number of Gaussians per category

   As can be seen in the results in the confusion matrix in Figure 4.1, the accuracy within the Positive samples was high. Within the Negative class, there were a high number of

false positives with 47 of them being classified were classified as Positive. Interestingly, within the Water class too, a significant number of them (8) were classified as Negative.

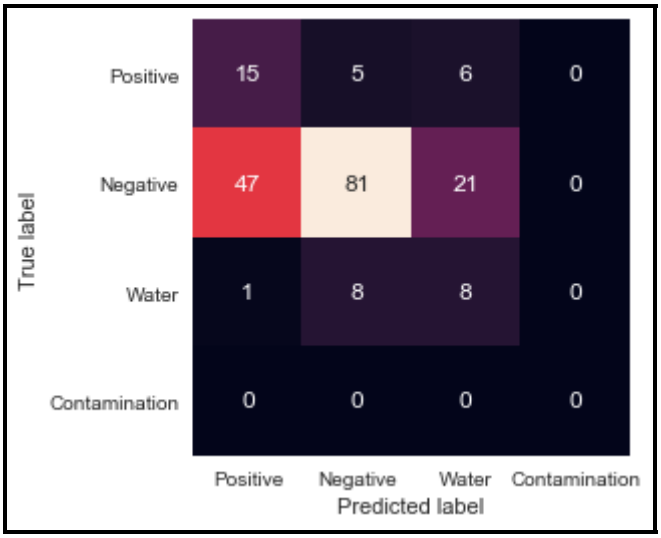


Figure 4.1: Confusion Matrix for GMMs with equal values of K (number of Gaussians) for each of the 4 well categories.

Number of observed agreements (diagonal values in the confusion matrix): 104 (54.17% of the observations)

Number of agreements expected by chance: 84.6 (44.05% of the observations)

Kappa= 0.181

Standard Error (SE) of kappa = 0.052. 95% confidence interval: From 0.079 to 0.283. The strength of agreement is considered to be 'poor'.

2. **Differing values of K (number of Gaussians) for each of the 4 categories.** In this test, different values of K were used for each category as shown in Table 4.2

K Values	
GMM for Positive class of wells	$K_P = 15$
GMM for Negative class of wells	$K_N = 30$
GMM for Water/Condensation class of wells	$K_W = 8$
GMM for Contamination class of wells	$K_C = 3$

Table 4.2: Modified number of Gaussians per category

As can be seen in the confusion matrix in Figure 4.2, there was a very high accuracy in the classification of the Water samples likely due to the ability of the model to generalize better due to a lower value of K compared to the higher values of K for the Positive and Negative classes.

It is interesting to note that higher values of K for the positive GMM did not result in any increase in classification accuracy for the Positive class. The Negative class on the other hand yielded a very high error rate with most of the Negatives being classified as Water.

Therefore a higher value of K for Negative did not yield good results. As noted in the AIC and BIC curves in Figure 3.28, a lower value of K appears to be more suited to the classification task.

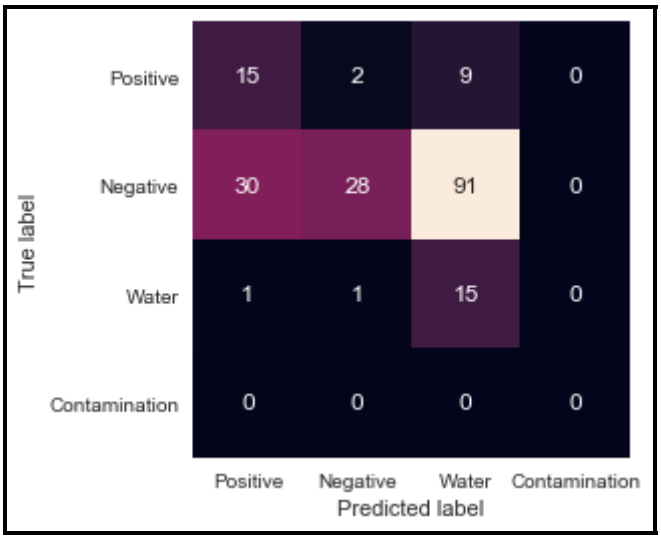


Figure 4.2: Confusion Matrix for GMMs with different values of K (number of Gaussians) for each of the 4 well categories.

Number of observed agreements: 58 (30.21% of the observations)

Number of agreements expected by chance: 40.5 (21.08% of the observations)

Kappa = 0.116.

Standard Error (SE) of kappa = 0.029. 95% confidence interval: From 0.058 to 0.173. The strength of agreement is considered to be 'poor'.

4.3 Results with Random Forest Classifier

1. **Model trained with a smaller training data set**

   In this test, the number of training samples utilized were as shown in the rightmost column in Table 4.2 in the previous section. It comprised of a total of 313 training samples with 100, 100, 97 and 16 well samples for Positive, Negative, Condensation and Contamination classes respectively. The results are shown in the confusion matrix in Figure 4.3

   Number of observed agreements: 102 (53.13% of the observations)

   Number of agreements expected by chance: 73.4 (38.22% of the observations)

   Kappa= 0.241

   SE of kappa = 0.048. 95% confidence interval: From 0.147 to 0.336. The strength of agreement is considered to be 'fair'.

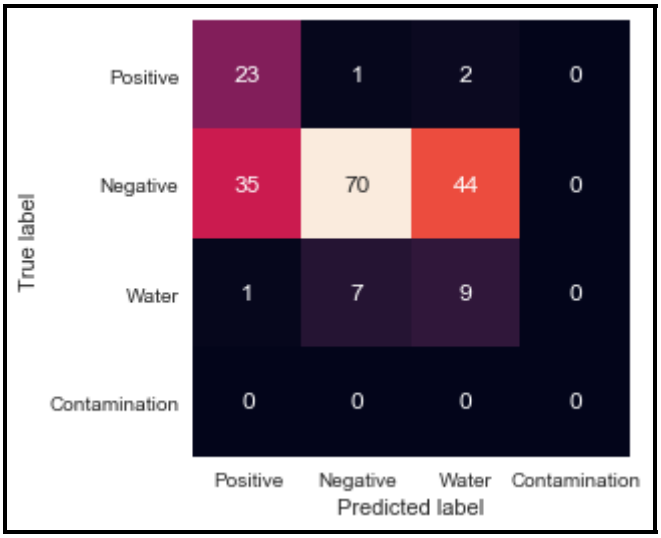


Figure 4.3: Confusion Matrix for Random Forest classifier with a training data set size of 313 wells across all 4 well categories.

(a) **Positive Class** A high degree of accuracy was seen for positive classes. Of 26 test samples, 23 were classified correctly.

(b) **Negative Class** Negative classes continue to suffer from mis-classification into nearly equal number of positive and water classes.

(c) **Condensation Class** Just over 50% of the samples with water or condensation in them were classified correctly. This is similar to the results of the Negative class. It is possible that the causes of missclassification of Negative into Water samples is related to the causes of mis-classification of Water into Negative. Therefore, an improvement in the accuracy of Negative or Water classification will contribute significantly to the overall improvement of accuracy.

(d) **Contamination Class** No classes of Contamination were available in the test sample.

2. **Model trained with a larger training data set** Here the random forest classifier was trained with a larger training data set, 1.8 times the training data as in the previous test. It comprised of a total of 571 training samples with 158, 300, 97 and 16 well samples for Positive, Negative, Condensation and Contamination classes respectively.

   It must be noted that the number of Negative class training samples were 3 times the number used in the previous test. The results shown in the confusion matrix in Figure 4.4 indicate how the classifier has used the additional training samples with a significant increase in the accuracy of the Negative class.

   This was the best test result obtained with an accuracy score of 77.6%. The Cohen-Kappa score was 0.349.

   SE of kappa = 0.075. 95% confidence interval: From 0.202 to 0.496. The strength of agreement is considered to be ‘fair’.

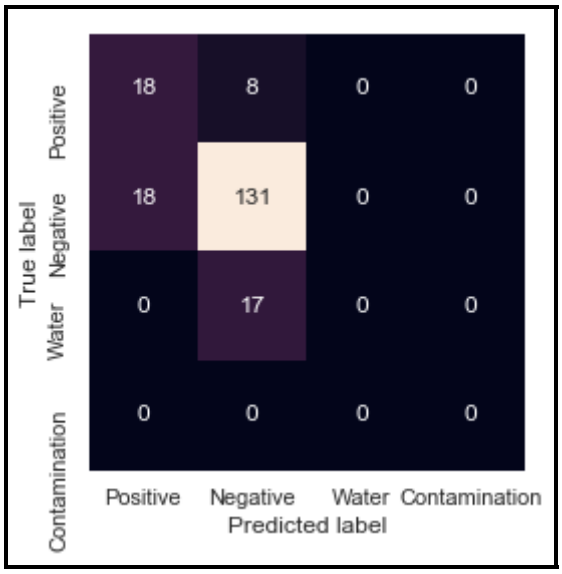


Figure 4.4: Confusion Matrix for Random Forest classifier with a training data set size of 571 wells across all 4 well categories.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In order for the machine learning based TB classifier to be useful in practice, we would like it's accuracy to be at least 90%. This is based on the False Discovery Rate (FDR) or Q-values of the current MPN calculator which is normally set at 0.05 (5%) or 0.01 (1%) depending on how relaxed or how stringent one desires the diagnostic criteria to be. This requires 9 out of 10 wells need to be classified correctly, which extrapolates to at least 43 out of 48 wells to be classified correctly for a 1% FDR.

As discussed in the previous Results section, the Random Forest approach with limited amount of training data resulted in a classification accuracy of 77.6%. While this level of accuracy is not yet sufficient to match human accuracy levels, it must be noted that the human classification is limited to identifying the Positive class as can be seen in the inputs provided the MPN calculator tool shown in Appendix A.1, while the ML based classification performs two additional classifications, Water/Condensation and Contamination. If the Water/Condensation class is also considered as Negative, then the ML classification accuracy improves significantly to 84.56%.

At this point, it must be noted that the primary purpose of this research report was to establish a working foundation of data and machine learning model pipeline that can be easily reused and extended with a specific focus to improve the classification accuracy levels to the desired levels of 95%.

Having discussed the image processing and classification capabilities of the overall solution, this section discusses ways in which the classification accuracy can be further improved, in particular for the Negative and Water (condensation) classes which saw the highest number of classification errors.

It is reasonable to expect that the classification accuracy can be improved from current levels because the extent of model optimization done till this point was fairly minimal due to a greater emphasis laid on deeply understanding the problem domain; then identifying, selecting and extracting a number of features relevant to the learning task and finally comparing two different types of machine learning models with respect to their classification accuracy,

The task remaining then is to further explore a set of choices in training data sizes, feature selection, feature extraction and model hyper-parameter optimization techniques that yield a much higher level of classification accuracy.

The choice of an accuracy improvement approach will clearly need to be specific to the ML algorithm. For example, in the GMM model, it is felt that testing multiple combinations of feature selection techniques (histogram of grayscale, histogram of LBP and histogram of HSV) and feature extraction techniques (circular versus rectangular regions) could yield better results than just hyper-parameter optimization as was discussed in the results section using different values of K (number of gaussians per well class). This is borne out by the fact that changes in the values of K did not result in a significant improvement in accuracy. In fact, higher values of K caused a deterioration in classification accuracy of the GMM model. The non-parametric Random Forest model yielded better classification accuracy than GMM for the scenarios tested in this work.

This work has demonstrated the feasibility of applying image processing, computer vision and machine learning techniques to classify sputum culture samples of TB patients. The techniques described in this work can be used to augment the current manual process to classify these culture samples. Machine learning based classification techniques can allow the culture samples to be processed and classified at the points of collection which are closer to the patients location, instead of requiring them to be sent to a centralized processing center such as the NHLS. This can significantly reduce the time and cost to process TB culture samples, thereby improving the overall efficiency and effectiveness of diagnosis of Tuberculosis.

Further, the machine learning based classification techniques discussed in this work need a relatively small sample size of labelled training data unlike other deep learning techniques like CNN that require a much larger corpus of labelled training data samples. The approach described in this work can be used by TB diagnostic laboratories who do not have sufficient human resources to allocate to the task of labelling training data, but still wish to benefit from machine learning and AI based techniques to improve their operational effectiveness. These techniques can also be applied to other disease diagnostics use cases that are based on sputum culture samples, beyond TB.

5.2 Future Work

This section describes future work that can be undertaken to further improve classification accuracy using feature selection and model tuning.

1. **Texture analysis over a circular region of interest instead of nearest enclosing rectangular boundary**

   Reuse the pre-extracted circular region of the image for texture analysis (as is done currently using Hough Circles) instead of re-extracting the bounding rectangular region using contour analysis. This will prevent the well borders from being included and could improve the classification accuracy.

2. **Use of number of contours as an additional feature**

   The number of contours itself within a particular image could be a useful feature for

classification. For example, a positive well image will have more contour shapes than other classes. This is a feature that can be used in training the model.

3. **Use of Hue and Saturation as feature descriptors for brightness of a well**

   A combination of H, S and V histogram values could be used to improve the classification accuracy. Using all the 3 channels could result in an improvement in testing results instead of just the V (Value) channel used to train the current model. In fact, the V values closely mirror the grayscale values of the pixel and do not seem to add additional information about the well image as was seen in Figure 3.27.

4. **Use of a Feature Minimization technique**

   A Feature Minimization technique like PCA (Principle Components Analysis) can be used to simplify the data sets. The impact of this on the performance of the learning models is identified as a further area of work.

5. **Larger training and testing data sizes**

   Of more than 600 images available at this point, only 25 images were used for training which represents just 4.1% of the overall data set. It is possible that a larger training data set allows for a more generalized model to be trained. For testing, only 4 images were randomly selected outside of the 25 training images. This is a very small portion of the overall data available and the classification accuracy discussed above may not be sufficiently representative of the full data set.



(a) With 225 training samples

(b) With 571 training samples

Figure 5.1: Learning curves with training and validation accuracy scores with different training data sizes. 20% of the training data was held-out as a validation test for cross-validation

The learning curve in Figure 5.1 shows an increase in cross-validation scores with increasing amounts of training data. Even at the maximum training data sizes shown in both the learning curves above, the validation accuracy approached a maximum of 85%

One of the constraints encountered in this project was the manual effort needed to update the training data spreadsheet with the training labels for each of the 48 labels in each well. This time spent on this was limited by the capacity of one individual, i.e. the author. The size of the training data available could be dramatically enlarged by enlisting additional

resources to label the training data. Effort can also be directed towards labelling more training samples for very low occurrence categories like Contaminated wells.

6. **Training Gaussian Models for each of the different feature categories per class**

   As was discussed in the the exploration and visualizations of the each of the three key feature categories per class - Histogram of Grayscale values, Histogram of Textures and Histogram of Brightness (using HSV values), the shapes of the histogram distributions of each of these feature categories differ significantly from each other. Therefore, a better fitting GMM can be obtained by using a separate GMM for each feature category instead of a single GMM model for all three feature categories.

   For the inferencing step, a mean of the log probabilities from each of the three GMMs can be taken instead of the single log probability score per category that is used in the current approach.

7. **Use of the trained GMM as a generative model to generate additional training samples**

   As the GMM is primarily a generative model, it can be used to generate additional training samples per category. The generated or synthetic training data can then be further transformed using transformations like scaling, rotations or affine transformations to increase the variety within the generated training samples. This will mitigate the need for manual labelling of the training samples.

8. **Use of GridSearch to find optimum values for the model hyper-parameters**

   Hyper-parameters are parameters of the model whose values are set before the training process begins. They are not directly learnt within the estimators (i.e. the specific scikit-learn implementation of a machine learning algorithm). In scikit-learn, they are passed as arguments to the constructor of the estimator classes (Scikit-learn 2018a).

   For example, in the RandomForestClassifier estimator, the main hyper-parameters are the following:

   (a) *max_features*: This is the number of trees in the forest. The larger the better, but also the longer it will take to compute. In addition, note that results will stop getting significantly better beyond a critical number of trees.

   (b) *max_features*: This is the size of the random subset of features to consider when splitting a node. The lower the greater the reduction of variance, but also the greater the increase in bias.

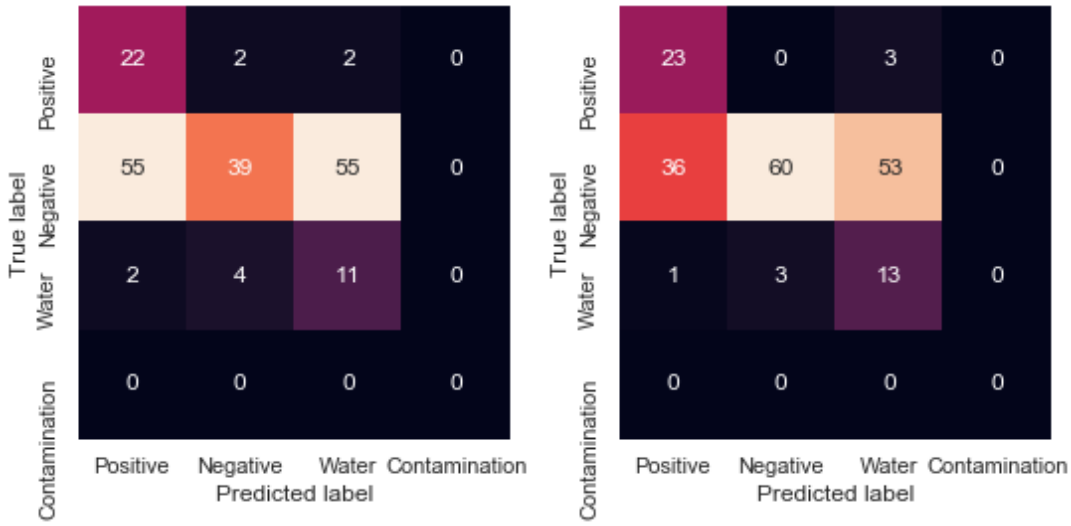
   For instance, an initial tuning of the hyper-parameters values shown in Table 5.1 resulted in an increase in the accuracy scores as shown in Figure 5.2.

9. **Use of Bayesian Gaussian Mixture approach**

   As a future extension, it is also proposed to use the Bayesian Gaussian Mixture approach which implements a variant of the Gaussian mixture model with variational inference algorithms.

Changes in hyper-parameters with resulting increase in accuracy		
Hyper-parameter	Previous value	Modified value
n_estimators	15	150
max_features	4	30
max_depth	100	None
min_samples_split	8	2

Table 5.1: Changes to select hyper-parameter values in the Random Forest Classifier



(a) With previous values of hyper-parameters (b) With modified values of hyper-parameters

Figure 5.2: Confusion Matrix for Random Forest classifier with two different sets of hyper-parameter values shown in Table 5.1

Variational inference extends the Expectation-Maximization (EM) by maximizing a lower bound on model evidence (which includes priors) instead of data likelihood. Variational methods are based on the same principle as expectation-maximization but also add regularization by integrating information from prior distributions (Scikit-learn 2018b) (Tzikas *et al.* 2008).

The variational algorithm needs more hyper-parameters than EM due to it's bayesian nature. The most important hyper-parameter is the concentration parameter. Specifying a low value for the concentration parameter will make the model put most of the weight on a few components and set the remaining components weights close to zero, while setting higher values for the concentration parameter will allow a larger number of components to be active in the mixture.

The key benefits of the variational inference approach are an automatic selection of the number of gaussian components and lesser sensitivity to the number of hyper-parameters.

10. **Use of Convolution Neural Networks**

Convolution Neural Networks (CNN) have been used to successfully classify other forms of TB diagnostic data sets such as chest radiographs (Stirenko *et al.* 2018). They were not explored in this work primarily due to the much smaller training data size that was used for classification. However, if additional effort can be directed to labelling the TB culture assay plate images that are available with the NHLS, then CNNs can also be used for the classification exercise.

Bibliography

[Cloete 2016] Kim Cloete. Treatable TB meningitis disables too many SA children. *Health24*, June 2016. https://www.health24.com/Medical/Tuberculosis/News/treatable-tb-meningitis-disables-too-many-sa-children-20160614.

[Gonzalez and Woods 2016] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 2016.

[Goodfellow *et al.* 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[James *et al.* 2017] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Stastical Learning with Applications in R*, volume 7th Printing. Springer Texts in Statistics, 2017.

[M.Bhuvaneswari 2016] M.Bhuvaneswari. Gaussian mixture model: An application to parameter estimation and medical image classification. *Journal of Scientific and Innovative Research*, 2016. http://www.jsirjournal.com/Vol5_Issue3_08.pdf.

[Pedregosa *et al.* 2011] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Prince 2012] Simon J.D. Prince. *Computer vision: models, learning and inference*. Cambridge University Press, 2012.

[Roser *et al.* 2018] Max Roser, Sophie Ochmann, Hannah Behrens, and Hannah Ritchie. Eradication of Diseases. *Our World in Data*, October 2018. https://ourworldindata.org/eradication-of-diseases#tuberculosis.

[Scikit-learn 2018a] Scikit-learn. Ensemble methods. *Scikit-learn*, 2018. https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees.

[Scikit-learn 2018b] Scikit-learn. Gaussian Mixture Models. *Scikit-learn*, 2018. https://scikit-learn.org/stable/modules/mixture.html.

[Stirenko *et al.* 2018] S. Stirenko, Y. Kochura, O. Alienin, O. Rokovyi, Y. Gordienko, P. Gang, and W. Zeng. Chest x-ray analysis of tuberculosis by deep learning with segmentation

and augmentation. In *2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO)*, pages 422–428, April 2018.

[Tzikas *et al.* 2008] Dimitris G Tzikas, Aristidis C Likas, and Nikolaos P Galatsanos. The variational approximation for bayesian inference. *IEEE Signal Processing Magazine*, 25(6):131–146, 2008.

[Unger *et al.* 2017] Katharina Unger, B. A. Honours Thesis, and Anton Hermann-Straße. Evaluation of machine learning frameworks on tuberculosis classification of chest radiographs. 2017. https://www.semanticscholar.org/paper/Evaluation-of-Machine-Learning-Frameworks-on-of-Unger-Thesis/5da1cec2e74e6056181206751299616 8f1edb562.

[VanderPlas 2016] Jake VanderPlas. *Python Data Science Handbook*. O'Reilly Media, 2016. https://jakevdp.github.io/PythonDataScienceHandbook.

[WHO 2018] WHO. Tuberculosis. *World Health Organization*, September 2018. https://www.who.int/news-room/fact-sheets/detail/tuberculosis.

[Xiong *et al.* 2018] Yan Xiong, Xiaojun Ba, Ao Hou, Kaiwen Zhang, Longsen Chen, and Ting Li. Automatic detection of mycobacterium tuberculosis using artificial intelligence. *Journal of Thoracic Disease*, 2018. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5906344/.

Appendix A

MPN Calculator

Figure A.1 shows the spreadsheet based MPN calculator that is used by the NHLS. Note that only the number of Positive tubes or wells is used an an input to the calculator.

MPN calculation program

This Excel program is free software. It is distributed in the hope that it will be useful, but without any warranty.
For assistance, contact wilrich@wiwiss.fu-berlin.de

How to use this program:

Enter the required data in the yellow cells and start computation by pressing "Ctrl+m". The results will be shown in the green cells (and are repeated to the right of each yellow input-table).
You can recalculate at any time by pressing "Ctrl+m" again.
You can enter data for up to 10 different cases with up to 10 different dilution factors.
The results shown in the green cells are based on the equations shown in the sheet "Equations" for information.
Please note that an MPN that falls into rarity category 3 is unreliable and should not be used.
Examples corresponding to the first three lines of the three tables in ISO 7218 are given in the sheet "Examples".

Name of experiment:	Sample 17 RV
Date of experiment:	
Target organism:	
Name of matrix:	
No. of case:	1
Name of case:	

dilution factor d*	volume in ml or g w**	no. of tubes n	no. of positive tubes x
0,1	0,5	3	3
0,01	0,5	3	3
0,001	0,5	3	3

Results of MPN calculation program

No. of case i	Name of case $case_i$	MPN	$\log_{10}$ MPN	SD $\log_{10}$ MPN	95% Confidence Limits Lower	95% Confidence Limits Upper	Rarity Index	Category
1		42000000	7,6	0,24	14000000	120000000	0,069	1
2		480000000	8,7	0,32	110000000	#########	1,000	1
3		290	2,5	0,27	86	1000	0,426	1
4		46	1,7	0,31	11	190	1,000	1
5		4800	3,7	0,31	1100	19000	1,000	1
6		8500	3,9	0,33	1900	38000	1,000	1
7		460	2,7	0,31	110	1900	1,000	1
8		46	1,7	0,31	11	190	1,000	1
9		4600	3,7	0,31	1100	19000	1,000	1
10		460	2,7	0,31	110	1900	1,000	1

No. of case i	Name of case $case_i$	MPN	$\log_{10}$ MPN	SD $\log_{10}$ MPN	95% Confidence Limits Lower	95% Confidence Limits Upper	Rarity Index	Category
1		42000000	7,6	0,24	14000000	120000000	0,069	1

* d_i is the dilution ratio used for inoculating that series of tubes: i.e. 1.0 (undiluted), 0.1 (diluted 1 in 10), 0.01 (diluted 1 in 100), etc...
** w_i is the volume of the dilution added to each tube in that series

Program dated 2010-07-14 | Equations | Examples

Figure A.1: Spreadsheet based MPN Calculator tool

Appendix B

Relevant Code Listings

Each of the code listings here demonstrate a specific technique used within the machine learning pipeline developed for this TB culture classification application.

B.1 Feature extraction from well regions within an image of a sputum culture assay plate

```
        # apply the OR operator to get the white pixels using the mask
        circle_image = rect_image | rect_mask

        # get all pixel coordinates where value is 255
        ind = np.where(circle_image==255)

        # get the pixel values for those coordinates from the original
    image - full 3 col or grayscale only?
        sample = gray[ind]
```

Listing B.1: Use of masks to extract pixels within a well region

B.2 Extraction of image texture information within a well

```
radius = 3
# Number of points to be considered as neighbourers
no_points = 8 * radius
# Uniform LBP is used
lbp = local_binary_pattern(crop, no_points, radius, method='uniform')
```

Listing B.2: OpenCV's LBP algorithm to obtain a comparison of pixel intensities of each pixel to it's 24 neighboring pixels

B.3 Use of contours to create a rectangular region around a well

```
def get_lbp(gray, ind):
    hsv_mask = np.zeros(gray.shape, np.uint8)
    hsv_mask[ind] = gray[ind]

    _,thresh = cv2.threshold(hsv_mask,1,255,cv2.THRESH_BINARY)

    contours = cv2.findContours(thresh,cv2.RETR_EXTERNAL,cv2.
    CHAIN_APPROX_SIMPLE)
    x,y,w,h = cv2.boundingRect(contours[0])

    crop = gray[y:y+h,x:x+w]


    radius = 3
    # Number of points to be considered as neighbourers
    no_points = 8 * radius
    # Uniform LBP is used
    lbp = local_binary_pattern(crop, no_points, radius, method='uniform')
    return lbp
```

Listing B.3: Use of contours to extract the bounding rectangle of a well on which texture analysis using LBP is performed

B.4 Training the GMM classifier

```
clf = mixture.GaussianMixture(n_components=int(N_COMP), covariance_type='
    full', verbose=1)
g_pos = clf.fit(x_train)
```

Listing B.4: Use of the scikit-learn's GMM classifier function to train the classifier for each class of well, in this case for the positive class represented by g_pos

B.5 Inferencing using the results returned by the GMM classifier

```
        # collect the features in a list
        data = (sample, lbp, hsv)

        x_reduced = get_feature_array(np.array(data)[np.newaxis])
        d = dict()

        d['P'] = g_pos.score_samples(x_reduced)
        d['N'] = g_neg.score_samples(x_reduced)
        d['W'] = g_conden.score_samples(x_reduced)
        d['C'] = g_contam.score_samples(x_reduced)

        label = str(keywithmaxval(d))
        y_hat[key] = label
```

Listing B.5: Scoring a test well with each of the 4 GMM models to select the class represented by the model returning the maximum score