FYP17028

Classification for Pathological Images Using Machine Learning

Final Report

Chi Ian Tang

3035209241

Supervisor: Dr. S. M. Yiu

Department of Computer Science

The University of Hong Kong

15 April 2018

Abstract

Diagnostic microscopy is currently used for the diagnosis of many common infections including bacterial vaginosis and malaria, but the dependence of diagnostic microscopy on human expertise limits its availability. Recent attempts of using machine learning algorithms in the development of automated diagnostic tools have been successful. In this project, three prototypes of a diagnostic system for bacterial vaginosis were developed using a number of computer vision and machine learning algorithms. The best performing prototype has an accuracy of 71.4% in correctly identifying the degree of infection and an accuracy of 85.7% in identifying the Nugent Score within an error margin of 2. A deployable version of the diagnostic system was developed, which can be used as the starting point for developing a robust and professional diagnostic system.

Acknowledgment

I would like to express my sincere gratitude towards Dr. S. M. Yiu, the supervisor of the project, for his time and effort in guiding me in the entire project, as well as giving me timely feedback on my work. I would like to express my appreciation to Dr. Dirk Schnieders, for his help and professional suggestions in the aspect of computer vision, and Ms. Bingbin Liu, who spent much time in the setting up the framework and building the first prototype of the diagnostic tool. In addition, I would like to express my gratitude towards Professor Patrick C. Y. Woo, for his professional feedback given in the perspective of a medical professional, and his help in the arrangements in the acquisition of blood smear images. I also want to express my appreciation to Mr. Chris Chi-Ching Tsang, for his help in arranging meetings with the medical experts as well as his help in facilitating communication between different parties, and also to all the medical experts for their help in the laborious data labelling tasks.

Table of Contents

List of Figures

List of Tables

Abbreviations

BV	Bacterial vaginosis
DBSCAN	Density-based spatial clustering of applications with noise
Fast R-CNN	Fast Region-based Convolutional Neural Network
Faster R-CNN	Faster Region-based Convolutional Neural Network
HIV	Human immunodeficiency virus
IoU	Intersection over Union
GPU	Graphical Processing Unit
MSER	Maximally stable extremal regions
R-CNN	Regions with CNN features
SSD	Single Shot MultiBox Detector
SQL	Structured Query Language
YOLO	You only look once

1. Introduction

1.1. Background

Bacterial infection is a common medical condition in humans, and several pathogens were found to be responsible for the development of malignant tumours [1]. Bacterial vaginosis (BV), one of the most common bacterial infections found in women at reproductive age, was estimated to affect tens of millions of people in the United States of America alone [2]. The prevalence of this infection varies by countries and can be as high as 50% [1]. Studies have also shown that this infection increases the risks of being infected with human immunodeficiency virus (HIV) [1], [3]. The Nugent Score System [4], which involves the investigation of Gram-stained vaginal smears from patients, is considered to be the gold standard in diagnosing bacterial vaginosis [1], [5], [6]. In addition, diagnostic microscopy is also the main diagnostic method for parasitic infections, including Malaria, in major hospitals [7], [8].

1.2. Motivation

Diagnostic microscopy, however, requires a considerable amount of training and skills, where the accuracy of the diagnosis often depends on the experience level of the microscopist [6], [9]. Furthermore, it could be time-consuming since it involves human diagnosis, and hence could be expensive for patients. In the light of the prevalence and consequences of aforementioned infections, an automated process could reduce the dependency on human expertise and provide a more affordable way to perform diagnosis.

Attempts in applying machine learning techniques to the diagnosis of several common infections have been successful with a high level of performance [10]. However, an automated system for the diagnosis of bacterial vaginosis based on patients' smears is still not available.

1.3. Objectives

This project aimed to explore the possibilities in employing machine learning and computer vision techniques in diagnostic microscopy. An automated diagnosis system for bacterial vaginosis was developed, such that the time and cost of bacterial vaginosis diagnosis could be reduced.

In addition, an analysis on the development process, especially in terms of limitations of applying machine learning and computer vision techniques in medical contexts was explored, such that this project can be used as a guideline for future projects using similar techniques.

1.4. Scope

The project consists of two main components. First, an automated diagnostic tool which estimates the degree of infection based on a blood smear image, with a simple interface was developed. The auxiliary data collection tool, which facilitated the collection of detailed information of blood smear images, was also developed. This project did not directly involve the acquisition of blood smear images from patients nor the labelling process, where data were provided by the medical experts from the Li Ka Shing Faculty of Medicine, the University of Hong Kong, and other publicly available sources. Second, a report on development process and limitations of applying machine learning techniques in diagnostic microscopy was produced.

1.5. Related Works

A number of recent studies have made use of computer vision and machine learning techniques on diagnostic microscopy. In particular, Quinn et al. [10] explored the use of convolutional neural networks (a machine learning algorithm) in detecting several infections including tuberculosis and hookworm. The detection tools were successfully developed with high accuracy. Kraus et al. [11] combined convolutional neural networks and image segmentation with multiple instance learning in classifying segmented images only using high-level annotations for the entire image. These studies showed that deep learning techniques had a range of advantages in diagnostic microscopy and demonstrated significant improvements over traditional techniques.

1.6. Outline of the report

The remainder of this report starts by outlining the major deliverables of the project, followed by the methodologies employed in the development of the deliverables. The results including the performance of the detection tools are then given. Major difficulties encountered and mitigation strategies in the development process are elaborated in the next section, followed by a conclusion at the end.

2. Outline of Deliverables

2.1. Data collection tool

A simple auxiliary tool, *Clicklable* (see Section 3.3.3 for detailed information), for labelling the microscopic images was developed to facilitate detailed data labelling. Data labelling is an essential step for supervised machine learning, where known truths about input data are annotated with the expected output and these input / output pairs are then used to train machine learning models. This tool facilitates this process by allowing the user to load a blood smear image and perform labelling by clicking and dragging using a pointer device to mark regions where bacteria are present. The labelled points are visualized with a shape around it, and high level of customization can be done. The aim is to reduce the time required for labelling the images, as well as to tailor the data representation.

2.2. Automated diagnostic system for bacterial vaginosis

The main objective of this project is to develop an automated diagnostic system for bacterial vaginosis with desirable accuracy. Most of the components will be written in Python, and a simple user interface is developed. A trained classifier will be the core component of this system, with other processing modules supporting the overall flow of the system, including image processing, segmentation and interpretation tools. This system will allow the user to select images and get predictions on the degree of infection.

In this project, three diagnostic system prototypes were developed with different sets of data, machine learning algorithms and model architecture. The three prototypes have different performances and resources requirement, and the final prototype with the highest accuracy is deployed as the automated diagnostic tool.

3. Methodology

3.1. Overview

In this project, an auxiliary image labelling tool was first produced. Images annotated with positions and types of bacteria, as well as the overall degree of infection by medical experts were then obtained.

Table 1. The Nugent Scoring system [4]

	Score		
Average Abundance per oil immersion field (1000X magnification)	Lactobacillus morphotypes	Gardnerella and Bacteroides morphotypes	Curved Gram-variable rods
0	4	0	0
< 1	3	1	1
1 – 4	2	2	1
5 – 30	1	3	2
> 30	0	4	2

The degree of infection is evaluated according to the Nugent Scoring system [4] (see Table 1), which is based on the average density of three types of bacteria present in the smear: Lactobacillus morphotypes (scored 0 – 4), Gardnerella and Bacteroides morphotypes (scored 0 – 4), and Curved Gram-variable rods (0 – 2).

Table 2. The interpretation of the Nugent Score [4]

Total Score	Interpretation
0 – 3	Normal
4 – 6	Intermediate
7 – 10	Bacterial vaginosis infection

The three scores for each type of bacteria is then summed to a score ranging from 0 to 10, which indicates the overall degree of infection (see Table 2).

After data collection, the development of an automated classification tool using machine learning typically involves the stages of pre-processing, segmentation, training and evaluation. The images are first pre-processed and segmented [12], and the segmented areas are then used as training data as well as testing data for the classification task. An evaluation of the performance of the classification task is carried out afterwards. These require both hardware and software support. The aim is to develop a machine learning model with high accuracy in estimating the degree of infection by examining a blood smear image.

3.2. Prerequisites

3.2.1. Hardware

The access to GPUs (Graphical Processing Units) will be required for training machine learning models. GPUs are optimized for parallel computations, and the nature of machine learning model training, which typically involves a large number of mathematical computations, is highly parallelizable and hence, can be completed in much shorter time using GPUs [13].

3.2.2. Software

Machine learning, computer vision and graphics libraries will be required in order to eliminate the time spent in developing such tools and to focus on the development of the diagnostic tool. A number of widely available libraries including Torch [14], Tensorflow [15] and Keras [16] are identified. Torch [14], a library implemented in the programming language Lua, was used in the first phase of the project because the auxiliary programs, which act as the interface between the machine learning training process and the GPUs available in the Department of Computer Science, are available from previous projects and using them could allow fast prototyping. Tensorflow [15] and Keras [16], libraries implemented in the programming language Python, were used in the development of the second and final prototypes because of the widely available resources including pre-trained machine learning models, tutorials and cloud computing platform support.

3.3. Data Collection

The collection of data is essential for the development of a classification system using machine learning methods, where these data are used for both developing and testing the system. Furthermore, the quality of data for training machine learning models has significant effects on the performance of the models [17]. Therefore, in order to achieve satisfactory performance, a tailored dataset is needed for this project. External sources of data tend to be limited in many ways, including the quantity of data, the variations among data, and the difficulty in combining different sources of data. The collaboration with the Li Ka Shing Faculty of Medicine, the University of Hong Kong, enabled efficient communication, the production of a tailored dataset for the project, as well as the possibility of obtaining original, unaltered data.

For supervised learning algorithms, data can be separated into two components: the input dataset, which consists of data to be given to the models, and the corresponding output dataset, which consists of facts known about the corresponding input data [18]. In this project, the input dataset consists of a mix of anonymised vaginal smear images collected from patients and bacterial colony images, and the output dataset consists of the detailed locations and types of bacteria, as well as the overall degrees of infection for vaginal smear images.

3.3.1. Smear Images

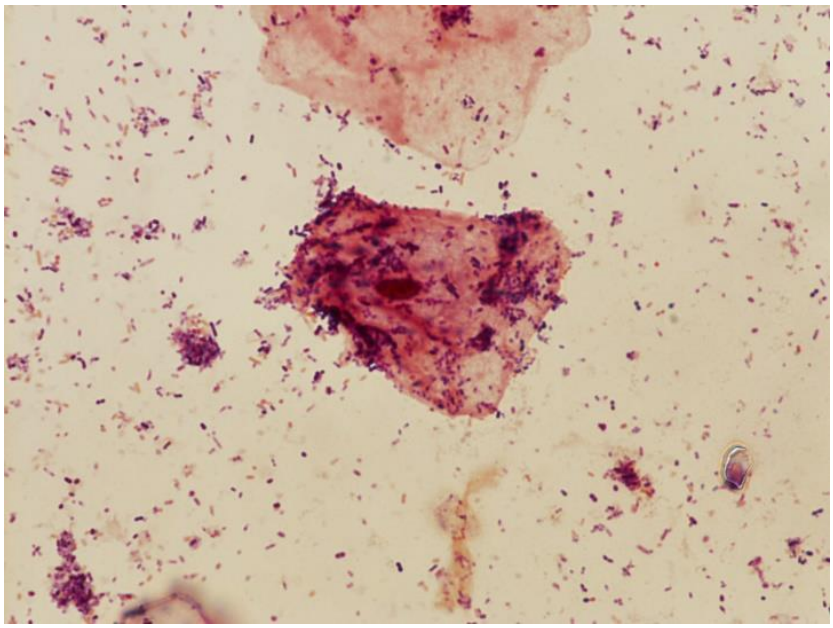

Figure 1. A Gram-stained vaginal smear

Images of Gram-stained vaginal smears (see Figure 1) of varying degrees of bacterial vaginosis infection were provided by the medical experts from the University of Hong Kong, where the images were anonymised and collected with the consent from the patients.

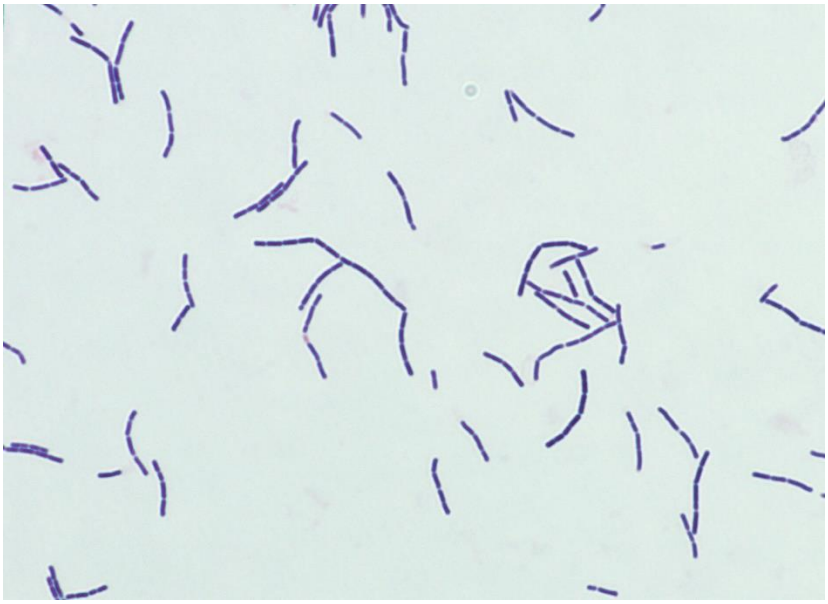

Figure 2. A Gram-stained bacterial colony smear

In addition, images of bacterial colonies (see Figure 2) were also obtained. Both types of images are collected to mitigate the shortcomings typically found in each type of images. Vaginal smear images are used in the actual diagnosis of infection by the medical professionals, and hence accurately capture the environment where the bacteria are found in

the human body. However, according to the feedback from the microscopists, many of the bacteria present in the smear images cannot be accurately identified by observing the images alone, due to the fact that some of the bacteria may share similar shapes and morphologies at different stages. On the other hand, although the bacterial colony images might not accurately capture the morphologies of the bacteria found in human body fluids, the purity of the specimen provides a guarantee on the type of bacteria that is present in these images. Hence, these two types of images are used in conjunction to achieve higher performance.

3.3.2. High-level Interpretation of Images

In order to develop an automated diagnostic tool, information about the images, including the degree of infection of the patient, is necessary. The high-level interpretations of the images including the Nugent Score and the overall degree of infection are usually readily available because they are usually recorded during diagnosis.

3.3.3. Detailed Labelling of Images and the tool *Clicklable*

However, the detailed labelling of the images, which includes the locations and types of bacteria of individual bacteria is usually not recorded due to the high amount of extra effort required.

Figure 3. The main user-interface of the data collection tool, *Clicklable*.

Hence, the aforementioned software, *Clicklable* (see *Figure 3*), was developed to facilitate the annotation.

3.3.3.1. User Interface of *Clicklable*

The tool *Clicklable* allows the medical professionals to load smear images previously captured and annotate the image by clicking and dragging on the locations of individual bacteria. There are two modes for labelling available: one is for marking point locations and the other is for annotating bounding box locations. Each annotated point location has a shape around it (for example, the red circle which can be at the top centre of Figure 3), and each annotated bounding box has the corresponding rectangular box drawn on the image.

Furthermore, a number of functionalities are implemented to enhance the user experience. When the user moves the cursor close to an annotation, the corresponding annotation is automatically highlighted (as seen by a semi-transparent white circle around the point, see Figure 3). A popup menu is shown when the user right-clicks on the annotation, and information about the annotation is shown (see Figure 4, where the location and the type of bacteria are shown as the first two items of the menu), together with functionalities to make changes to the label.

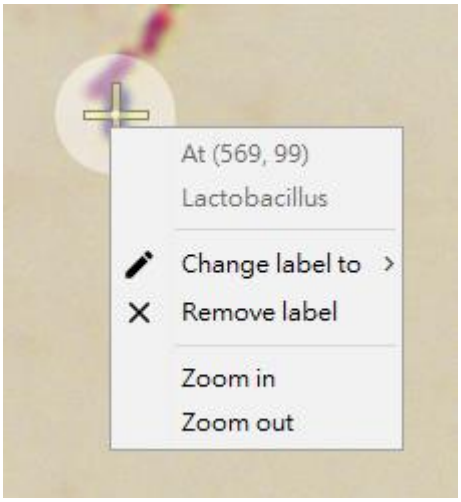

Figure 4. The popup menu in *Clicklable*

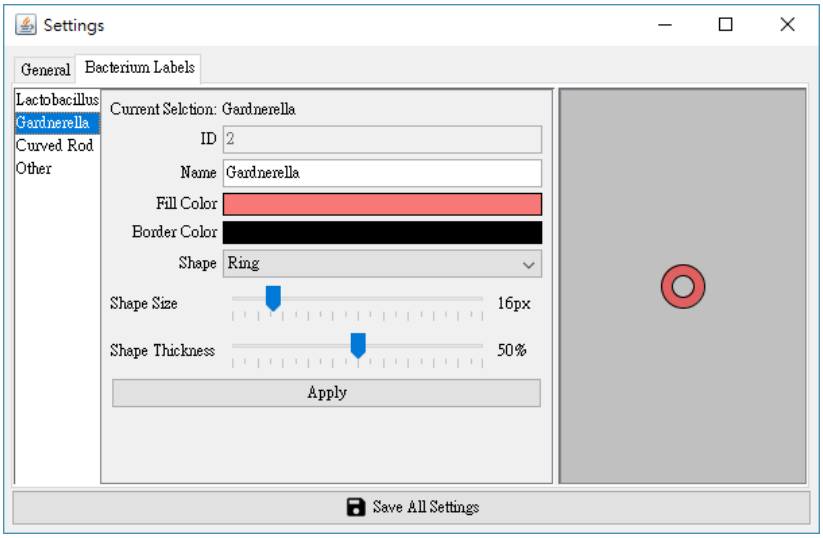

Figure 5. Annotation settings in *Clicklable*

In addition, the user-interface elements are highly customisable, where the user can choose the shape, fill and background colours, sizes of the annotations, by changing the corresponding settings (see Figure 5).

These functions allow the user to customise the tool to fit their needs.

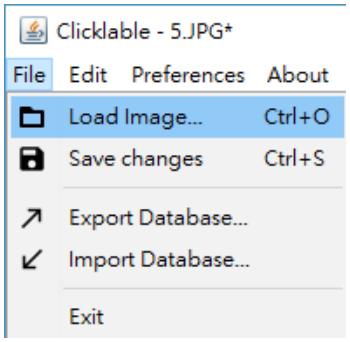

Figure 6. The "File" menu with options to export the data in *Clicklable*

The data recorded by the tool can be easily exported to a single data file to facilitate data transferral (see Figure 6).

3.3.3.2. Supporting technologies in *Clicklable*

In order to ensure the usability and reliability of the software, various technologies are used in the development of *Clicklable*.

Firstly, Java is the major programming language used in the development of *Clicklable*. Although it requires an installation of the Java Runtime Environment, it is widely available (over 15 billion devices run Java software [19]) and it is independent of the underlying operating system that it is running. This reduces difficulties in distributing this tool to different platforms and also lowers development time.

Secondly, all the data are stored in a database using a Structured Query Language (SQL) database engine, SQLite. SQLite is a reliable database engine which is resilient against failure and relatively light-weight [20]. Since the data stored in *Clicklable* are simple (only annotation and basic file information), the small amount of extra resources required, and the robustness of the engine are very desirable features.

3.4. Image Processing

After the collection of data, different image processing techniques will be employed to reduce the variations between training samples and hence to increase the reliability of the machine learning models. This involves the pre-processing stage and the segmentation stage.

3.4.1. Pre-processing

In this stage, variations between images due to different background lighting, degrees of staining and image acquisition techniques are calibrated.

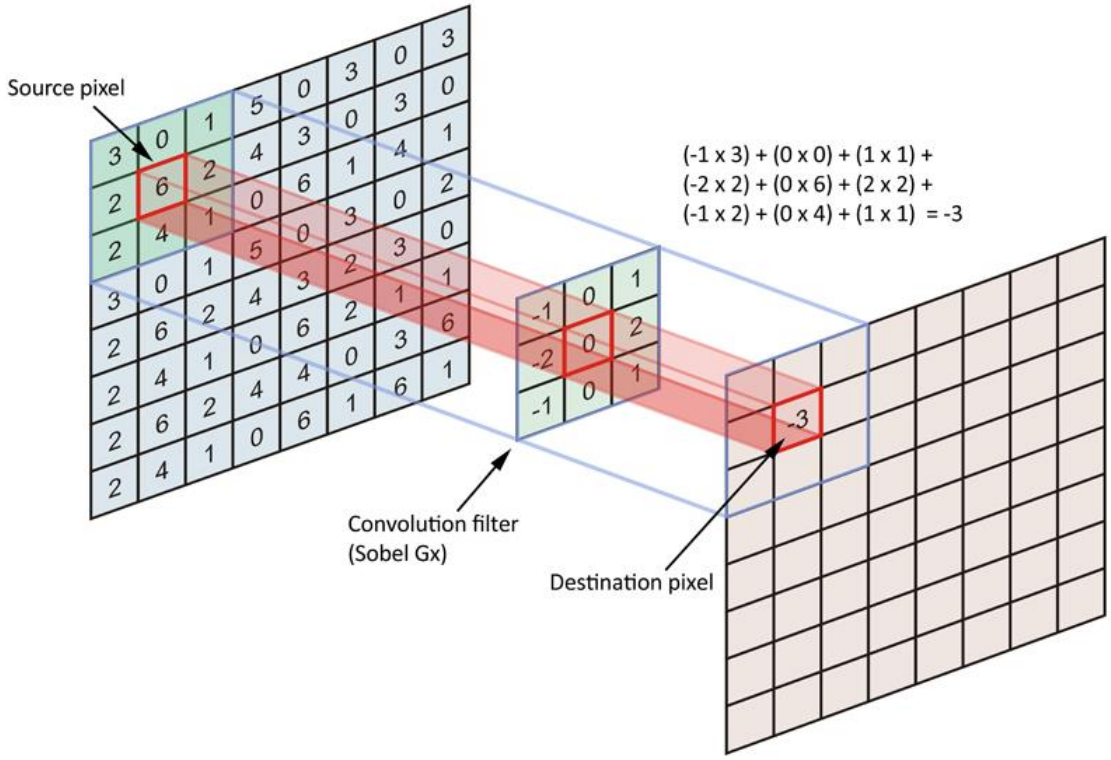

Figure 7. An example of convolution on an image with a Sobel filter [21]

Noises in images can often be effectively removed by applying spatial filtering, a computer vision technique which involves the convolution (also known as filtering) of the image with a kernel, a weighted matrix (see Figure 7). The convolution of an image f with kernel (also called convolution filter) g is defined on the set of real numbers as [22]:

$$(f * g)[\mathrm{m},\mathrm{n}] = \sum_{k}\sum_{l} f[\,\mathrm{m}-k,\mathrm{n}-l\,]\,\mathrm{g}\,[k,l]$$

The convolution operation combines the values of the neighbourhood of each pixel in the image. For example, as shown in Figure 6, the highlighted region around the source pixel, is convoluted with the convolution filter to obtain the destination pixel. The calculation is done by multiplying each value in the source region with the corresponding value in the filter, followed by a summation operation (as shown in the top-right corner of Figure 6). This is similar to perceiving an image at a distance, where the information of individual details is not directly perceivable, but rather the general information in an area. The size of the kernel as well as the weights of the kernel are adjusted for different uses. In particular, Gaussian filters

are commonly used for reducing noises before edge detection [23].

Level of illumination, on the other hand, can be effectively calibrated by subtracting an empty film (control image) [12], thresholding or analysing the histogram and apply histogram transformation.

Finally, the variations in the scales of images, if not handled properly, could result in meaningless estimations from the model due to that fact that the morphology, in particular, the length of bacteria is important in identifying the identity. This can be resolved when the magnification of the microscope is known. However, if the information is not available, it is possible to rely on the assumption that healthy human red blood cells and platelets have similar sizes and a more advanced technique called granulometric analysis [24], can be used to estimate the sizes of the cells and scale accordingly [12]. Different combinations of aforementioned techniques were applied in different prototypes according to the variations observed generally in the data.

3.4.2. Segmentation

Segmentation is the process of dividing the image into smaller patches of images. Two different segmentation strategies, namely region of interest segmentation and sliding window segmentation, were used in the development process of different prototypes, according to the classification algorithm used.

3.4.2.1. Region of interest segmentation

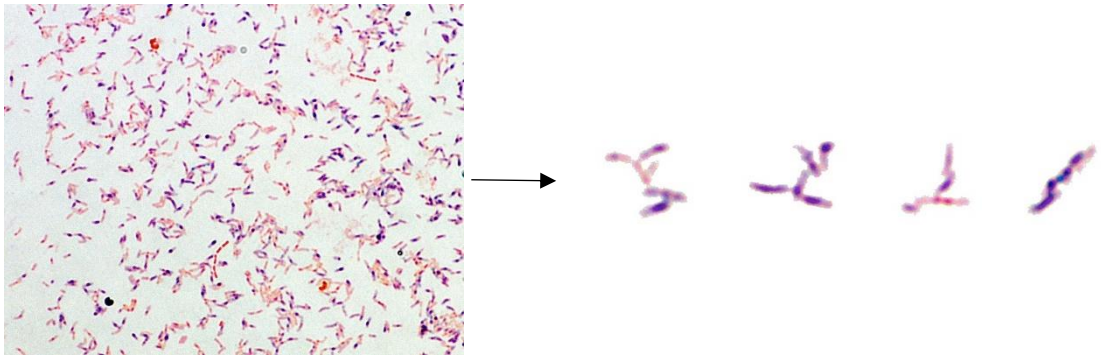

Figure 8. The segmentation of a smear image into regions of interest

The main goal of this segmentation strategy is to separate the images into small regions which contain one or more bacteria (see Figure 8). Blob detection algorithms, as well as data

clustering algorithms, are used for this task. In particular, Maximally stable extremal regions (MSER), Density-based spatial clustering of applications with noise (DBSCAN) and Otsu's method were used.

MSER, proposed by Matas et al. [25], is a blob detection method which was originally proposed for identifying the correspondence areas or objects of images taken from different perspectives. This method is adaptive to a number of common transformations in images taken of the same objects, where the regions identified are invariant to linear transformations of brightness and relatively stable [25]. These are the desired properties and features of the method used for segmenting the smear images, such that regions identified are not easily affected by the variation in illumination.

DBSCAN [26] is a data clustering algorithm which is used in identifying clusters in data points such that region of nearby neighbouring data points is identified as a single cluster. This algorithm is robust against outliers, as well as highly flexible in terms of the shapes of the clusters, which are applicable to the smear images.

Otsu's method [27] is a clustering-based image thresholding algorithm which is used in the reduction of grey-level images into binary images. An adaptive version of the Otsu’s method [28] can be used to adaptively separate background from foreground and cluster neighbourhoods of similar intensities. It is possible to specify the typical bacteria size as the window size and perform flexible clustering on smear images.

This segmentation strategy was used in both the first and the second prototypes of the project because the classification task is directly performed on the input images (see Section 3.5 Direct classification).

3.4.2.2. Sliding window segmentation

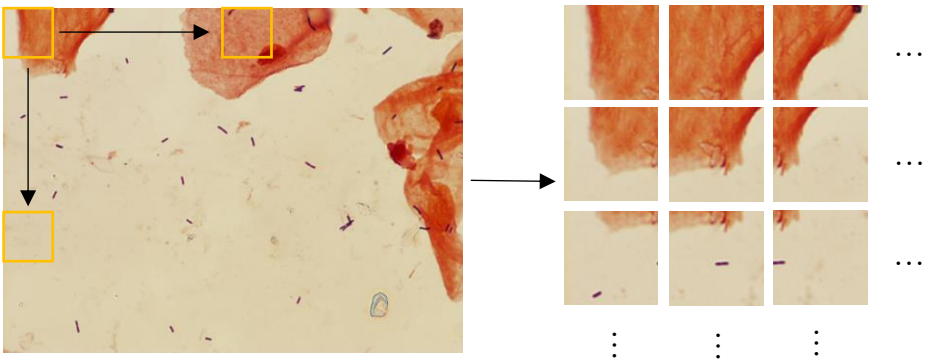

Figure 9. Sliding window segmentation on a smear image

This segmentation strategy aims to reduce the image size of individual image patches used for object detection (see Section 3.6 Detection). It involves sliding as fixed size image window across the entire image, cropping each patch covered by the window as a new image (see Figure 9).

This segmentation strategy was used in the final prototype of the project, because the classification and localization tasks are combined into one learning task, and due to the increased complexity in the machine learning model, smaller input images allow faster learning and require fewer resources (see Section 3.6 Detection).

3.5. Direct classification

After the region of interest segmentation, areas of interests or local frames of the images are identified. A classifier which distinguishes among the target bacteria categories, namely Gardnerella Lactobacillus, Gardnerella and curved rods is trained. In addition, different techniques in determining the number of bacteria in each area of interest were explored. A number of machine learning algorithms including neural networks, support vector machine and fuzzy logic are potential candidates for developing the classifier. This project mainly focuses on the use of neural networks.

3.5.1. Artificial Neural networks

Artificial neural network [29] is a machine learning algorithm, where a neural network is formed by combining a collection of artificial neurons. These artificial neurons are modelled by a mathematical function (see Figure 10), defined on the set of real numbers from inputs $x_0, x_1, \dots, x_i$, weights $w_0, w_1, \dots, w_i$, bias b, and activation function f to output y, as [30]:

$$y = f\left(\sum_i w_i x_i + b\right)$$

This aims to model a neuron in the human brain (see Figure 11), the fundamental unit of computation of the human brain.

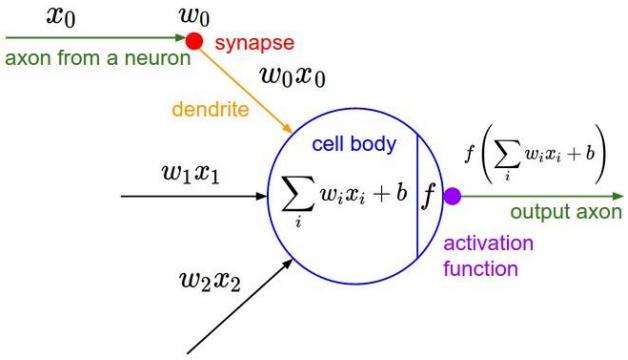

Figure 10. The mathematical model of a neuron (a node in artificial neural networks) [28]

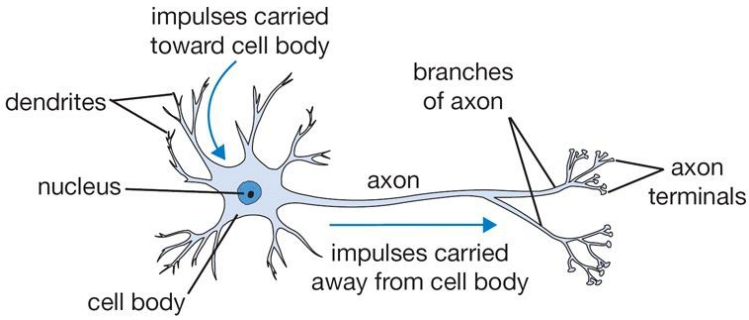

Figure 11. The diagram of a human neuron [28]

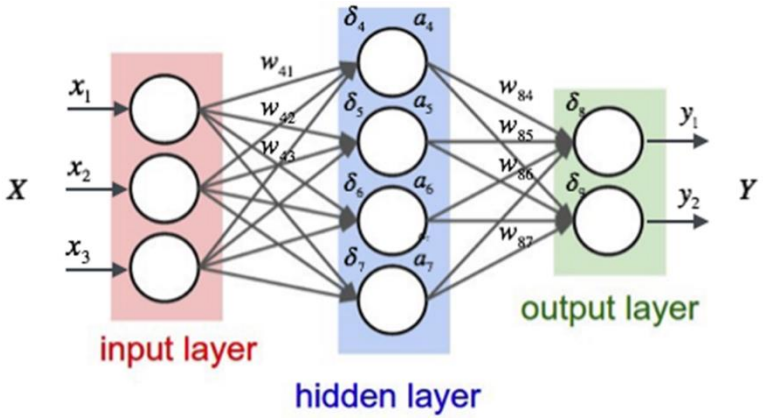

Figure 12. A 3-layer neural network [28]

These artificial neurons are then connected, to form an artificial neural network. They are separated into three groups: input, output and hidden (see Figure 12). For example, in Figure 12, there are 3 neurons in the input layer, 4 neurons in the hidden layer and 3 neurons in the output layer. These nodes are interconnected such that values except for the input nodes are calculated based on the values of other nodes and variable parameters. Supervised learning for classification, where each learning sample is provided with its desired output, involves finding the parameters in the network such that when presented with new data, the network is

able to generate desired output with high accuracy. This generalization process typically does not require handcrafted features or weightings of the input, where the network "learns" by inferring the relationship between the inputs and the outputs from the training samples. This significantly reduces the necessity of expertise in the related area for tailoring the important features. A variety of architectures have been proposed, mainly differing in how the network is structured, how the parameters are tuned, and what functions are used in the calculations. Different architectures are chosen based on the problem to solve.

3.5.2. Convolutional neural networks

Convolutional neural networks [31], a type of artificial neural networks, make use of the convolution operation (as presented earlier in section 3.4.1) in addition to standard linear operations.

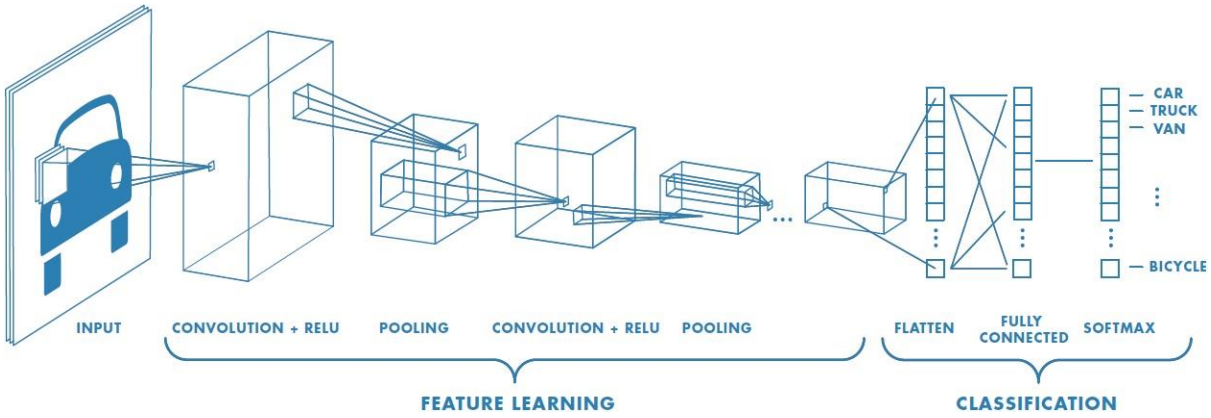

Figure 13. Example of a network with many convolutional layers. [32]

A convolutional neural network for classification typically starts with the input image as the input nodes and subsequently applies convolutions (filtering) and sub-sampling (max pooling) on the values until the output layer which indicates the likelihood of being in a certain category is reached (see Figure 13). During training, the weights of the convolution filters are adjusted to fit the expected outcome. This type of architecture is very effective in dealing with image inputs, due to the nature of the convolution operation which uses values from local neighbourhoods of the image. The first pattern recognizer which achieved human-level performance on several tasks was based on this learning method.

Since the segmented areas are essentially a part of the image and because of the successes seen in other similar projects, the convolutional neural network is a strong candidate for the architecture of the classifier in the project.

3.5.3. Residual Network

As the depth (number of layers) of artificial neural networks increases, more complex functions can be modelled. However, the problem of overfitting [33], where the machine learning model captures a function that is overly complex and does not generalize well to unseen data, becomes more significant as the model becomes more complex. Furthermore, as the complexity increases, it becomes more difficult for the model to optimize, where the training error and the testing error remain higher than a less complex model [34].

Regularization [35] is a common strategy for mitigating the effects of overfitting, in which the model is penalised if it heavily relies on a small set of nodes. In order to combat the problem of high training and testing error, the residual learning framework was proposed by He et al. [34].

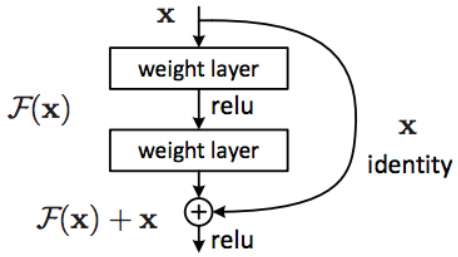

Figure 14. A shortcut connection in residual networks [34]

The residual learning framework involves the addition of the shortcut identity connection, where the output from one layer is directly added to layers beyond the next layer (see Figure 14). These connections allow layers to learn the identity mapping easily when an identity function is sufficient to capture the relation [34]. This is particularly useful for deep networks because in many cases, a deeper plain network (networks without shortcut connections) does not optimize as well on training as a shallower one even if the deeper network has higher modelling power [34], and the addition of shortcut connections helps the deep networks to capture a simpler function. In addition, these connections only slightly increase the computational complexity, which does not have a noticeable impact on training time for each iteration.

3.5.4. The network architecture of the first prototype

The first prototype has a machine learning model consisting of 7 convolutional layers together with 3 fully connected layers (see Figure 15). In Figure 15, “3x3 conv, 64” refers a convolutional layer with 64 filters, each of 3x3 filter size, “max pool, /2” refers to a maximum pooling layer with filter size 2x2, and a stride (number of position moved in each step) of 2, and “fc 1024” means a standard layer with 1024 neurons fully connected. Each convolutional layer is interleaved with a maximum-pooling layer, which is an operation also based on the convolution operation but only the maximum value in the filter neighbourhood is selected. A maximum-pooling layer with size 2x2 and stride of 2 effectively halves the input tensor in two dimensions, keeping only the maximum value of each 2x2 box (quartering in size). This follows the standard architecture for a convolutional neural network with pooling operations [36].

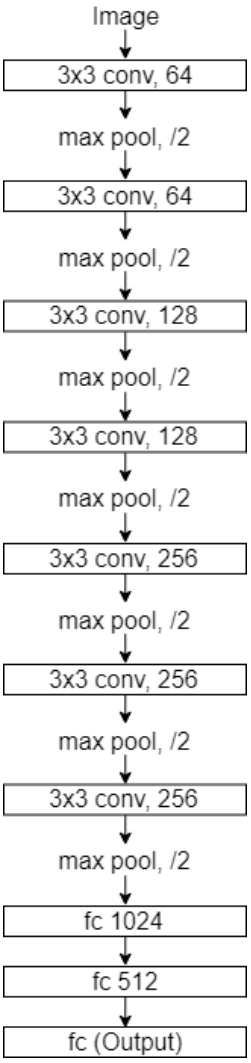

Figure 15. Network architecture for the first prototype

3.5.5. The network architecture of the second prototype

In the light of having contrasting performance in the first prototype (see Section 4.3 Performance of the first diagnostic tool prototype), the second prototype of the diagnostic system adopts a deeper residual network architecture with the view to increase performance as well as reduce the risk of having high training error.

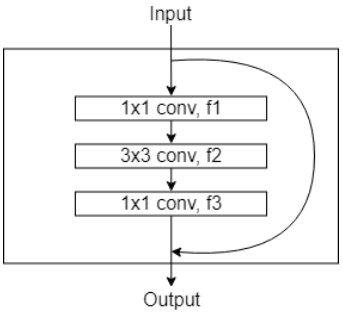

Figure 16. A residual block in a "Bottleneck" residual network

In this prototype, the Bottleneck variant [34] (see Figure 16) of residual networks is used, which has a similar complexity yet reducing the requirement for computational power. Furthermore, Bottleneck convolutional residual blocks (see Figure 17) are used to replace the maximum pooling layers, due to the change in dimensions after the first convolution with a stride of 2.

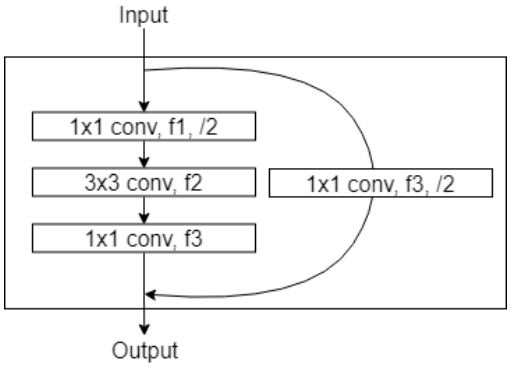

Figure 17. A convolutional residual block in a "Bottleneck" residual network

In addition, the operation of global average pooling, proposed by Lin et al. [37] is used towards the final layers, similar to that of the models used in [34]. A global average pooling layer, which is similar to a normal pooling layer, is a pooling across the entire spatial resolution of each filter, giving a vector with the size of the number of filters (see Figure 18). This has the effect of reducing overfitting compared to that of using fully connected layers [37].

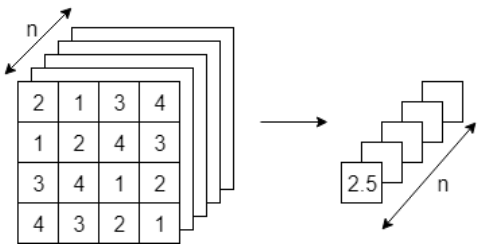

Figure 18. Applying global average pooling on a 3D tensor.

The overall architecture is composed of a total of 4 standard residual blocks, 4 convolutional residual blocks, 1 standard convolutional layer, 1 fully connected layer, 1 maximum pooling layer and 1 global average pooling layer (see Figure 19), with a total of 30 layers with trainable parameters (3 for each identity residual block and 4 for each convolutional residual blocks).

In Figure 19, a “res conv, 32-32-128” block is a convolutional residual block with Bottleneck structure (see Figure 17), with f1, f2 and f3 being 32, 32 and 128 respectively. A “res id” block is the standard residual block with identity shortcut connection.

This network architecture resembles the ResNet-50 network in [34], with fewer layers because of the computational constraints and the problem complexity.

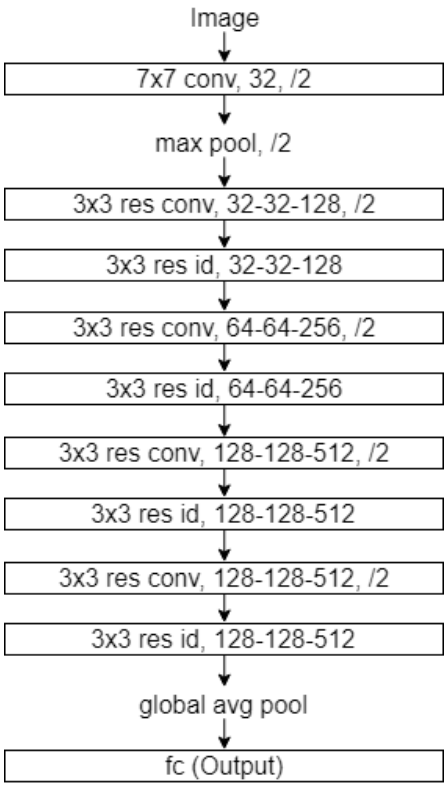

Figure 19. Network architecture for the second prototype

3.6. Detection

In the final prototype developed in this project, the object detection approach is taken instead of the separated segmentation and classification approach found in first two, with the aim of tackling some of the problems identified in the first and second prototypes (see Section 5.2 and Section 5.4).

Object detection is the task of combining object localization and object classification, where the target is to both find the position and the class of the objects.

3.6.1. Faster Region-based Convolutional Neural Network (Faster R-CNN)

Faster Region-based Convolutional Neural Network (Faster R-CNN) [38] is one of the top performing object detection algorithm based on deep learning. It is an improvement on Regions with CNN features (R-CNN) [39] and Fast Region-based Convolutional Neural Network (Fast R-CNN) [40], by combining the network for proposing potential regions and the classification network to form an end-to-end model for object detection.

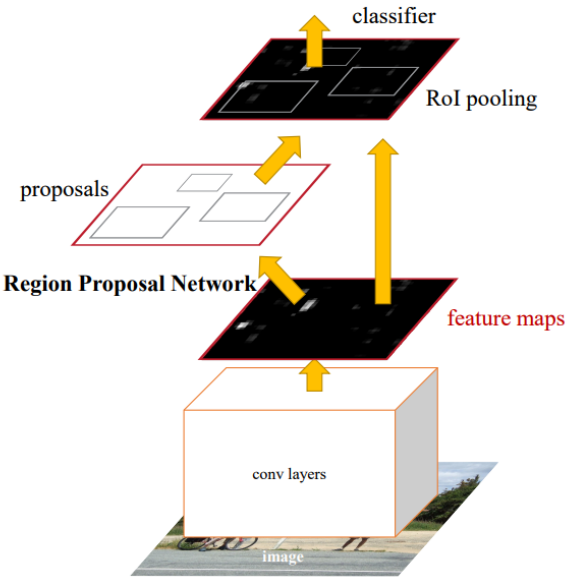

Figure 20. The Faster R-CNN architecture [36]

The Faster R-CNN detection systems consist of two main components: the Region Proposal Network which proposes regions that could contain objects in question, and the Fast R-CNN detector, which takes in the proposed regions and further refines the bounding boxes and classifies the region [40]. Earlier layers of the entire network are shared between the Region Proposal Network and the Fast R-CNN detector, in order to speed up the training and inference process [39].

3.6.2. The network architecture of the final prototype

In the final prototype, a pretrained base network of ResNet-50 [34] architecture is used. This network is versatile yet deep, so the model is pretrained with natural object detection datasets. The convolutional layers are then used as the first convolutional layers in Faster R-CNN, giving the output of a feature map. After the convolutional layers, one convolutional layer with filter size 3x3 and 512 filters followed by two separate 1x1 convolutional layers are used to form the Region Proposal Network, with one of them outputting the proposed

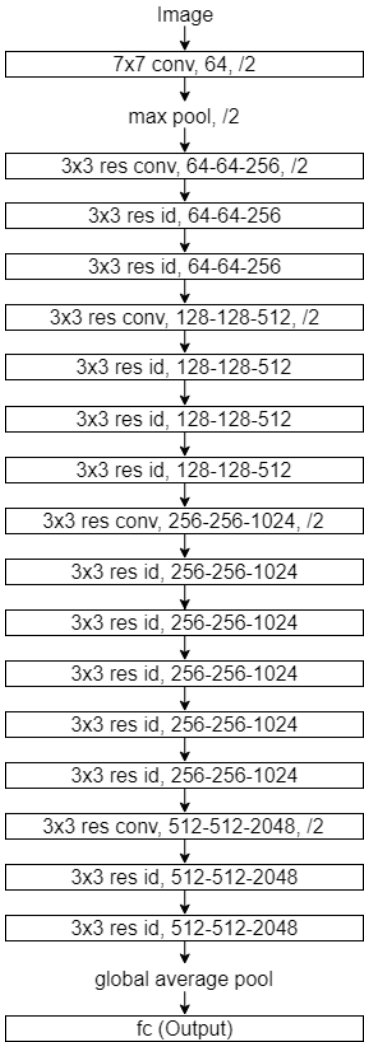

Figure 21. The ResNet-50 Architecture [32]

regions and the other with the corresponding confidence in its prediction. The model then extracts the corresponding parts of the feature map according to the proposed regions, and feed into two fully connected layers, and give a refined position of the bounding boxes and classifies the object present in the region [39].

3.7. Interpretation

After identifying the number of different bacteria present in the smears, a final data analysis which estimates the degree of infection will be done. The numbers of bacteria in each category of the Nugent Score System [4] are identified and an overall interpretation based on the same system is made.

3.8. Performance Evaluation

The performance of the models is evaluated by determining the accuracy of the predictions for images which are not used in training process. Performance evaluation can be separated into two stages: validation and testing. Data collected are typically separated into three sets accordingly: the training set, the validation set, and the testing set, where only the training set is used for training the model. The validation set is used to evaluate the performance of the model, and the hyper-parameters including the architecture, number of learning iterations, are tuned to maximize the performance. The testing set, on the other hand, is reserved for the final evaluation after the hyper-parameters are tuned and is used to reflect the generalizability of the model. The reason for separating the performance evaluation into two stages is that tuning hyper-parameters to maximize performance actually leaks certain information about the testing data into the model, which could lead to a false performance of the model because the model might only show good performance on the current dataset and fail to generalize [41]. Hence, an exclusive set of testing data is reserved for the final evaluation of the model.

In this project, there two sets of evaluation metrics used for each prototype, namely low-level and high-level evaluation.

3.8.1. Low-level evaluation of the first and second prototypes

For the first and second prototypes, the low-level evaluation is performed by determining the accuracy of the classifier in correctly classifying unseen segmented regions into the corresponding classes.

3.8.2. Low-level evaluation of the final prototype

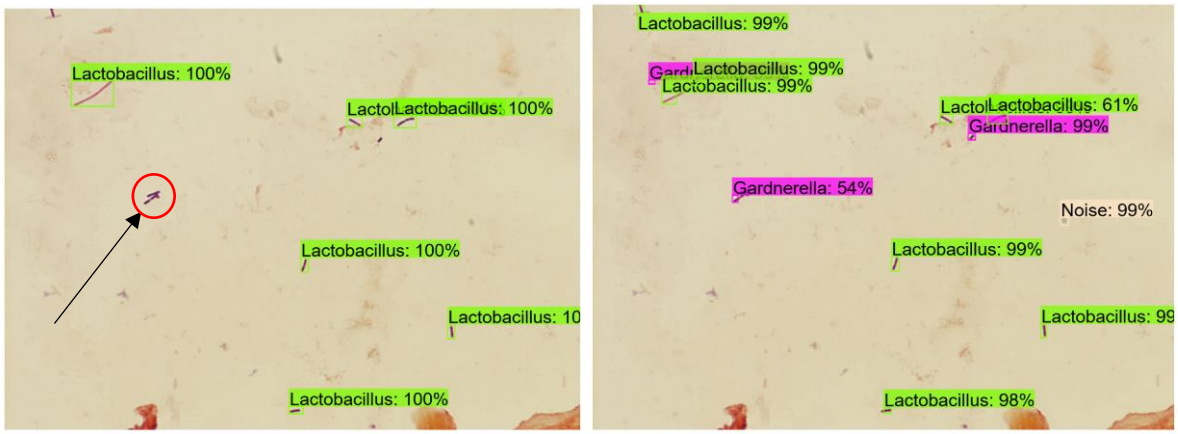

Figure 22. A comparison between the ground truths and predictions of an unseen image (Left: Ground truth bounding boxes; Right: Predicted bounding boxes)

For the final prototype, since the detection pipeline with Faster R-CNN is used, different metrics should be used.

Due to the fact that not all bacteria in the smear images were labelled by the technicians (see Figure 22, where the bacteria pointed by the arrow is not labelled), it is inappropriate to penalise the model when it predicts a region which is not labelled (the corresponding predicted with Gardnerella in Figure 22). Hence, the low-level evaluation metrics is taken to be the accuracy of the classifier correctly classifying the regions which are labelled. A region predicted by the classifier and a region labelled are said to be overlapping if the Intersection over Union (IoU) (the ratio between the area of the two regions' intersection and the area of the two regions' union) is at least 0.5.

3.8.3. High-level evaluation of all prototypes

Apart from measuring the performance on classifying each region, the overall performance in correctly determining the degree of infection (Normal, Intermediate and Infected), as well as the performance in estimating the Nugent Score (from 0 to 10) [4] of one full image are also measured.

4. Results

4.1. Work completed

4.1.1. Data collection

The first batch of images was obtained during meetings held in March and April 2017. This includes 40 bacterial colony images and 31 vaginal smear images. The smear images are annotated with high-level interpretations, including Nugent Score and overall degree of infection only.

The second batch of images was obtained during meetings held in October 2017. This batch consists of 119 vaginal smear images, all annotated with high-level interpretation. After the development of the data labelling tool, the collection of detailed point labelling started in late October 2017 and finished in mid-December 2017. The dataset was later augmented from point labels to bounding box labels in March 2018.

4.1.2. Data Labelling Tool

The development of the data labelling tool, *Clicklable*, was completed and distributed to the medical experts in October 2017. The feedback from the medical professionals was satisfactory.

A second version of the tool was developed in March 2018 with the additional functionalities of marking bounding boxes, which allowed the data augmentation.

4.1.3. Diagnostic system

The first phase of the project, which involved the development of the first prototype of the diagnostic system was completed in September 2017. Basic image segmentation tools were then developed, and the auxiliary programs for training the model learning models were also adapted for the project. The first classifier prototype for the type of bacteria was trained and evaluated.

The second phase of the project, which included the review of the scope of the project, the development of the second prototype and streamlining the auxiliary tools, was finished in late January 2018.

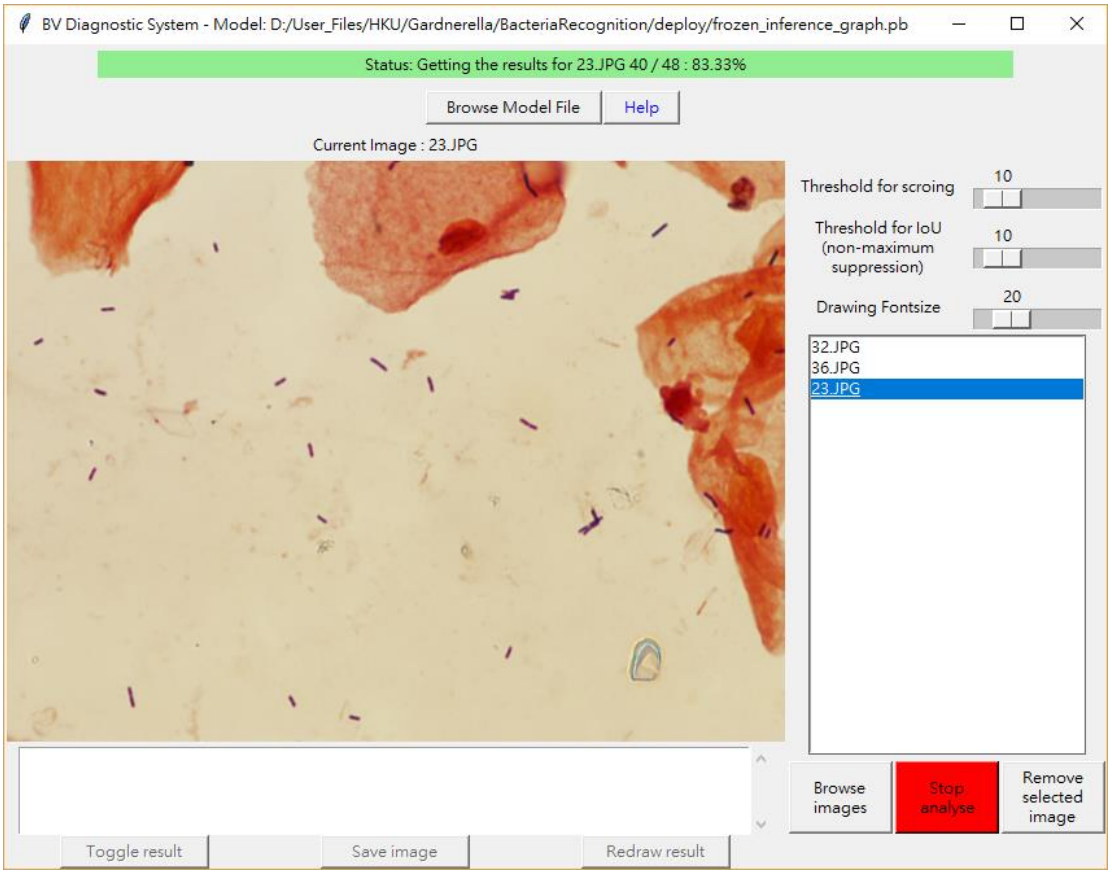

Figure 23. The main user-interface of the *BV Diagnostic System*

The final phase of the project was finished in April 2018. In this phase, a new way of tackling the problem with state-of-the-art deep learning algorithms was explored, which led to the development of the final prototype. A deployed diagnostic system was completed (see Figure 23).

The tool has a relatively simple user interface, where the user can choose which machine learning model (the model used in the final prototype is provided) to use for diagnosis. The user can then select the images to be analysed and click the “Analyse selected image” button to start the diagnosis process. The entire diagnosis process is automated and monitored, where the user can track the progress of the diagnosis in the status bar (see Figure 23).

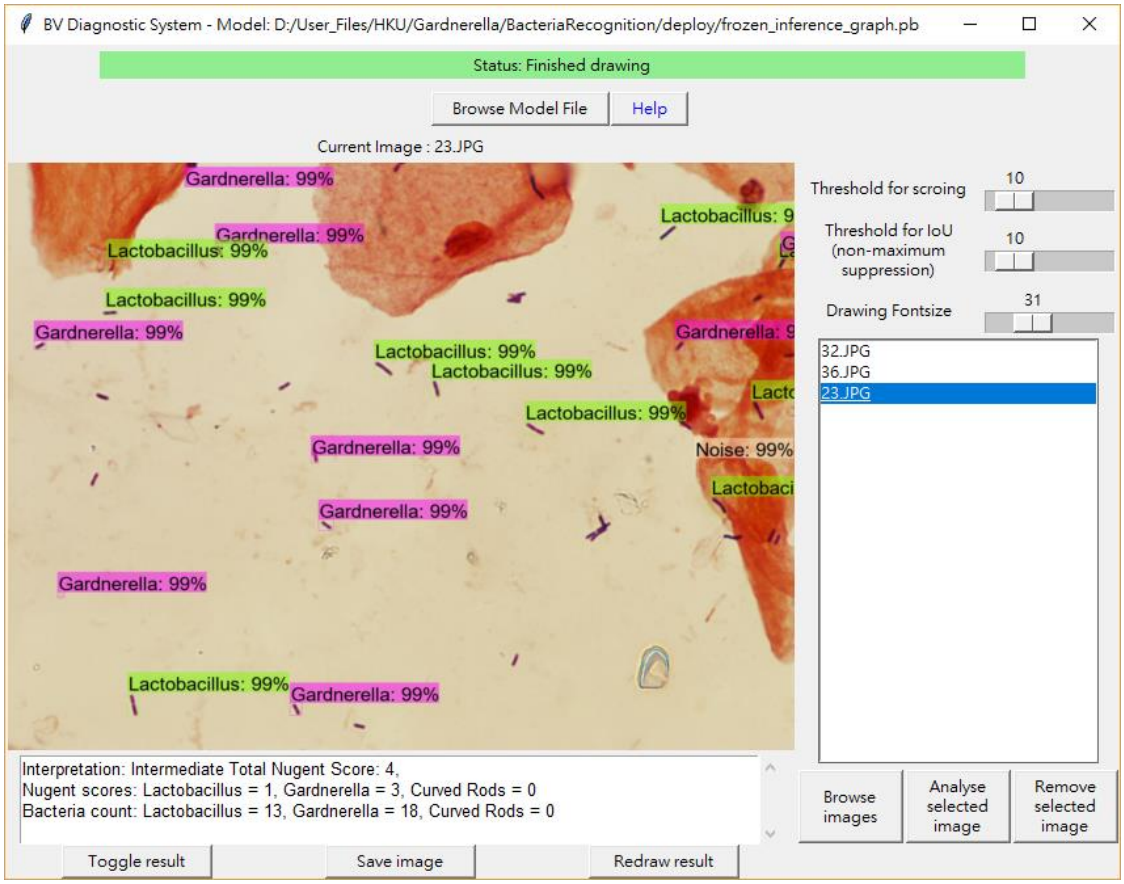

Figure 24. The results screen of the *BV Diagnostic System*

After the analysis, the results are visually presented to the user (see Figure 24), where the detected regions are annotated accordingly, with the type of bacteria and the confidence of the predictions. A certain level of customizability is available to the user where the threshold for minimum confidence to be counted as a valid prediction, and visualization size can be adjusted. The results can be turned on and off, and export to a digital image for further investigation. A text summary of the diagnosis is also provided at the bottom, with both high-level and low-level evaluation metric including the degree of infection, the Nugent Score, and the number of bacteria detected. Further improvements to the diagnostic tool including a higher degree of customization available could potentially improve the user experience.

4.2. Project Schedule

This project is separated into three phases, each with important deliverables and milestones at completion.

4.2.1. Schedule of phase 1

Table 3. The project schedule for phase 1

Item	Finished on
Training of the first batch of machine learning models and the analysis of their performance	29 September 2017
Submission of detailed project plan and the construction of project web page	1 October 2017
Meetings with medical experts to get feedback for the performances of the models	14 October 2017

4.2.2. Schedule of Phase 2

Table 4. The project schedule for phase 2

Item	Finished on
Further investigation into image processing modules	14 November 2017
Collection of new data from the medical experts	14 December 2017
Development of the second prototype based on new data and new image processing techniques	25 January 2018
First presentation	29 January 2018

4.2.3. Schedule of Phase 3

Table 5. The project schedule for phase 3

Item	Finished on / Finish by
Exploring deep learning algorithms in object detection	25 February 2018
Augmentation of data to higher level details	14 March 2018
Development of the final prototype using the augmented data and new algorithms	7 April 2018
Final fine-tuning of the integrated diagnosis system	12 April 2018
Final presentation	24 April 2018
Project exhibition	2 May 2018

4.3. Performance of the first diagnostic tool prototype

The first batch of data collected consists of 71 images, and out of all images, 31 of those, which were collected from the patients, the Nugent Score is also available for them and hence are used for testing. All images were directly segmented using MSER. The segmented images which originated from the bacterial colonies were used as the training and evaluation data for the machine learning model, with 30% of the data reserved for validation. All the remaining segmented images, which were originally collected from the patients, were then used as testing data. These images were not used for training nor validation due to the lack of detailed labelling of individual bacteria in the patient images. The first classifier prototype was trained using mini-batch gradient descent (each with 16 samples) with momentum for around 30000 steps and fine-tuned according to the performance on the validation set.

Table 6. Validation results on the type of bacteria in segmented images of the first classifier prototype

		Predicted type of bacteria by the model			
		Lactobacillus	Gardnerella	Curved rods	Other
Actual type of bacteria	Lactobacillus	**499**	6	1	2
	Gardnerella	0	**559**	3	4
	Curved rods	0	2	**589**	0
	Other	2	5	0	**63**

The first classifier prototype has an accuracy of 98.6% in the validation stage, with the detailed performance shown in Table 6. Out of the 1735 segmented regions within the validation set, 1710 were accurately classified by the model. Some confusion between bacteria types is observed, for example, there are 6 segmented images of bacteria type Lactobacillus wrongly classified as Gardnerella by the model (as shown in the cell by the column “Gardnerella” and row “Lactobacillus” in Table 6), but overall this is a highly satisfactory performance which indicates that the model generalizes well for bacteria found in the colony images.

Table 7. Testing results on the degree of infection of the first classifier prototype

		Estimated degree of infection by the model		
		Normal	Intermediate	BV Infection
Actual degree of infection	Normal	**8**	0	0
	Intermediate	5	**3**	0
	BV Infection	6	6	**3**

However, in the final evaluation using patient images, the model only achieved an accuracy of 45.1% (see Table 7) for the high-level evaluation on the degree of infection. Out of the 31 patient images, only 14 of them are accurately estimated the degree of infection. For the remaining 17 patient images, all of them are underestimated by the model.

Table 8. Testing results on Nugent Score of the first classifier prototype

		Estimated Nugent Score by the model										
		0	1	2	3	4	5	6	7	8	9	10
Actual Nugent Score	0	**0**	0	0	0	0	0	0	0	0	0	0
	1	0	**2**	1	0	0	0	0	0	0	0	0
	2	0	0	**0**	0	0	0	0	0	0	0	0
	3	2	1	0	**0**	0	0	0	0	0	0	0
	4	0	1	1	1	**0**	0	0	0	0	0	0
	5	0	1	1	1	0	**0**	1	0	0	0	0
	6	0	0	0	0	0	1	**0**	0	0	0	0
	7	0	0	0	2	1	0	0	**0**	0	0	0
	8	0	1	2	1	1	1	0	0	**1**	0	0
	9	0	1	0	0	0	1	0	0	0	**0**	0
	10	0	0	0	0	0	1	2	2	0	0	**0**

Furthermore, the model only achieved an exact-match accuracy of 9.6% (3 out of 31), an accuracy of 22.5% (7 out of 31) within an error margin of 1, and an accuracy of 32.3% (10 out of 31) within an error margin of 2 (see Table 8) for the high-level evaluation on Nugent Score. Almost all the estimations by the model are underestimations, which can be seen from the majority of values lying below the diagonal. It can be inferred that the model tends to underestimate the severity of the infection.

This drastic difference in performance indicates that the model does not generalize well to the smear images and may have the problem of overfitting.

4.4. Performance of the second diagnostic tool prototype

The second batch of data consists of 119 patient images, with 57 being diagnosed with infection, 38 intermediate, and 24 normal. There are 2 images with a Nugent Score of 0, 21 images with a score of 1, 1 image with a score of 2, 0 of them with a score of 3, 15 of them

with a score of 4, 10 of them with a score of 5, 13 of them with a score of 6, 17 of them with a score of 7, 16 of them with a score of 8, 22 of them with a score of 9, and 2 of them with a score of 10. There are in total 522 annotated locations of Lactobacillus, 2528 locations of Gardnerella and 246 locations of curved rods. All images were pre-processed and segmented using the adaptive Otsu's method. 99 out of 119 images are used for training. All the segmented images from the training set and the corresponding labels are used for training. The second classifier prototype was trained using the Adam optimization algorithm with mini-batch of 64 samples for around 12000 steps and fine-tuned according to the performance on the validation set.

Table 9. Validation results on the type of bacteria in of the second prototype

		Predicted type of bacteria by the model			
		Lactobacillus	Gardnerella	Curved rods	Other
Actual type of bacteria	Lactobacillus	**42**	10	0	0
	Gardnerella	10	**78**	7	2
	Curved rods	4	16	**5**	11
	Other	1	1	2	**143**

The second prototype achieved an accuracy of 80.7% (268 out of 332) in the validation stage, with the detailed performance shown in Table 9. A higher degree of confusion between bacteria types is observed, especially between Gardnerella and curved rods. Overall this is a satisfactory performance in the newly labelled dataset. It is important that this performance should be directly compared to that of the first prototype because in the first prototype, the validation is done on colony images where the distribution of data can be different from those found in patients.

Table 10. Testing results on the degree of infection of the second prototype on all images

		Estimated degree of infection by the model		
		Normal	Intermediate	BV Infection
Actual degree of infection	Normal	**15**	9	0
	Intermediate	5	**25**	8
	BV Infection	0	13	**44**

In the final evaluation using all patient images, the model achieved an accuracy of 70.6% (see

Table 10) for the high-level evaluation on the degree of infection. Out of the 119 patient images, 84 of them are accurately estimated the degree of infection. It is desirable that the model does not overestimate or underestimate the degree of infection by a large margin, which can be seen that none of the images which are labelled “BV infection” estimated to be “Normal” or labelled “Normal” estimated to be “BV infection”.

Table 11. Testing results on Nugent Score of the second prototype on all images

		Estimated Nugent Score by the model										
		0	1	2	3	4	5	6	7	8	9	10
Actual Nugent Score	0	**0**	0	1	1	0	0	0	0	0	0	0
	1	0	**3**	1	9	5	3	0	0	0	0	0
	2	0	0	**0**	0	1	0	0	0	0	0	0
	3	0	0	0	**0**	0	0	0	0	0	0	0
	4	0	0	1	2	**2**	3	4	2	1	0	0
	5	0	0	1	0	2	**4**	2	1	0	0	0
	6	0	0	0	1	0	1	**7**	3	1	0	0
	7	0	0	0	0	0	0	5	**7**	3	2	0
	8	0	0	0	0	0	0	1	8	**4**	3	0
	9	0	0	0	0	0	2	3	7	7	**1**	2
	10	0	0	0	0	0	0	2	0	0	0	**0**

For the high-level evaluation on Nugent Score, the model achieved an exact-match accuracy of 23.5% (28 out of 119), an accuracy of 58.8% (70 out of 119) within an error margin of 1, and an accuracy of 82.3% (98 out of 119) within an error margin of 2 (see Table 11).

It can be seen that the performance is higher compared to the first prototype, which indicates the new adopted model and the detailed dataset have improved model in generalizing to unseen data.

4.5. Performance of the final diagnostic tool prototype

The same batch of data as used in the second prototype (second batch) was used in the development of the final prototype. However, instead of annotated point locations of bacteria,

the labels are augmented to bounding boxes around the corresponding labels. There are in total 531 annotated locations of Lactobacillus, 2411 locations of Gardnerella and 232 locations of curved rods. Each image (of resolution 1280 x 960) is segmented into 165 160x160 smaller patches by the sliding window segmentation, where each time the window is moved 80 pixels in one direction. Similarly, 99 out of the original 119 images are used for training. All the segmented images from the training set and the corresponding labels are used for training. The final classifier prototype was then trained using mini-batch gradient descent (each with 8 samples) with momentum for around 22000 steps and fine-tuned according to the performance on the validation set.

Table 12. Validation results on the type of bacteria in of the final prototype

		Predicted type of bacteria by the model				
		Lactobacillus	Gardnerella	Curved rods	Other	Missed
Actual type of bacteria	Lactobacillus	**15**	21	0	0	12
	Gardnerella	1	**82**	1	1	123
	Curved rods	3	3	**0**	4	24
	Other	0	0	0	**0**	14

In this prototype, the evaluation metric is the calculated using only the regions which are labelled by the technicians (see Section 3.8.2). Among the areas which are also captured by the model, the accuracy of identifying the type of bacteria is 74.0% (97 out of 131) (see Table 12). However, the accuracy dropped to 33.4% (97 out of 290) if the missed labels (where the model does not give a prediction) of Lactobacillus, Gardnerella and Curved rods are taken into account. However, this metric again cannot be directly compared to the previous prototypes because in this model, the locations of the bounding boxes are also predicted by the mode, rather than relying on other segmentation algorithms.

Table 13. Testing results on the degree of infection of the final prototype on all images

		Estimated degree of infection by the model		
		Normal	Intermediate	BV Infection
Actual degree of infection	Normal	**15**	9	0
	Intermediate	0	**31**	7
	BV Infection	0	18	**39**

In the high-level evaluation on degree of infection using all patient images, the model achieved an accuracy of 71.4% (see Table 13). Out of the 119 patient images, 85 of them are accurately estimated the degree of infection. This model again has a desirable property that there is no overestimation or underestimation on the degree of infection by a large margin.

Table 14. Testing results on Nugent Score of the final prototype on all images

		Estimated Nugent Score by the model										
		0	1	2	3	4	5	6	7	8	9	10
Actual Nugent Score	0	**0**	0	1	1	0	0	0	0	0	0	0
	1	0	**0**	1	11	8	1	0	0	0	0	0
	2	0	0	**0**	1	0	0	0	0	0	0	0
	3	0	0	0	**0**	0	0	0	0	0	0	0
	4	0	0	0	0	**2**	4	8	1	0	0	0
	5	0	0	0	0	1	**3**	4	2	0	0	0
	6	0	0	0	0	1	2	**6**	3	1	0	0
	7	0	0	0	0	0	2	5	**4**	6	0	0
	8	0	0	0	0	0	0	7	6	**3**	0	0
	9	0	0	0	0	0	0	4	11	6	**1**	0
	10	0	0	0	0	0	0	0	2	0	0	**0**

In terms of Nugent Score estimation, the model achieved an exact-match accuracy of 16.0%

(19 out of 119), an accuracy of 48.7% (58 out of 119) within an error margin of 1, and an accuracy of 85.7% (102 out of 119) within an error margin of 2 (see Table 14).

It can be observed that the performance is similar to that of the second prototype, with slightly higher performance in identifying the degree of infection.

5. Limitations and mitigation strategies

5.1. Limited amount of data

The availability of images was relatively low in the development, where the total number of images is below 80 in the first phase, and below 200 in the later phases. This restriction on data makes it impractical to directly develop an end-to-end classifier using entire images.

To mitigate this problem, the images were segmented in all prototypes, and the machine learning models were chosen accordingly. Data augmentation techniques including random flipping, rotation and translation were also used to increase the variety of data.

5.2. Similarity among bacteria

Figure 25. Segmented area containing a bacterium type “Lactobacillus”

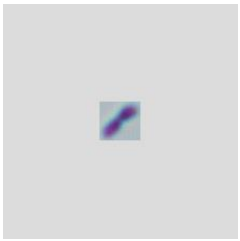

Figure 26. Segmented area containing a bacterium type “Gardnerella”

It is sometimes difficult to distinguish between the bacteria types Gardnerella and Lactobacillus (see Figure 25, Figure 26) due to the similarity of morphology and variations in staining. The medical experts who provided us with the data also confirmed the ambiguity. This limitation could be the main source of inaccuracy in prediction in all prototypes developed, which could potentially undermine the feasibility of developing a highly accurate automated diagnostic system.

In the final phase of the project, an object detection algorithm was selected such that the surroundings of the bacteria are also used as input to the models, aiming to reduce the effects of such similarity.

5.3. High variation in smear images

As the smear images are obtained in different batches and are prepared by hand, there are sometimes significant variations in the degree of staining and level of illumination.

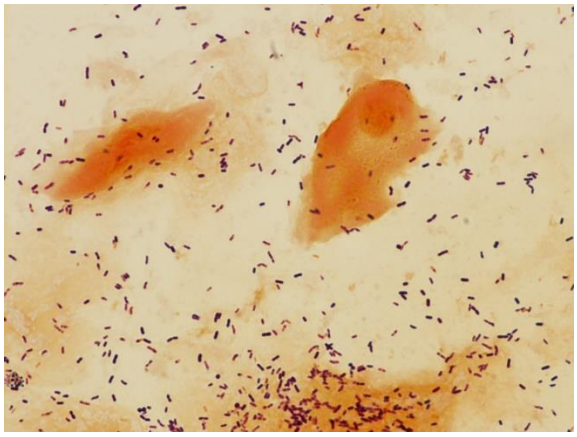

Figure 27. A patient smear image in image batch 1

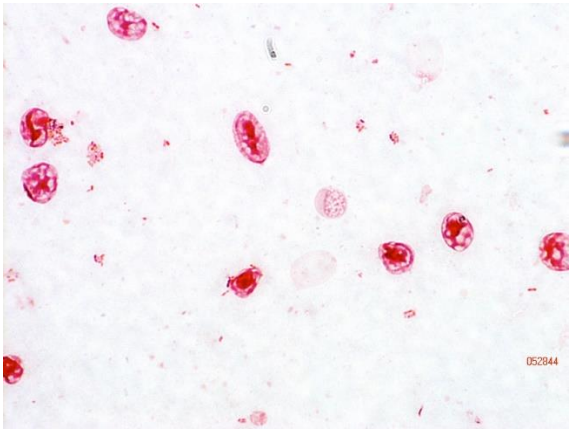

Figure 28. A patient smear image in image batch 2

Among some of the images, the variation is very noticeable (see Figure 27, Figure 28), where the levels of staining are significantly different.

Image processing algorithms including spatial filtering and smoothing were applied in later prototypes to mitigate this issue. However, certain Gram-negative bacteria including curved rods might be removed by image processing because of their pale colours. As a result, only the algorithms with mild effects on Gram-negative bacteria were used.

5.4. Discrepancies in morphology in different environments

A noticeable difference in morphology of the same type of bacteria in different environments (colony and patient) is observed. This could be the explanation for the contrasting difference in performances in the first prototype, because it was trained only using the colony images.

This problem was tackled in the second phase of the project, where a larger, more detailly labelled set of images was sought from the medical experts. The difference in performance between validation and testing was then significantly narrowed.

5.5. Incomplete labelling of data

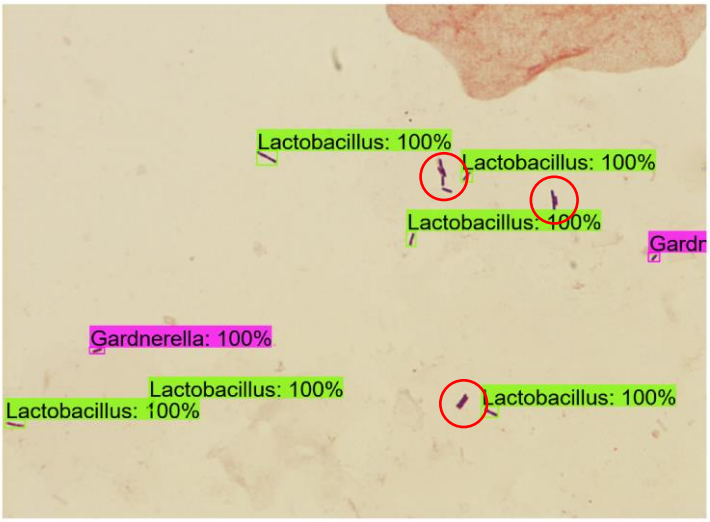

Figure 29. The ground truth labelling for an image (unlabelled areas are circled)

Although a more detailly labelled dataset was obtained, most images are not fully labelled (see Figure 22, Figure 29), where there are many bacteria instances without any label. In many cases, it cannot be determined whether a region, without any ground truth label, recognised by the model as a certain type of bacteria is a false-positive or true-positive. This made it difficult to perform an evaluation on the models and fine-tuning.

Furthermore, in the final prototype, since a Faster R-CNN architecture is used, it is necessary in the training process to suppress false-positives because it is common to allow the model to propose a large number of regions to accommodate the complexity of the problem. In this project, the penalty for false-positives (determined by whether the predicted label is wrong or there is no label in the region) is reduced by lowering the fraction of false-negatives samples in the learning process. However, this strategy is not perfect for such problem, and there might still be cases where a correctly identified region is penalised due to the lack of labels.

5.6. Difference between real-life diagnosis and this project

According to the medical professionals, in real-life diagnosis, it is typical to examine 10 to 20 smear images before making a final diagnosis. However, in this project, each image is taken as a standalone sample and an estimation of degree of interpretation is performed.

This problem is amplified because of the sensitivity of the Nugent Score system, where the difference in score determined when there is just one mis-classified sample could be as high

as 2. For example, a false-positive of the Gardnerella bacteria type could increase the score by 2 if there is originally no Gardnerella present in the image. Hence, a dataset with information about images from the same patient could be used to better reflect the performance of the models.

5.7. Imbalanced Dataset

From the information about the statistics of data given in Section 4.4 and Section 4.5, it can be observed that the numbers of labels annotated as Lactobacillus, Gardnerella and curved rods have a ratio of around 2:10:1. This imbalance in the dataset makes the models developed prone to bias in classifying regions to be Gardnerella. This can be observed from the low number of curved rods predicted in any region.

Selective data augmentation can be applied where certain labels are augmented more often than the others, however, due to the relatively small dataset, it could lead to overfitting problem where very similar data are used to train much more times than normal. More complex techniques including extra classification model for scarce classes might be required to solve this problem.

6. Future Development

As mentioned in Section 5, there are multiple problems and limitations that require further work to overcome.

First of all, in terms of data, a more complete dataset with full labelling would be the most beneficial to the object detection models. A larger and more balanced dataset could also allow the performance to improve. More samples in different staining and illumination could reduce the dependence on image processing techniques to accommodate such variance and let the model infer the generalized representation of the bacteria classes directly. Information across images from the same patient could be useful in better evaluating the performance of the models.

It is also possible to take an interactive approach to improving the quality and quantity of data by inviting the medical professionals to review the regions identified by the models which were not labelled any ground truth. This is because it can be observed that there are many such detections present in the prediction.

In addition, non-region-based object detection architecture could be used to tackle the problem. Algorithms including YOLO (“You only look once”) [42] and SSD (Single Shot MultiBox Detector) [43] do not rely on an underlying region proposal network, but rather tackle it as a regression problem. These architectures might pose new challenges and allow better understanding of the nature of the data.

Furthermore, new deep learning architectures and learning algorithms including Capsule Network [44] and focal loss (RetinaNet) [45] which showed improvement over other architectures in object detection can be used to develop a better performing diagnostic tool.

With a higher availability of data and computational power, an end-to-end machine learning model might be practical where images without segmentation are used as input for training.

7. Conclusion

Diagnostic microscopy is the gold standard for the diagnosis of many infections, including bacterial vaginosis. However, the high costs and the dependence on human expertise in microscopic diagnosis are preventing it from becoming widely available. There have been successful cases of developing automated diagnostic tools for some common infections using machine learning or related techniques. This shows a promising possibility for the success in developing one for bacterial vaginosis.

This project explores the practicability and possibility of using machine learning and computer vision techniques in the development of such tool. In order to facilitate the development of similar tools in medical contexts, the limitations in the development process are discussed with proposed mitigation strategies, such that this project can be used as an example for future research.

Three diagnostic tool prototypes were developed, with the first one using standard convolutional neural networks and the first batch of data, second one using residual network and the second batch of detailly labelled data, and the final one using Faster R-CNN as the object detection algorithm on top of a residual network and bounding boxes annotated data.

Throughout the three phases of this project, an increase in performance of the diagnostic tool prototypes was observed, where the testing accuracy of identifying the degree of infection increased from 45.1% to 70.6% and to 71.4%, and that of estimating the Nugent Score within an error margin of 2 increased from 32.3% to 82.3% and 85.7% for the three prototypes respectively.

The results have shown that the limitations of the dataset in terms of the amount of data, the balance of data and completeness of labelling could be the factors limiting the performance of the models.

Due to the restriction on the amount of data, this project did not tackle the classification task directly using whole blood smear images. With a higher availability of data, the approach of using blood smear images as a whole might be possible in future research, and the difference in terms of performance and complexity could be explored. Also, different object detection and machine learning algorithms and architectures, including YOLO, SSD, Capsule Network and focal loss, have shown new possibilities in tackling this problem, which could be applied to further improve the diagnostic tool.

8. References

[1] C. Kenyon, R. Colebunders and T. Crucitti, “The global epidemiology of bacterial vaginosis: a systematic review,” *American Journal of Obstetrics and Gynecology,* vol. 209, no. 6, pp. 505-523, 2013.

[2] E. H. Koumans, M. . Sternberg, C. . Bruce, G. M. McQuillan, J. S. Kendrick, M. Y. Sutton and L. E. Markowitz, “The prevalence of bacterial vaginosis in the United States, 2001-2004; associations with symptoms, sexual behaviors, and reproductive health,” *Sexually Transmitted Diseases,* vol. 34, no. 11, pp. 864-869, 2007.

[3] J. Atashili, C. Poole, P. Ndumbe, A. Adimora and J. Smith, “Bacterial vaginosis and HIV acquisition: a meta-analysis of published studies,” *AIDS,* vol. 22, no. 12, pp. 1493-1501, 2008.

[4] R. P. Nugent, M. A. Krohn and S. L. Hillier, “Reliability of diagnosing bacterial vaginosis is improved by a standardized method of gram stain interpretation,” *Journal of clinical microbiology,* vol. 29, no. 2, pp. 297-301, 1991.

[5] B. Sha, H. Chen, Q. Wang, M. Zariffard, M. Cohen and G. Spear, “Utility of Amsel Criteria, Nugent Score, and Quantitative PCR for Gardnerella vaginalis, Mycoplasma hominis, and Lactobacillus spp. for Diagnosis of Bacterial Vaginosis in Human Immunodeficiency Virus-Infected Women,” *Journal of Clinical Microbiology,* vol. 43, no. 9, pp. 4607-4612, 2005.

[6] R. Chawla, P. Bhalla, S. Chadha, S. Grover and S. Garg, “Comparison of Hay’s Criteria with Nugent’s Scoring System for Diagnosis of Bacterial Vaginosis,” *BioMed Research International,* vol. 2013, pp. 1-5, 2013.

[7] World Health Organization., “Microscopy,” 14 March 2017. [Online]. Available: http://www.who.int/malaria/areas/diagnosis/microscopy/en/. [Accessed 27 September 2017].

[8] K. C. Hazen., “Microscopy,” October 2016. [Online]. Available: http://www.merckmanuals.com/professional/infectious-diseases/laboratory-diagnosis-of-infectious-disease/microscopy. [Accessed 13 February 2017].

[9] J. Bennett, Mandell, Douglas, and Bennett's principles and practice of infectious diseases, Philadelphia: Elsevier/Saunders, 2015.

[10] J. A. Quinn, R. Nakasi, P. K. Mugagga, P. Byanyima, W. Lubega and A. Andama, "Deep convolutional neural networks for microscopy-based point of care diagnostics," in *Machine Learning for Healthcare Conference*, 2016.

[11] O. Kraus, J. Ba and B. Frey, "Classifying and segmenting microscopy images with deep multiple instance learning," *Bioinformatics,* vol. 32, no. 12, pp. i52-i59, 2016.

[12] F. Tek, A. Dempster and I. Kale, "Computer vision for microscopy diagnosis of malaria," *Malaria Journal,* vol. 8, no. 1, p. 153, 2009.

[13] D. Steinkraus, I. Buck and P. Simard, "Using GPUs for machine learning algorithms," in *Eighth International Conference on Document Analysis and Recognition*, 2005.

[14] R. Collobert, S. Bengio and J. Mariéthoz, "Torch: a modular machine learning software library," IDIAP, 2002.

[15] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard and M. Kudlur, "TensorFlow: A System for Large-Scale Machine Learning," in *OSDI*, Savannah, 2016.

[16] F. Chollet and others, "Keras," GitHub, 2015.

[17] V. Sessions and M. Valtorta, "The Effects of Data Quality On Machine Leraning Algorithms," in *Conference: Proceedings of the 11th International Conference on Information Quality*, Cambridge, MA, USA, 2006.

[18] MathWorks, "Introducing Machine Learning," 2016. [Online]. Available: https://www.mathworks.com/content/dam/mathworks/tag-team/Objects/i/88174_92991v00_machine_learning_section1_ebook.pdf. [Accessed 30 November 2017].

[19] Oracle, "Go Java," 29 August 2016. [Online]. Available: https://go.java/index.html. [Accessed 30 November 2017].

[20] SQLite, "SQLite Home Page," 2017. [Online]. Available: https://sqlite.org/index.html. [Accessed 30 November 2017].

[21] A. Durville, “Computer Vision with Convolution Networks,” 11 February 2017. [Online]. Available: https://github.com/OKStateACM/AI_Workshop/wiki/Computer-Vision-with-Convolution-Networks. [Accessed 23 October 2017].

[22] S. Damelin and W. Miller, The Mathematics of Signal Processing, Cambridge University Press, 2011.

[23] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on pattern analysis and machine intelligence,* vol. 6, pp. 679-698, 1986.

[24] P. Maragos, “Pattern spectrum and multiscale shape representation,” *IEEE Transactions on pattern analysis and machine intelligence,* vol. 11, no. 7, pp. 701-16, 1989.

[25] J. Matas, O. Chum, M. Urban and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions.,” *Image and vision computing,* vol. 22, no. 10, pp. 761-767, 2004.

[26] M. Ester, H. P. Kriegel, J. Sander and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” *Kdd,* vol. 96, no. 34, pp. 226-231, 1996.

[27] N. Otsu, “A threshold selection method from gray-level histograms.,” *IEEE transactions on systems, man, and cybernetics,* vol. 1, no. 62-66, p. 9, 1979.

[28] R. F. Moghaddam and M. Cheriet, “AdOtsu: An adaptive and parameterless generalization of Otsu's method for document image binarization.,” *Pattern Recognition,* vol. 6, no. 2419-2431, p. 45, 2012.

[29] B. Yegnanarayana, Artificial neural networks, PHI Learning Pvt. Ltd., 2009.

[30] Stanford University, “CS231n Convolutional Neural Networks for Visual Recognition,” 28 November 2017. [Online]. Available: http://cs231n.github.io/neural-networks-1/. [Accessed 30 November 2017].

[31] L. Yann, “LeNet-5, convolutional neural networks,” [Online]. Available: http://yann.lecun.com/exdb/lenet/. [Accessed 21 October 2017].

[32] MathWorks, “Convolutional Neural Network,” 2017. [Online]. Available:

https://www.mathworks.com/discovery/convolutional-neural-network.html. [Accessed 23 October 2017].

[33] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences,* vol. 44, no. 1, pp. 1-12, 2004.

[34] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *IEEE conference on computer vision and pattern recognition,* 2016.

[35] I. Goodfellow, Y. Bengio, A. Courville and Y. Bengio, Deep learning, vol. 1, Cambridge: MIT press, 2016.

[36] A. Karpathy, "CS231n Convolutional Neural Networks for Visual Recognition - Convolutional Neural Networks (CNNs / ConvNets)," Stanford University, 24 May 2017. [Online]. Available: http://cs231n.github.io/convolutional-networks/. [Accessed 14 April 2018].

[37] M. Lin, Q. Chen and S. Yan, "Network In Network," arXiv preprint arXiv:1312.4400., 2013.

[38] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems,* 2015, pp. 91-99.

[39] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2014, pp. 580-587.

[40] R. Girshick, "Fast r-cnn," arXiv preprint arXiv:1504.08083, 2015.

[41] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *Ijcai,* vol. 14, no. 2., pp. 1137-1145, 1995.

[42] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2016, pp. 779-788.

[43] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu and A. C. Berg, "Ssd: Single shot multibox detector.," in *European conference on computer vision,* Springer,

Cham, 2016.

[44] S. Sabour, N. Frosst and G. E. Hinton, “Dynamic routing between capsules.,” in *Advances in Neural Information Processing Systems*, 2017, pp. 3859-3869.

[45] T. Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, “Focal loss for dense object detection.,” arXiv preprint arXiv:1708.02002, 2017.