Addis Ababa University

College of Natural Sciences

Developing a computer-aided diagnosis model for TB using region-based convolutional neural network

Ibrahim Muse Ibrahim

A Thesis Submitted to the Department of Computer Science in Partial Fulfilment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

November 2020

Addis Ababa University

College of Natural Sciences

Ibrahim Muse Ibrahim

Advisor: Ayalew Belay (Ph.D.)

This is to certify that the thesis prepared by Ibrahim Muse, titled: *Developing a computer-aided diagnosis model for TB using region-based convolutional neural network* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Name	Signature	Date
Advisor: Ayalew Belay (Ph.D.)		
Examiner: Dida Midekso (Ph.D.)		
Examiner: Yaregal Assabie (Ph.D.)		

Abstract

Tuberculosis (TB) is an infectious disease caused by the bacteria Mycobacterium tuberculosis or simply M. tuberculosis. It is primarily an infection of the lungs, but it can also affect other parts of the body. TB is one of the leading causes of death in developing countries, although most are preventable if diagnosed early and treated. Among the available tools, Sputum smear microscopy is widely used for TB diagnosis.

Manual TB screening is tedious work and prone to error due to workload and a dearth of properly trained technicians, manual recognition of the bacillus from the microscopic image takes a long time and requires expert handling of the equipment for the TB identification.

To overcome the manual detection issues and develop an automatic TB diagnosis model, we used deep neural networks. We proposed an automatic TB diagnosis and segmentation model composed of Mask R-CNN, Hungrain Algorithm, and Hard example mining for the microscopic image. The proposed model works in a sequential manner where it first detects, classifies, and segments the bacillus objects then the Hungarian Algorithm and Hard example mining is used to further enhance the performance and overcome the problem of high False Positive rate.

We carried out experiments to evaluate the performance of our proposed model, we used the metrics of recall, precision, and F-score. We collected the sputum images ZNSM-iDB dataset which is publicly available dataset in the internet and used it for both training and testing. Our experimental results show values of 99.25%, 91.04%, 94.96% for recall, precision, and f-score respectively. which is a significant improvement by the proposed approach compared to existing methods, thus helping in more accurate disease diagnosis.

Keywords: Instance Segmentation, Computer Aided Diagnosis, TB Detection.

Acknowledgments

I would like to thank my parents for their harsh criticism and motivation. Then I would like to thank Dr. Ayalew Belay, for supervising my thesis, I am especially grateful for the freedom he gave me during this thesis, his attentive guidance, invaluable trust, and constructive criticism.

Table of Contents

List of tables

List of figures

Acronyms/Abbreviations

AFB	Acid-fast bacilli
CDC	Centre for Disease control
CNN	Convolutional Neural Network
EPTB	Extra-Pulmonary TB
FCN	Fully Connected Network
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Network
GPU	Graphical Processing Unit
MTB	mycobacterium tuberculosis complex
NMX	Non-Max Suppression
NTM	Non-tuberculin mycobacteria
PTB	Pulmonary Tuberculosis
R-CNN	Region-Based Convolutional Neural Network
RELU	Rectified Linear Unit Layer
ROI	Region of Interest
RPN	Region Proposal Network
TB	Tuberculosis
TP	True Positive
VGG	Visual Geometry Group
WHO	World Health Organization
ZN	Ziehl-Neelsen

Chapter 1: Introduction

1.1 Background

Tuberculosis is an airborne infectious disease caused by bacterial microorganisms called Mycobacterium tuberculosis which is harmful bacteria that can be fatal to humans, and animal lives [1]. Tuberculosis poses a large problem in low-income countries and developing countries, it is the single culprit behind most of the deaths among individuals aged fifteen to forty-nine years [2]. It mainly affects the lungs although it can affect many other organs throughout the body. Since it is an airborne disease it mainly spreads through the air, when an infected patient coughs, sneezes, or transmits their saliva through the air [3].

Common symptoms of active lung TB are cough with sputum and blood at times, chest pains, weakness, weight loss, fever, and night sweats. Many countries still rely on a long-used method called sputum smear microscopy to diagnose TB. Trained laboratory technicians look at sputum samples under a microscope to see if TB bacteria are present [4].

In many developing countries, air pollution from industrial growth, automobiles, dust, high population density, and poor sanitary conditions provide an environment for the spread of airborne diseases and causes the local population to be affected with various other kinds of serious and dangerous respiratory diseases. Even worse, most of the affected people who fall prey to these diseases are from a poor background, live in congested lodgings, and lead to the spread of these diseases. In cases, when Tuberculosis is either diagnosed late or not diagnosed at all, it could be fatal [5]. The detection of bacteria is very important to prevent the disease and maintain the health of the world's population against Tuberculosis [1].

According to a report from the World Health Organization [4], Tuberculosis is the world's top infectious killer today. TB is the ninth leading cause of death worldwide and the leading cause of a single infectious agent, ranking above HIV/AIDS. Over 25% of TB deaths occur in the African Region.

According to a report from the World Health Organization [6], at the beginning of infection of TB, symptoms could be mild for many months. This can lead to delays in seeking care and results in the transmission of the disease to others. People with active TB can transmit the disease to another 10–15 people over a year. Without proper treatment, 45% of HIV-negative people with TB on average and nearly all HIV-positive people with TB will die, the overwhelming majority of deaths in the World Health Organization African Region are caused by HIV/AIDS, malaria, and tuberculosis (TB). Along with neglected tropical diseases (NTDs), these diseases are diminishing the quality of life of individuals and affect entire countries' ability to develop safer societies and communities.

While a sputum smear is a simple inexpensive test, manual TB screening is tedious work and prone to error due to workload and a scarcity of properly trained technicians. Technicians view the smears slides with microscopes, looking for rod-shaped objects that may be Mycobacterium tuberculosis which is the bacteria responsible for TB disease. However, they may diagnose a positive TB slide as smear-negative because of the sparseness of acid-fast bacilli, or because too few fields have been examined. This often leads to low recall rates. Automatic methods are the best solution to improve the low sensitivity of TB diagnosis, reduce human variability in slide analysis, and speed up the screening process [7]. Conventional microscopy is mostly used in low and middle-income countries where the TB prevalence rate is high because it is less expensive, easier to use and maintain [8]. As such, in this thesis, we will focus on this kind of microscopy.

1.2 Motivation

Tuberculosis (TB) is the most lethal infectious disease and it is difficult to detect. Fortunately, TB is not an automatic death sentence, with the right, effective and early diagnosis it is possible to treat, control, and overcome the disease [9].

Recent years have also seen a scary rise in cases of TB World health organization estimated that there were 200,000 new TB cases and Ethiopia ranked the 10th of the world's 22 high burden countries for TB, and second after Nigeria in Sub-Saharan Africa [6]. Poor diagnosis or no diagnosis leads patients to remain in their communities for longer periods and transmit and create more TB patients. Even after diagnosis, because there are few diagnostic and treatment facilities and a lack of trained health professionals and drugs, patients do not start treatment immediately [10].

The rise of innovative approaches such as artificial intelligence deep learning and mobile health had a promising effect and become very efficient in the span of the last few decades. An artificial intelligence approach to process TB sputum test for diagnosis and detection of the existence of TB from sputum in a quick and cost-effective way could reduce workload and increase accuracy and help with the early diagnosis and treatment and control of TB in developing countries. Thus, we are motivated to develop a model for the diagnosis of TB using deep learning.

1.3 Statement of the problem

All regular lung diseases have nearly fundamentally the same symptoms and subsequently, are all hard to recognize and differentiate them from each other. They are brought about by bacteria and their diagnoses require tedious and costly examination, these diseases require rather quick medicinal treatment to counter serious consequences [11].

Even if a person has symptoms of TB, it is often difficult to diagnose it, and it is particularly difficult to diagnose rapidly. Rapid diagnosis is what is needed to provide and administer effective TB treatment.

According to a report from the World Health Organization [4], in 2018, 55% of pulmonary cases were bacteriologically confirmed, a slight decrease from 56% in 2017 despite increases in TB notifications. There is still a large gap between the estimated number of incident cases (9.0–11.1 million globally in 2018) and the number of new cases reported (7.0 million), due to a combination of underreporting of detected cases and underdiagnosis.

Underdiagnosed can happen because of several reasons such as poor geographical and financial access to health care, limited or lack of symptoms that might lead to delay in seeking of health care, failure to test for TB when people seek health care from health facilities, and diagnostic tests that are not sufficiently sensitive or specific to ensure accurate identification of all cases [4].

Manual recognition of the bacillus from the microscopic image takes a long time and requires expert handling of the equipment for the TB identification [11]. Another problem associated with the TB identification procedure is the separation of the overlapping bacilli. Since the severity of the TB disease is determined through the bacilli count, segmentation of the bacilli is required. The overlapping bacilli need to be separated and all bacillus found, segmentation of the overlapping bacilli is required. Estimation of the size and the state of the bacilli without the segmentation of the overlapping bacilli leads to mistakes in detection [12].

The segmentation of the bacilli has a complex process, the bacilli features such as shape are not discriminatory enough due to the variances in the features and sharing the same features with debris and other particles. Because of this, the bacillus doesn't have a certain and uniform shape across all the occurrences where several objects despite being bacilli, have a totally different shape that it can be confused with other objects. Furthermore, The regions in the sputum image that represent the background and bacillus share the same color, intensity, and textures so it is extremely difficult to segment the background and the bacillus [13].

In many cases, the bacillus is faint, occluded, and obscured by cells or debris and remnants, or inside macrophages, this gives the bacillus a hazy outline, which could lead it to be overlooked in the detection. Furthermore, the bacillus in the sputum image is so numerous sometimes that they become smushed together and superimposed and indistinguishable from each other [14].

In recent years, researchers [13,14,16] have applied deep learning methods that allow learning discriminative features from bacilli for detection and classification. Convolutional Neural Network (CNN) is the main engine for all these methods. In [13,14 ,16], the authors splitted the microscopic image into smaller patches, each containing an image object that could potentially be a TB bacillus. The CNN operated on patches (not the whole image) of the used datasets. The main drawback of these methods were the methods of splitting the larger microscopic image into such away. The accuracy of the method was largely dependent on the patching step. Some authors did not even reveal the details of how the patching was done [13 ,16], whether it is automated or even done manually.

The recent work of Panicker et al [18] tried to overcome this drawback by using an initial stage of image binarization and pixel classification to locate foreground objects (bacilli, non-bacilli, artifacts) and then construct the required patches. Each patch presumably containing one foreground object and then it is fed to the CNN stage for final classification into bacilli and non-bacilli. While this method automates the image patching, its overall accuracy depends on the success of the first binarization/pixel-classification step, which is often error-prone for the

challenging conventional bright-field microscopy. It also suffers from touching foreground objects and over-stained images.

All the previous methods have combined the workflow of image processing techniques, machine learning, and manual intervention for TB identification. The researchers heavily relied on hand-crafted sets of color shape descriptors for that goal and extraction of patches, which created a need for a laboratory technician expert for the extraction of the feature and resulted in rather low detection accuracy. There is a need for further study to improve the performance of the existing research by addressing some of the existing gaps. For instance, the manual batching generation and the exclusion of images with occluded and overlapping bacilli which would not be the case in the real world. It was therefore imperative to develop a new strategy for automatically detecting tuberculosis while putting into consideration all these deficiencies.

1.4 Objectives

General objective

The general objective of this thesis is developing a computer-aided diagnosis model for TB using region-based convolutional neural networks

Specific objectives

The specific objectives are: -

- Conduct literature review on TB and deep learning.
- Collect and gather the sputum image dataset.
- Select an appropriate deep learning method that is capable of efficient instance segmentation of TB.
- Develop a model for detecting Mycobacterium tuberculosis bacteria using the selected deep learning method.
- Develop a prototype for automatic detection of TB using the developed model.
- Evaluate the performance of the developed model.
- Compare the performance of the existing works to ours.

1.5 Methods

To complete and achieve the objectives of this thesis, the following methods will be used: -

- **Literature review**

A literature review is used to provide background and conceptual knowledge of the related proposal areas, including deep learning techniques best suited for big data and algorithms that could be used for image segmentation and classification, it also helps to learn about the available dataset for the domain and their accessibility.

- **Data collection**

Relevant and useful dataset for spear sputum that could be used to detect the Mycobacterium tuberculosis bacteria, train, and test the model will be collected.

The dataset will be collected from the publicly available dataset. The collected dataset will be divided randomly into training and test dataset. The training dataset is used to train the model and test data will be used for validation.

- **Prototype development**

To evaluate the performance of the proposed model, a prototype will be developed. Different appropriate tools will be selected and used to develop the prototype. We will try to reuse different existing methods, frameworks, and software components as much as possible.

- **Evaluation methods**

A prototype of the model will be developed, and the performance of the developed model will be evaluated through the prototype.

1.6 Scope and limitation

This thesis is mainly focusing on developing Computer-aided diagnosis (CAD) for TB to assist doctors in the interpretation of TB test images. Although various investigations can be used to help diagnose tuberculosis, for this thesis the TB testing techniques used will be only Sputum smear Microscopy images only from conventional microscopy.

The bacteria that cause tuberculosis (TB) are classified as mycobacteria. Many families of mycobacteria exist. Most of the types that do not cause tuberculosis (called non-tuberculosis mycobacteria) can cause infections in certain people, sometimes with symptoms similar to those of tuberculosis. Although many pulmonary diseases have similar symptoms and are caused by the same bacteria family, they have different classification techniques and different symptoms, this thesis only focuses on Tb diagnosis and detection of mycobacteria that cause pulmonary TB.

1.7 Application of study

This system would aid in the diagnosis of the different levels of tuberculosis. It can help doctors in the interpretation of TB test images. It could help patients and doctors alike with faster testing than the traditional testing way which could save time and money. It can facilitate the examination of a much larger number of patients at little extra cost, without additional technicians. This automated way can reduce fatigue by providing images on the screen and avoiding visual inspection of microscopic images.

1.8 Organization of the rest of the thesis

The rest of this thesis is organized as follows. In Chapter 2, we reviewed different literature works to gain a concrete understanding of TB diagnosis processing in general and TB deep learning diagnosis in particular. In Chapter 3, existing and recent research works that are related to TB diagnosis are investigated to understand how far the domain problem is explored so far and what are the limitations and gaps that need to be addressed. Then, a TB diagnosis model to solve all the discovered gaps in detail in Chapter 4 is presented. In Chapter 5, the experimental setup and results of the proposed model are presented in-depth. Finally, Chapter 6 concludes this thesis with recommendations and directions for future works.

Chapter 2: Literature reviews

2.1 introduction

An extensive literature review is carried out for this thesis work to get a deeper understanding of TB processing in general and TB diagnosis in particular including the morphological characteristics of TB and various TB features that carry relevant cues and information and their extraction methods and the different classification techniques and deep learning methods.

We try to provide an overview of the domain knowledge concepts related to our domain problem, we will review various relevant and related issues and concepts concerning healthcare and deep learning. We also focus on tools and technologies that have been used to develop the proposed model and implementation.

2.2 Tuberculosis

Tuberculosis (TB) is an infectious disease caused by a bacterium called Mycobacterium tuberculosis (M. tuberculosis). The genus Mycobacterium can be tuberculosis or non-tuberculosis mycobacterium. The mycobacterium tuberculosis complex(MTB) is the causative agent of TB in humans and animals, it is the major cause of human TB all over the world [19].

Mycobacterium tuberculosis is a large non-spore-forming, nonmotile, facultative rod straight, curved or bent shaped bacterium. The bacterium shape when it is straight rod-shaped is 2-4 micrometres in length and 0.2-0.5 um in width and when it is bent shaped it normally means multiple bacteria are touching or overlapping which will give it unique different shape and characteristics as shown in figure 2.1. It is an obligate aerobe. MTB complexes are always found in the well-aerated upper lobes of the lungs. The bacterium is an intracellular facultative bacteria, it has a slow generation time taking 15-20 hours, a characteristic that may add to its virulence [19]. Mycobacterium bacterium is impermeable by some dyes and stains which makes it acid-fast bacteria.

Mycobacterium bacterium is an acid-fast bacterium because it is impermeable to certain dyes and stains. Because of this, once stained, acid-fast bacteria will retain dyes when heated and

treated with acidified organic compounds. The most used acid-fast staining method for Mycobacterium tuberculosis is called Ziehl-Neelsen stain. When this method is used, the smear is stained using carbol-fuchsin (a pink dye), and then decolorized with acid-alcohol. The smear is counterstained with methylene-blue or certain other dyes. Acid-fast bacilli appear pink in a contrasting background that would normally appear in either red, blue, or green [19].

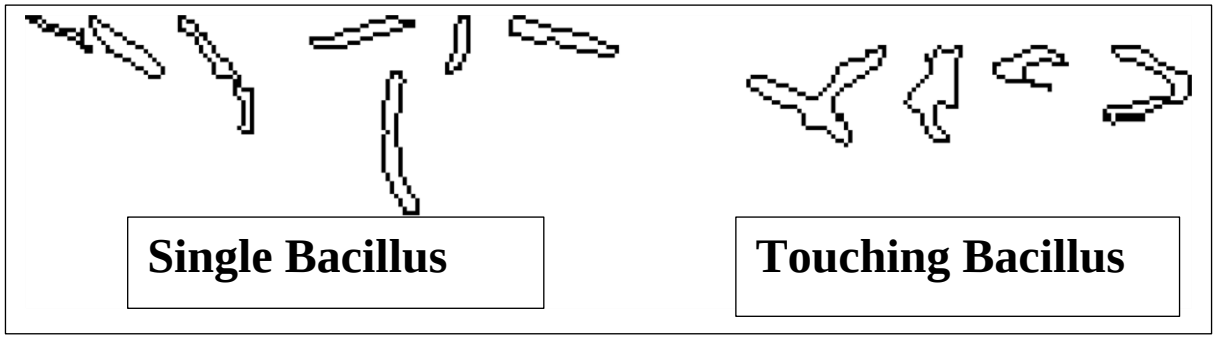


Figure 2.1: morphological structure of mycobacterium [20]

MTB bacteria is very difficult to detect by technicians as it takes a long time, technicians will look under a microscope for each field view for any time from 5-40 minutes. One of the biggest problems is how the bacteria doesn't have certain, specific and uniform shape among all the occurrence, it very difficult to divide bacteria into clusters because they don't share features where features such as shape, thickness, and length can vary among them. Figure 2.2 shows how variant the bacteria is in nature, several bacteria have a shape that can't be characterized as MTB and can be confused as debris. they can be straight, curved, or bent shaped [21].

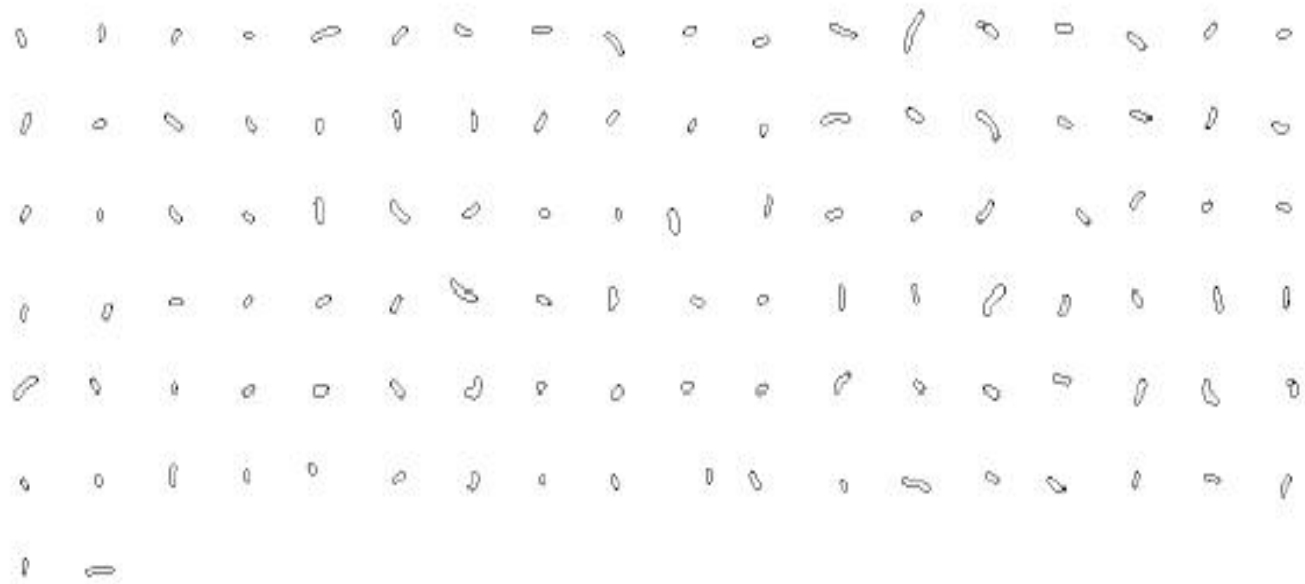


Figure 2.02: Morphological structure of mycobacterium [21].

There are two types of tuberculosis; pulmonary tuberculosis (PTB) and extrapulmonary tuberculosis (EPTB). Pulmonary TB is any bacteriologically confirmed or clinically diagnosed case of TB involving the lung parenchyma or the tracheobronchial tree. Extrapulmonary TB is tuberculosis that affects organs other than the lungs, such as lymph nodes, abdomen, genitourinary tract, skin, joints, bones, and meninges [22]

2.2.1 Clinical manifestation of tuberculosis

According to Raviglione [8], when someone is affected with PTB, there will be several symptoms the patient will show, Those symptoms include but not limited to:-

- coughing that lasts for 2 weeks or more.
- appetite and weight loss.
- sputum production.
- Haemoptysis.
- chest pain.
- breathlessness.

- and/or constitutional symptoms like fever, night sweats, tiredness, loss of appetite can also occur.

Those symptoms are used as suggestive clinical features and are essential for the diagnosis of PTB, after such symptoms are identified the patient is told to get tested [6]

2.2.2 Pathogenesis of pulmonary tuberculosis

When a healthy individual inhales bacillus, the first implant is done in the lungs at the bronchiole or alveolar level. The bacilli multiply and produce the primary lesion there. Some bacilli pass into the hilar lymph nodes causing lymph node enlargement [19].

The bacilli from the alveolar lesion, the Ghon focus, and the enlarged hilar lymph nodes can be more widely disseminated via the lymphatic system or bloodstream, leading to serious complications such as meningitis, bone joint, and renal tuberculosis [6]. The host response to tuberculosis is through cell-mediated immunity, and the cells involved include Macrophages and T lymphocytes. The lymphocytes recognize TB antigen and release cytokines such as gamma interferon, which activates macrophage at the site of the lesion [23].

Hypersensitivity to the organism appears at 8-10 weeks after the infection and the infected individual becomes tuberculin test positive. It is estimated that 10% of the infected individuals develop clinical tuberculosis during their lifetime. Around 50% of these will develop TB during the first year of infection and the rest many years later [24].

2.2.3 Diagnosis of TB disease

The existence of a dependable, quick, and affordable diagnostic process that is promptly available and accessible to the world is very important for the control and mitigation of TB.

Several diagnostic techniques have been utilized for the diagnosis of tuberculosis (TB), including both invasive and non-invasive procedures depending on the type of TB suspected. TB can be diagnosed by medical history, physical examination, chest x-ray, and microscopic laboratory tests.

- **Chest radiograph**

A chest radiograph is used to detect chest abnormalities in the posterior-anterior chest area. Lesions may appear anywhere within the lungs and may differ in size, shape, density, and cavitation. These abnormalities may suggest TB, but cannot be used to authoritatively diagnose TB. However, a chest radiograph may be used to rule out the possibility of pulmonary TB in a person who has had a positive reaction to a TB blood test and no symptoms of the disease. Chest radiology works adequately only when there is a high level of infection so this procedure cannot be used for TB diagnosis in the early stages [25].

- **Microbiology diagnostic**

When a patient is suspected to have active infection with tuberculosis, the patient is diagnosed by detecting the tuberculosis bacilli within the sputum smear under a microscope [24]. The diagnosis is done by examining the stained sputum smear. The clinicians regularly search for the presence of AFB in an amplified sputum smear microscopic image.

Three specimens of sputum are drawn from the patient on two consecutive days and stained with ZN staining procedure. Clinicians examine at least 100 fields of view and spend at least five full minutes for each field [26]. The presence of acid-fast-bacilli (AFB) on a sputum smear or other specimens indicates TB disease as shown in Figure 2.3.

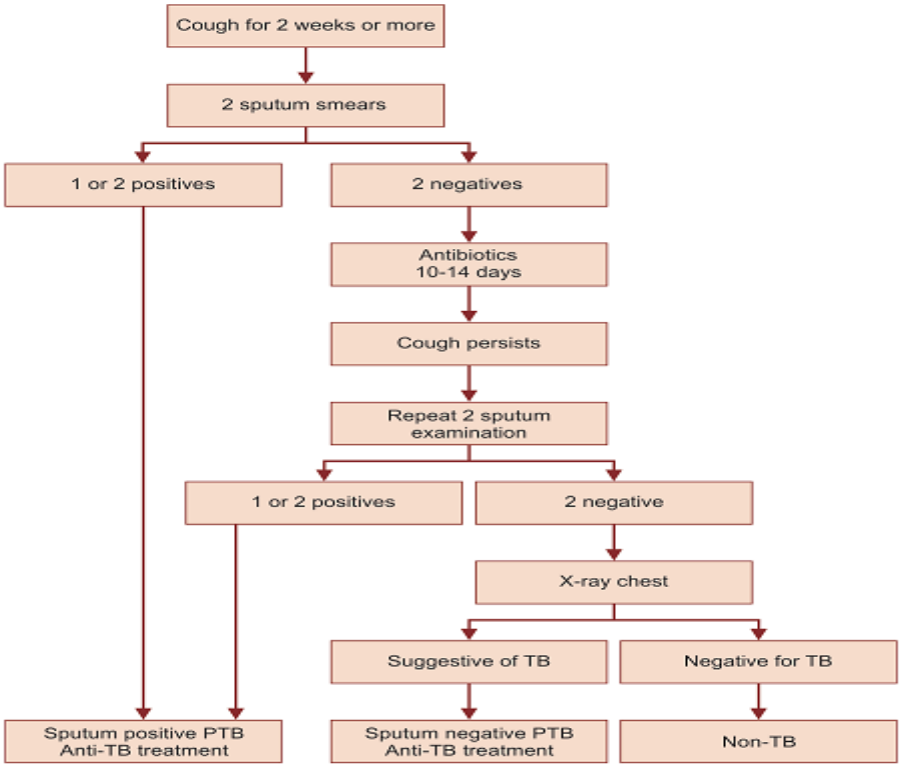


Figure 2.3: Diagnosis process of TB [27]

TB detection for the most part relies upon the clinical finding of the disease and distinguishing the causing bacilli. Among the numerous TB diagnostic tests, the microscopic stand-alone-test is preferred because it can be performed quickly, simply, inexpensively, accurately, and its availability, operational feasibility, ability to identify the highly infectious forms of TB, the smear-positive PTB case is why the sputum microscopy remains the mainstay of diagnosis [8].

After sputum samples are collected from the TB patient, the samples are stained with the ZN-staining procedure, if there are TB bacilli in the slide it will appear in red-pink and the non-bacilli region will appear in blue**.** So, the microscope images of ZN-stained sputum smears contain a pink-colored bacilli region, blue-colored non-bacilli debris region, and the background [28].

The ZN- Staining procedure is used for staining mycobacterium tuberculosis. To stain the smear, first Carbolfuchsin stain, Acid- alcohol 3%v/v, Malachite green 5g/l or Methylene blue 5g/l are evenly spread over the central area of the slide with continues rotational movement.

Then the slide is dried by applying heat which will dry the smear and then the smear is covered with carbolfuchsin stain, the smear is then washed off the stain with clean water and 3% v/v acid-alcohol is applied to the smear for about 2-5 minutes. The smear is washed with water and Malachite green stain with Methylene blue is applied for about 1-2 minutes and the smear is washed and air-dried [8].

Because of the waxy coat nature of Mycobacterium cell wall, it retains an aniline dye even after processing it and decolorizing it with acid and alcohol; they are thus named Acid Fast Bacilli (AFB). This characteristic enables the detection of the bacilli by microscopy. The unique, lipid-rich cell wall mainly composed of mycolic acid and peptidoglycan does not take up stain easily and resists decolorization when destained with an acid-alcohol wash. This allows mycobacteria to be visualized microscopically using Ziehl-Nielsen stained sputum smears in which the organisms will be seen as red rods embedded in a blue background of the counterstain as shown in Figure 2.4 [19].

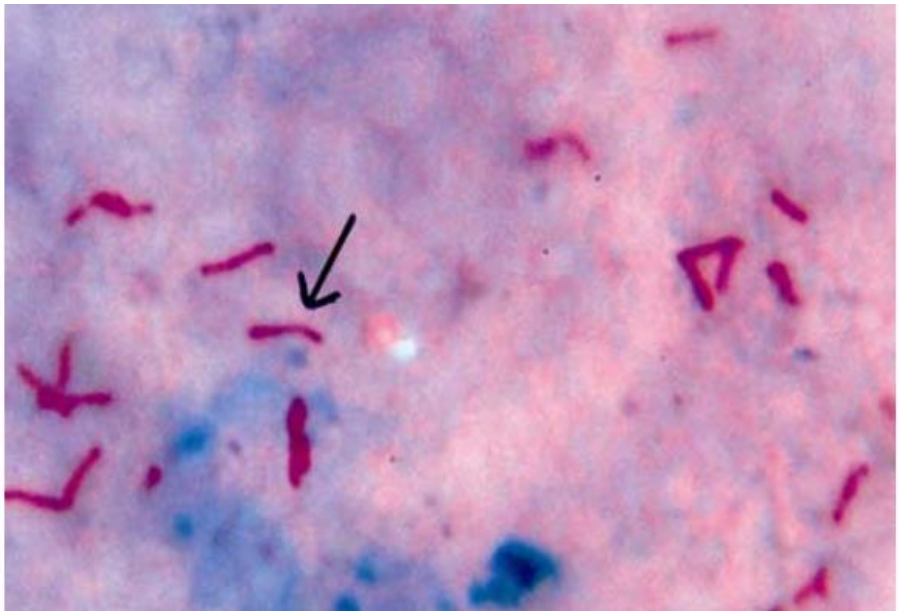


Figure 2.4: Microphotograph of TB bacillus, Mycobacterium tuberculosis on a smear slide stained with Ziehl-Neelsen. Bacilli are long and rod, bent or curved shape; a bacillus is indicated by a black arrow; [28]

When acid-fast bacilli are seen in a smear, they are counted. There is a system for reporting the number of acid-fast bacilli that are seen at a certain magnification. According to the number of acid-fast bacilli seen, the smears are classified as 4+, 3+, 2+, or 1+. The greater the number, the more infectious the patient as presented in Table 2.1 [29].

Table 2.1: Smear Classification Results

Smear Result (Number of AFB observed at 1000X magnification)	**Smear Interpretation**	**Infectiousness of Patient**
4+ (>9/field)	Strongly positive	Probably very infectious
3+ (1-9/field)	Strongly positive	Probably very infectious
2+ (1-9/10 fields)	Moderately positive	Probably infectious
1+ (1-9/100 fields)	Moderately positive	Probably infectious
+/- (1-2/300 fields)	Weakly positive†	Probably infectious
No acid-fast bacilli seen	Negative	Probably not infectious

2.3 Computer-aided diagnosis

Since the 1950s researchers have been studying the prospect of using computers to investigate and solve problems in the field of biology and medicine. It began around the 1960s when researchers tried to use computers for medical image processing and analysis [30]. These authors proposed computerized image analysis for automated diagnosis in radiology, they tried to create an automated computer diagnosis system that could help radiologists in detecting abnormalities and diseases in X-ray images. However, due to limited processing and computing power, lack of data, and insufficient image processing techniques and equipment, those attempts failed.

Ever since then, there has been steady growth and interest in computer diagnosis systems for a handful of diseases and illnesses with the main goal being to produce a system that can rival experts in performance due to the exponential growth of medical information and difficulties.

Although this degree of automation still seems far away, there have been many advances that have made the goal of computer-aided detection and diagnosis closer and feasible.

In medical systems, the task of image analysis can be broken down into 3 essential parts: detection, description, and differential diagnosis. As medical images came to be acquired or displayed through the use of computers, the possibility of having computers that perform any or all of these interpretive steps has been researched and contemplated later this field became known as Computer-Aided Diagnosis (CAD). CAD has been under development for more than 3 decades, researches into this domain has been growing at an alarming rate with more recent medical imaging innovations for x-ray, mammography, chest CT scan, and related technologies that can improve the diagnostic process for other medical images that include vascular imaging, virtual colonography, and automated quantitation of image-derived metrics [31].

CAD can be used to overcome human limitations like the fatigue factor that is common in screening examinations for microscopic and mammography. This is the case when radiologists face an increasingly high-volume of screening examinations that may contain multiple pathologic findings, this can be stressful, tedious and mistakes will be high. Yet, because these examinations are standardized, and because the appearance of pathologies of interest (e.g., bacteria) have relatively multiple different appearances, these types of examinations lend themselves to computer detection algorithms. The more specific the abnormality and focused the detection task is, the better the computer algorithm will likely be [32].

Whether it is a machine or human a perfect observer may never be possible, the use of computers for the detection of abnormalities and diagnoses based on the objective analysis of the clinical information can show improvement in the overall performance and help mend and/or overcome human weaknesses. The difficulty and complexity of medical diagnosis have promoted for the development of computer-aided diagnosis, apart from that the availability of complex clinical data for many diseases, the existence of large amounts of diagnostic

knowledge, and the new advances in the fields of AI, data mining, and machine learning have also contributed [31].

2.4 Computer vision

Computer vision as a domain has been around since the early 60s. It is an interdisciplinary field that is used to teach computers how to develop a high-level understanding by interpreting information present in digital images. In the beginning, computer vision was predominantly a research-based field that rarely ventured outside university and lecture halls because it just wasn't practical back then [33].

Computer vision can empower computers with the properties of human vision, computer could be in the form of a smartphone, drones, CCTV, MRI scanner, etc. Computers use various sensors for perception, the sensors can create images in a digital form that can be understood and interpreted by the computer. Deep learning techniques have been used lately to solve the different problems that arise in computer vision [30]. Computer vision can automate the sorts of information-gathering tasks that the human visual system performs automatically. Simply, it is used to train the computers to become able to automatically process and extract information from images. The research of computer vision, imaging processing, and pattern recognition has made substantial progress during the past several decades [30]. Computer vision techniques have shown enormous outreach for various application areas such as fashion, autonomous cars, and medical fields where it can have a potentially huge impact on healthcare. There have been many research efforts on disease diagnosis, prognosis, surgery, therapy, medical image analysis, and drug discovery [34]. However, detecting medical symptoms is still a challenging problem in the computer vision domain [30].

The basic mechanisms behind computer vision are simple and haven't changed much since the early days, it works by first extracting meaningful features from the raw pixels and matching the extracted features to known labelled ones to achieve recognition. As the

features directly affect the detection, more and more complex features were used such as extraction of edges, lines, texture, and lighting information [35].

But before long it showed up clear that extracting robust and discriminative features were not enough, different elements that belong to the same class can have different properties and look very different such as different-looking cats. Although those elements do share common features, those features are not discriminatory enough. While this has worked for simple image matching tasks, more complex applications of the computer vision such as instance classification cannot be solved by essentially contrasting pixel features from inquiry images with those from labelled. This is where machine learning comes into play, with an increasing number of researchers shifting their interest and trying to tackle image classification through statistical discrimination of images [35].

Taken together, current research problems give birth to the rise of deep learning researches. Since the rise of deep learning, computer vision has demonstrated the ability to become useful for wide and various applications and areas. With the plethora of training data and models being developed today, computer vision applications can now identify, recognize, and track nearly any object. It can be now used to identify a vast range of fields, it can also be used for identifying things the human eye can't detect, such as minuscule defects in highly refined products or cancerous cells during a medical procedure [36].

According to Garcia-Garcia et. al. [37], computer vision is classified into multiple distinct subtasks. These are image classification, object detection, semantic segmentation, and instance segmentation. Image classification refers to the task that predicts the probability of an object belonging to a class, it does not provide pixel-level information. Whereas, object detection locates the different instances of objects in an image and predicts the class of each of them. Semantics segmentation is another subtask where it predicts the class for each pixel in an image. On the other hand, instance segmentation refers to the combination of object detection

and semantic segmentation. It locates object instances with pixel-level accuracy as shown in Figure 2.5 [38].

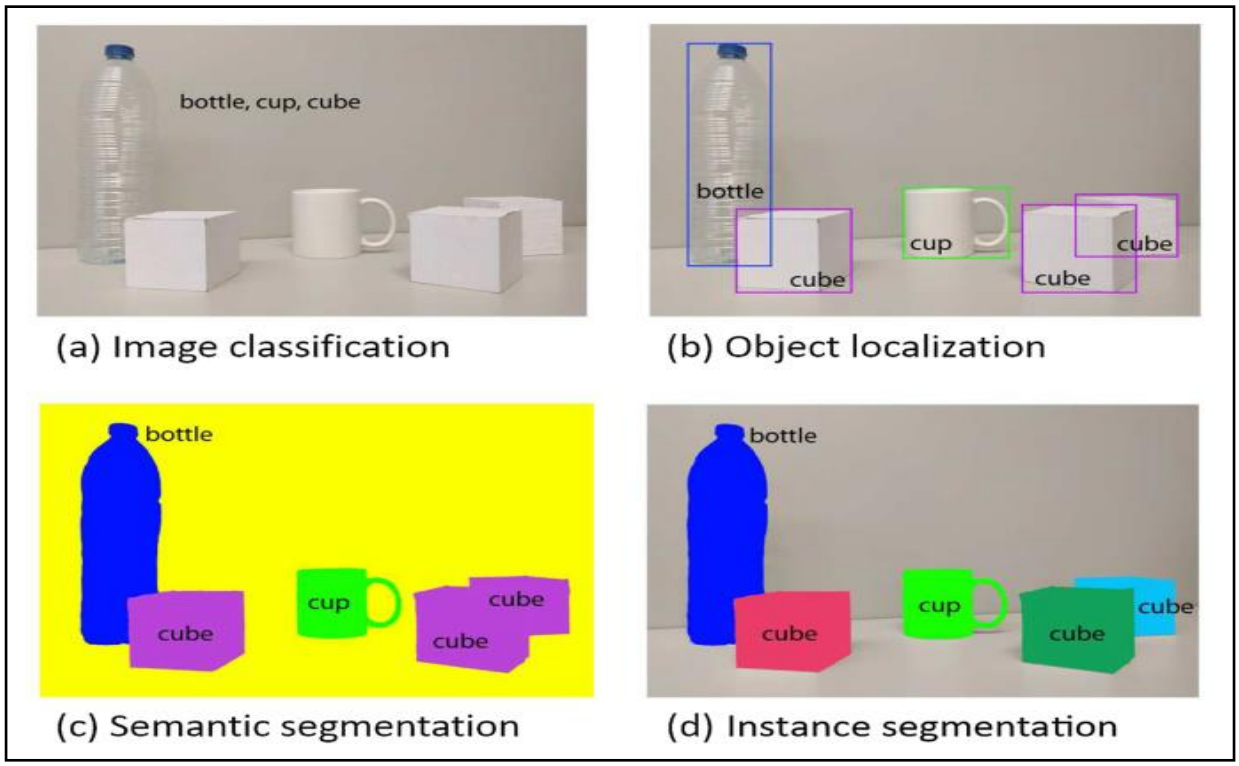


Figure 2.5: Evolution of object recognition or scene understanding from coarse-grained to fine-grained inference: classification, detection or localization, semantic segmentation [37]

Instance segmentation is the natural evolution of Object Detection, Semantic Segmentation, while object detection is used to detect the different objects in an image, semantic Segmentation is used to predict the class of every pixel, creating a mask for every class present. Both object detection and semantic segmentation are performed simultaneously, locating object instances with pixel-level accuracy. Each pixel of the image is classified into one of the pre-defined classes as in Semantic Segmentation. On top of that object detection is also performed. This means that each resulting mask is not going to contain all the objects from the same class, but rather only one instance of one class, i.e., different objects from the same class will have different masks, it allows to locate different instances of the same class appearing in an image [38].

2.5 Machine Learning

Only intelligent systems, biological or otherwise have the ability to learn. In artificial systems, training or learning is the process of updating the representation of the internal system in response to external stimuli to perform a specific task. Machine learning is an algorithm that uses rules and parameters on input data to learn from this input. It has two learning types supervised and unsupervised. Supervised learning is a type of learning that uses a labelled training dataset for learning whereas unsupervised learning discovers the representation of the data without class labels. In supervised learning, we have examples of expected output whereas unsupervised learning inferences are drawn from the input data without no labelled data. Training a machine learning model is difficult due to numerous details that can hinder it from achieving its full potential [39].

Machine learning has made it possible for the computers and machines to make decisions that are data-driven aside from just being programmed explicitly for following through with a specific task. These sorts of algorithms also as programs are created in such a way that the machines and computers learn by themselves and thus, are ready to improve by themselves once they are introduced to data that's new and unique to them altogether. Traditional machine learning is a very old field that uses methods and algorithms that have been around since the early 60s. Its applicability in feature extraction is limited to specified explicit rules. These conventional machine learning techniques heavily relies on data representations that need huge domain expertise and sophisticated engineering [40].

However, deep learning has made it possible to overcome this issue through neural networks. Artificial neural networks **(**ANNs**)** also known as neural networks **(**NNs**),** are powerful machine learning tools that are excellent at processing information, recognizing usual patterns or detecting newer ones, and approximating complex processes by learning from examples that they encounter. It is an algorithm that was originally motivated by the goal of having machines that can mimic the human brain, It consists of an interconnected group of artificial neurons

which are physically cellular systems capable of processing, obtaining, and storing information, and using it for experiential knowledge [41].

In digital pathology and cell biology, the high-dimensional microscopic data contain complex patterns and dependencies in images such that they exhibit very complicated relationships with disease expressions and image labels. Additionally, images have extensive variations because of the heavy staining process and data preparations. All of these problems significantly challenge many generic image analysis methods and conventional machine learning algorithms in the tasks of microscopic bacteria detection, segmentation, and classification [42]

2.5.1 Deep Learning

The initial development of neural networks was fuelled by an ambition to create a system that can simulate the human brain. Deep learning(DL) has been around since the 1950s when Frank Rosenblatt [43] came up with the basic idea that deep learning was built upon which is called perceptron. It was a machine learning algorithm that was inspired by how human neurons work and the fundamental block of the first neural networks with a better learning procedure on top of it, this method was capable of recognizing characters.

Since then deep learning has been heavily researched upon and several research papers introduced neural networks with multiple layers of perceptrons put one after the other that could be trained using a rather straightforward scheme called backpropagation [35]. Soon after that, the first convolutional neural network **(**CNN**),** the ancestor of current recognition methods, was developed and applied to the recognition of handwritten characters with some success.

Recent researches in Artificial Intelligence (AI) led to the spread of modern techniques of machine learning based in deep structures.

The term deep learning comes from the use of deep neural networks that have multiple layers of neurons, deep neural networks are networks that use multiple hidden layers between the input and output layers where each layer processes its input and passes the results to the next

layer. All of the layers are trained to extract abstract information hierarchically from basic features such as edges, lines, or color gradients to the more advanced features. Deep learning can process raw data directly (e.g., RGB images) and learn the representations automatically, which can be used for image segmentation, object detection, or image classification. Compared with hand-crafted features, learned representations require fewer human interventions and provide performance enhancement [44].

Due to their ability to learn high-level features from data, deep learning has been successfully used for several computer vision and natural language processing applications. Over the last few years. Deep learning models have shown to be capable to outperform previous state-of-the-art machine learning techniques in several fields. Due to this and the abundance of complex data from different sources, deep learning has been applied extensively and set new benchmarks in different areas including object detection, motion tracking, action recognition, human pose estimation, and instance segmentation. Deep learning models for computer vision achieved statistically impressive results, they are currently the state-of-the-art in many computer vision and image processing problems, in particular image classification [45].

Deep Learning is providing a major breakthrough in solving the problems that couldn't be solved by traditional machine learning approaches. As a result, it is currently being used to solve hard scientific problems at an unprecedented scale. It is additionally a standout amongst the most prevalent logical research interests nowadays. From time to time, new deep learning systems are being conceived, beating cutting edge machine learning and notwithstanding existing deep learning procedures. However, it requires high computation power to train, the deeper they are the harder it is to train.

The appearance of large, high-quality, publicly available labelled datasets, along with the advancement of superior parallel GPU computing allowed for significant acceleration in deep models' training. This and the emergence of powerful frameworks like TensorFlow [46], Theano [47], and Mxnet [48], has made the process a lot simpler and easier and brought the

ability of fast prototyping. It has turned out to be a lot simpler to prepare huge scale deep neural networks with a great many parameters, there have been huge advances in the plan of network structures, models, and training strategies.

Recently, deep learning has been attracting huge interest in microscopy image analysis, including nuclei detection, cell segmentation, tissue segmentation, image classification. Due to the ability of deep learning to learn abstract feature representations in a hierarchical way it can discover hidden data structures and representations in large-scale and high-dimensional image data for microscopic image analysis. Meanwhile, deep learning can significantly reduce or eliminate the burden of feature engineering in conventional machine learning techniques. Nowadays, deep learning is the major method among the best solutions for many tasks in microscopy image analysis, and thus it holds great promise for the field [49].

The most diffused DL architectures are Convolutional Neural Networks (CNN), which can classify images into several categories, automatically learning features through convolutional layers that combine multiple non-linear processes. It can learn effective hierarchical feature representations that characterize the typical variations observed in visual data, including images and video, which make them very well-suited for most of the visual classification tasks [50].

- **CNN**

Convolutional Neural Networks (CNNs), are the most used deep learning model to solve computer vision tasks today, especially in image classification. CNNs were inspired by the structure of the visual system in the research proposed in [51] where they studied the receptive field of a cell in the visual system of cats.

A model based on that research was proposed in Neocognitron [52] which uses the local connectivities between neurons and hierarchically organized transformations of the image, The authors proposed that when neurons with the same parameters are applied on patches of the previous layer at different locations, a form of translational invariance is acquired. Later

convolutional neural networks were designed employing the error gradient and it attained very good results in a variety of pattern recognition tasks such as face recognition, object detection, powering vision in robotics, and self-driving cars [53].

The basic building blocks of CNNs are: -

- convolutional layer: is a layer used to convolve the image using various kernels and intermediate feature maps and generate various feature maps.
- pooling layer: is a layer used to reduce the spatial dimensions (width × height) of the input volume for the next convolutional layer, it does not affect the depth dimension of the volume.
- fully-connected layers: it is a layer used to convert the 2D feature maps into a 1D feature vector which could either be fed forward into a certain number of categories for classification or be considered as a feature vector for further processing.

Each layer plays a different role, every layer of a CNN transforms the input volume to an output volume of neuron activation, eventually leading to the final fully-connected layers, resulting in a mapping of the input data to a 1D feature vector.

Architectures such as AlexNet [54], VGG [55], ResNet [56], and GoogLeNet [57] have become very popular, they are used as subroutines to obtain representations that are then offered as input to other algorithms to solve different tasks.

Overall, CNNs were shown to significantly outperform traditional machine learning approaches in a wide range of computer vision and pattern recognition tasks, their exceptional performance combined with the relative easiness in training are the main reasons that explain the great surge in their popularity over the last few years [58].

- **R-CNN**

Region-CNN (R-CNN) is a CNN-based deep learning method for a handful of computer vision subtasks such as object detection, object classification, and instance segmentation, It is currently

the state of the art method. R-CNN family contains Fast R-CNN and Faster R-CNN for object detection and Mask R-CNN for instance segmentation [59].

A convolutional neural network (CNN) is mainly used for image classification. A typical CNN can only classify and tell the class of the object but not where they are located. While it is possible to regress bounding boxes directly from a CNN it can only regress one object at a time. If there are multiple objects in the visual field then the CNN bounding box regression cannot work well due to interference. In CNN, for every input, a sliding window is used to search the entire input of objects. While this simple solution is effective for single object detection, it suffers with the input that has multiple objects in the field of view, whether those objects are in the same class or different objects [60].

In R-CNN, the images are divided into region proposals and then CNN is applied for all of the regions so instead of the network working on a massive number of regions, several boxes are proposed that may contain any objects, these boxes are called regions. The size of the regions is calculated and determined and only the correct regions are fed into the artificial neural network. In R-CNN the CNN focuses on a single region at a time that way interference is minimized because it is expected that only a single object of interest will be found in a given region. The regions in the R-CNN are detected in equal size before they are fed to a CNN for classification and bounding box regression [59].

Because the number of occurrences of the objects of interest in microscopic images are not fixed for that reason, a standard convolutional network will not suffice. A naive approach to solve this problem would be to take different regions of interest from the image and to use a convolutional network to detect and classify if an object is present within the region. However, the challenges with the process are the objects have different aspect ratios and spatial locations within the image which gives the network a difficult time learning the features. There are models like Mask R-CNN that are more appropriate and can overcome all this issue and find all the occurrences of objects in the input images.

- **Mask R-CNN**

Traditional image detection and segmentation heavily use manual feature selection to perform instance segmentation, The Mask R-CNN model is used for classification and instance segmentation. It is currently the state-of-the-art method in instance segmentation [61]. It is a deep neural network developed for solving the instance segmentation issue in computer vision. It is used to segment all the objects in the input whether it is image or video. It is an extension of Faster R-CNN where it extends the capability of the network by adding instance segmentation [62]. Mask R-CNN has been shown to surpass all the previous state-of-the-art model results on the COCO instance segmentation task, it also excels in the COCO object detection task [63].

While CNN is mainly used for image classification and isn't capable of segmentation, Mask R-CNN can perform object detection, localization, and instance segmentation at the same time, it predicts the class of every pixel creating a mask for every class present [63]. Since Mask R-CNN is a network capable to perform multiple tasks on the input, it has a multi-task loss function that combines the loss of classification, localization, and segmentation mask with each sampled ROI the Loss is defined as:

$$L = L_{cls} + L_{box} + L_{maks} \quad (1)$$

Where L_{cls} is Classification Loss is, L_{box} is Bounding Box Regression Loss and L_{mask} Is Mask Loss.

Mask R-CNN has components such as Feature Pyramid Network (FPN), Region Proposal Network (RPN), Region of interest Align (RoiAlign).

- **FPN**

it is a feature pyramid that can make predictions independently at all levels. FPN shows significant improvements in overall existing state-of-the-art methods and can achieve higher accuracy. Detecting objects in different scales is challenging in particular for small objects. Feature Pyramid Network (FPN) is a feature extractor designed for such a pyramid concept with

accuracy and speed in mind. FPN composes of a bottom-up and a top-down pathway, the pathways are the computation of the backbone of the convolutional network in a feedforward way as shown in figure 2.6 [64].

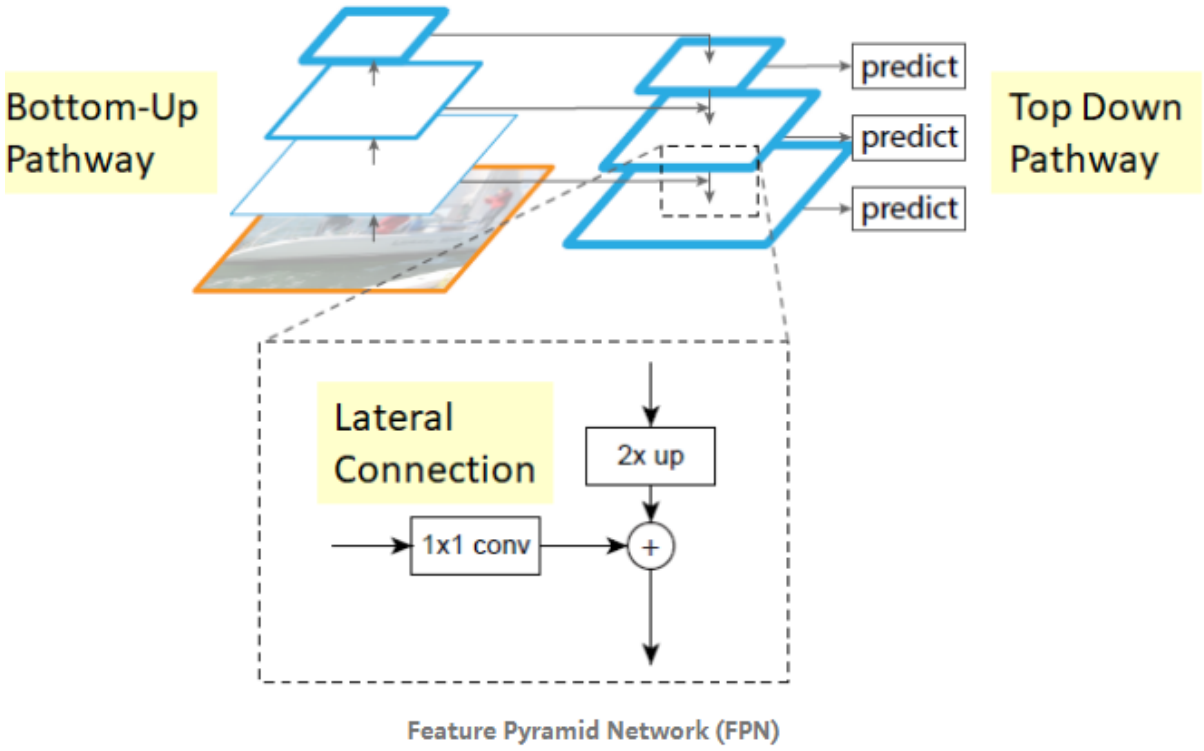


Figure 2.6: Bottom-up and top-down pathway [64].

- **Region proposal network (RPN)**

Region proposal Network (RPN), is a network that can generate proposals or regions that contain an object from the input. This network is robust against translations, therefore one of the key properties of this network is translational invariant, where it can keep the spatial information of each object. It is less time-consuming and more cost-efficient than previously proposed algorithms.

RPN loss is as follows:

$$L = L_{cls} + L_{Box} \tag{4}$$

where L_{cls} is Classification Loss and L_{box} is Bounding Box Regression Loss

Classification Loss – is the gradients that are computed over a simple cross-entropy loss function commonly used in multi-class classification.

Regression Loss - The regression loss is defined as

$$L = R(t - t^*) \quad (5)$$

where t and t^* is the vector representing the four vertices of the predicted and ground truth bounding box, and R is the robust loss function.

- **ROIALIGN**:

Conventional CNN uses the Region of interest Pool (RoIPool) layer. RoIPool quantizes all the floating-number in the region of interest to the discrete granularity of the feature map, the quantized region of interest is then divided into spatial bins then quantized all over again. Regular ROI pooling changes the topology of the features because it performs quantization, these quantization introduce misalignments between the ROI and the extracted features**,** It loses a lot of data in the process Every time it does that, part of the information about the object is dropped. Using regular ROI pooling would negatively impact the ability to predict pixel-accurate masks, to address this, Mask R-CNN uses the RoIAlign. Unlike RoIPool, In RoIAlign all hard quantization of the RoI boundaries or bins is totally avoided [61].

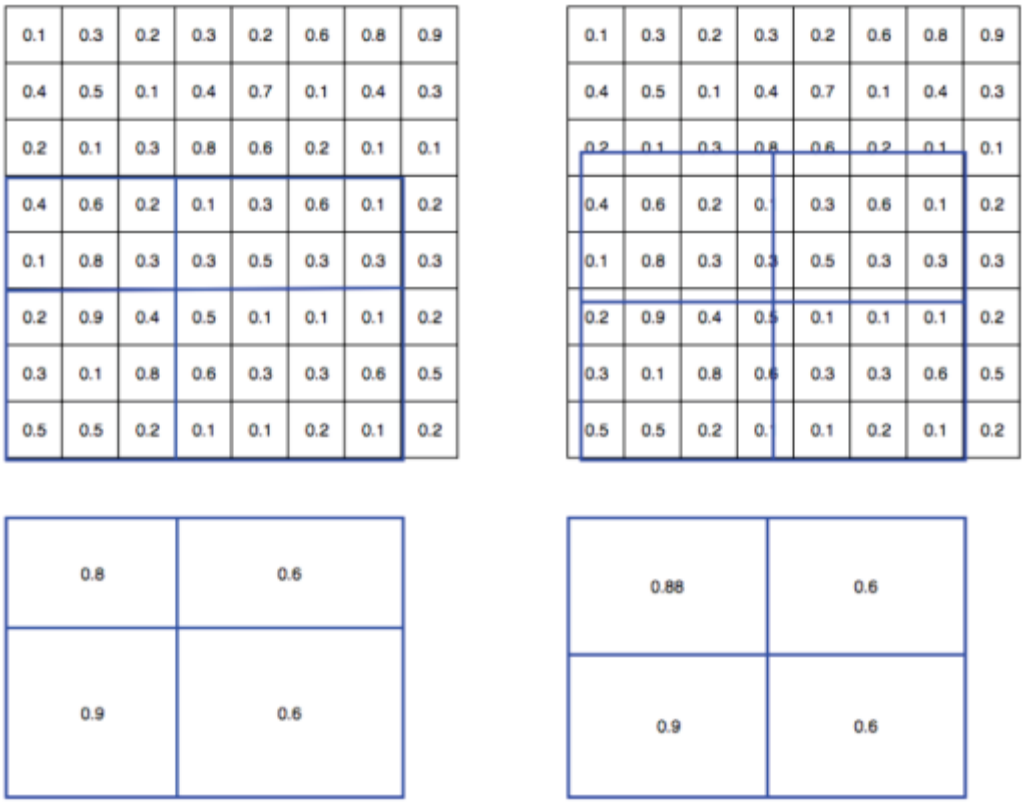


Figure 2.7: RoIPool and RoIAlign [61]

2.5.2 Deep Learning overfitting problem

The objective of deep learning is to create a model that can automatically solve problems on it is own. To do that, deep learning models are normally trained with a training dataset that consists of situations where the output is known, and later how well the model learned is checked using the testing dataset that is new to the model.

Overfitting is when the deep neural network is training well using the training dataset but testing terribly with the test dataset, in a way the network is memorizing certain patterns rather than learning. It is an extremely common problem in deep neural networks. The more expressive and deeper the network is (i.e., very expressive deep neural networks that contain multiple non-linear hidden layers), the more it is prone to overfitting [65].

Fortunately, there are a few techniques that are proved to be effective in handling overfitting to some degree such as regularization and data augmentation [66].

Regularization: It is the ability of a model to adapt to unseen new data taken from the same dataset the model was trained on i.e., the model ability to generalize on the test dataset [66]. The most well-known regularization technique are: -

- Dropout: it is a regularization technique that is used to zero out the activation values of neurons that are randomly selected during training. That way the network is capable of learning robust features rather than using then the predictive capability of a small subset of neurons in the network [66].
- Batch normalization: it is another regularization technique that is used to normalizes the activations in a layer. Normalization subtracts the batch's mean from each activation and divide by the batch's standard deviation. [67].

Data augmentation: Data augmentation overcomes the overfitting problem by fixing the problem of having small training dataset by extracting more information from the original dataset through augmentations. These augmentations are used to artificially inflate the training dataset size by either data warping or oversampling. Data warping augmentations transform existing images such that their label is preserved. Data augmentation techniques increase the training data available to the model without collecting new labelled data. The idea is to take a labelled data point, perform a modification that does not change the meaning of the data point by applying domain-specific techniques to examples from the training data that create new and different training examples. and adding it to the training dataset [68].

2.5.3 Evaluating Deep Learning network

In deep learning, model validation is the process where a trained model is evaluated with a testing data set. For validating a deep neural network, metrics are used such as precision, recall, and F-Score.

To calculate these metrics, first the confusion matrix of True Positives, False Positives, and False Negatives are calculated. True Positives (TP) is the number of objects correctly detected by the model, False Positives (FP) is the number of objects wrongly detected by the model, and False Negatives (FN) is the number of objects missed and not detected by the model [69]. Table 2.2 shows representations of the confusion matrix.

Table 2.2: Confusion Matrix

Confusion Matrix		Conditions	
		P	N
Test Results	P	TP	FP
	N	FN	TN

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Where TP is True Positives, FP is False Positives

Precision is a metric that is used to quantify the number of correct positive predictions made, the precision is the ratio of correct positive predictions out of all positive predictions made [70].

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

Where TP is True Positives, FP is False Positives

Recall is a metric that is used to quantify the number of correct positive predictions made out of all positive predictions that could have been made. Unlike precision that only comments on

the correct positive predictions out of all positive predictions, recall indicates the missed positive predictions. In this way, recall provides some notion of the coverage of the positive class [70].

$$\mathrm{F-Score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

Where TP is True Positives, FP is False Positives

F-Score is a single measure that is used to summarize model performance, it is used to combine both the result of precision and recall into a single measure that captures both properties [70]. F-score can be used to level out the performance of the model in the case where one of the matrics is high and the other is low. For example, a good recall can level out a poor precision and produce an okay or reasonable F-score. Neither precision nor recall alone can't generalize and tell how a model is performing, there could be excellent precision and terrible recall or terrible precision with excellent recall. F-score is used to express both concerns with a single score. It is heavily used in imbalanced classification problems.

2.6 Hungarian algorithm

Hungarian algorithm is a combinatorial optimization algorithm that solves the assignment problem in polynomial time and which anticipated later primal-dual methods. It was developed and published in 1955 by Harold Kuhn, who gave the name "Hungarian method" because the algorithm was largely based on the earlier works of two Hungarian mathematicians: Dénes Kőnig and Jenő Egerváry [8 - 9]. Hungarian algorithm can perform N by N matching, where each element in the set can have exactly one matching by using bipartite matching.

A Bipartite Graph G = (V, E) is a graph in which the vertex set V can be divided into two disjoint subsets X and Y such that every edge e ∈ E has one endpoint in X and the other endpoint in Y [8]. A matching M is a subset of edges such that each node in V appears in at most one edge in M as shown in Figure 2.8.

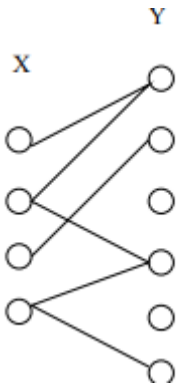


Figure 2.08: bipartite graph matching

Hungarian algorithm can be used for matching in images where all the object locations are known, it can calculate loss function that shows how the objects are matching. The algorithm is used by multiple researches in the field of computer vision latest research being [73] where the authors developed an end-to-end detection model that utilizes an online Hungarian algorithm. At each batch, the model will compare and match between the batch and ground truth and removing all wrongfully detected objects. Even though using the Hungarian algorithm this way is very costly and can make the training time very long, this work proved the Hungarian algorithm can be implemented in deep learning quite easily if all the preliminary research is done.

2.7 Hard example mining

Hard example mining had existed for the last 20 years. It was originally called bootstrapping. Bootstrapping was first presented in the research of Sung and Poggio [74] in the 1990s for a face detection model. The main idea was to slowly increase and bootstrap the training set by adding those examples the model detected as a false alarm. They used an iterative training approach that iterates between training and adding new false positives.

Bootstrapping methods train a model with an initial subset of negative examples, and then collect negative examples that are incorrectly classified by this initial model to form a set of hard negatives. A new model is trained with the hard-negative examples and the process can be repeated as many as wanted.

Hard examples are the images in the training dataset that the model misses detecting and misclassifies and struggles with. Classification on an imbalanced dataset is very hard because the model will learn one class better than the other. The class with the most data points and examples will converge better and have a higher detection rate. To mitigate this, hard example mining algorithms have been widely used [74]. A hard example mining algorithm is an algorithm that is used when the training set has a large imbalance between the number of annotated objects and the number of background examples (image regions not belonging to any object class of interest).

To obtain high performance using discriminative training it is often important to use large training sets. In the case of object detection, the training problem is highly unbalanced because there is vastly more background than objects. This motivates a process of searching through the background data to find a relatively small number of potential hard examples. Felzenszwalb et al. [75] proved if done right, hard example mining methods can be made to converge to the optimal model defined in terms of the entire training set.

Chapter 3: Related work

3.1 Introduction

In this Chapter, previous works related to detection, segmentation, and estimation of TB bacteria in sputum are reviewed. The research works related to the detection of TB using digital image processing are reviewed, the approach, datasets used, results obtained and gaps that exist in previous works related to detection, segmentation, and estimation of TB bacteria in sputum are also identified.

In the remaining part of this chapter, we will discuss the method and existing work for automatic TB detection using conventional microscopic images.

3.2 Detection of TB using Traditional Machine Learning approaches

Costa Filho et al. [11] presented a model for the automatic detection of TB. The authors came up with a two-step process starting with the Bacillus segmentation and then post-processing. For the segmentation, they used 30 features as input. The features were selected from four-color space: RGB, HSI, YCbCr, and Lab. For the post-processing, they used three fillers for separating the bacillus from the debris and artifacts which are geometric filter, Rule-based filter, and size filters, all the filters use the Red Green Blue (RGB) color space. The model had used only the geometric characteristics for the evaluation, and the authors disregarded any touching, occluded, and overlapped bacilli and consider it as non-TB while calculating the result of their proposed model.

Jun-Jun [76] presented a predictive model using high-resolution computed tomography (HRCT). The model can diagnose final smear-positive (SP) active PTB for patients with initial negative acid-fast bacilli (AFB) sputum smears. The authors computed the predictive score from the image samples to identify the TB disease. However, the authors did not perform a sensitivity analysis to determine how modifications may affect the formulation of a relative score to be used for prediction purposes, also the study included patients from different hospitals, and while regional differences may be present.

João et al. [77] developed an automatic detection based on the Artificial Neural Network. They used the Multilayer Perceptron (MLP) based on adaptive resonance theory (iART) for the detection purpose. This model uses data collected from 136 patients suspected for smear-negative pulmonary tuberculosis in a general hospital. This model only works well for the detection and classification of the smear-negative samples. Since the authors used MLP and iART modules that operate in a complementary fashion, all the discordant results require further clinical investigation

Priya and Srinivasan [78] presented an automated TB detection system based on a combination of a support vector machine classifier and the neural network based classifier. They used Multi-Layer Perceptron neural network activated by Support Vector Machine. For the selection of the Fourier descriptors, they chose to use fuzzy entropy measures for the segmentation. The dataset they use was collected by them and they recorded it under standard image acquisition protocol. However, the segmentation approach suffered from the presence of the artifacts in the images, and it suffered from the balancing problem.

Ghosh et al. [79] proposed a method for the segmentation of the Bacilli solely through shape, color, and granularity feature. They used a gradient-based region growing technique for finding the contour boundary, they first start by highlighting the Mycobacterium region filtering the pink region. The pink region is segmented from the output matrix and the threshold value for a binary convention is calculated. This will be applied upon each contour that is segmented by the label matrix technique. However, their method failed to identify overlapping bacillus, the detection technique fails when two or more bacteria are touching.

Chao Xu et al. [80] presented a segmentation method by combining hierarchy tree analysis with gradient vector flow snake, skeletons of the objects were generated by mathematical morphology, The skeletons of the objects were used for structure analysis based on the hierarchy tree. And the gradient vector flow snake was used to estimate the object edge.

However, the touching bacilli are considered as another class of objects or negative ones and can’t detect it.

3.3 Detection of TB using Deep Learning approaches

Kant et al. [17] proposed an approach that used a patch-wise detection strategy where a microscopic field view of 20x20 patches from the input image are extracted and classified one at a time for the presence or absence of bacilli. The authors used a simple Fully-Convoluted Neural Network Architecture without any fully connected units, with Rectified Linear Units and a last of layer Softmax activation function. two instances of the same neural network architecture are trained separately. The microscopic images were splitted into smaller patches that contain an object that could be a TB bacillus. The CNN was trained on patches rather than the whole image. The biggest drawback of the method was how the patches were generated and the split of the larger image into such patches, the accuracy of the method largely depends on this preliminary patching step. They didn’t specify whether the patching was done manually or automated.

Panicker et al. [18] proposed a method for segmenting the foreground and background of the image into objects like single bacilli, touching bacillus, and other debris artifacts. The segmented foreground objects are then given to a trained convolutional neural network (CNN) and CNN will classify the objects into bacilli and non-bacilli. The CNN is trained with patches rather than the whole image, the required patches are constructed presumably containing a foreground object and then fed into the CNN for classification. The patch construction depends on the initial stage of image binarization and pixel classification to locate foreground objects and the performance of the CNN heavily depends on the competence and success of the first binarization/pixel-classification which could be very error-prone for the challenging conventional bright-field microscopy due to the overstaining, and touching foreground objects.

Lopez et al. [16] developed a CNN model for detecting TB bacillus from sputum smear images using RGB, grayscale, and R-G patches versions and then training 3 different CNN models as

clusters. The authors used patches for training rather than full smear microscopy images, However, the authors didn't mention how the patches were generated, whether it is manual or automated, and the proposed system needs extra requirements to work with full smear microscopy images.

3.4 Summary

In this chapter, having studied the work of different researchers that are related and relevant to our proposed model we came up with the following major problems of the present TB diagnosis methods, which may lead us to a better solution.

Current research works on TB detection from sputum images suggest that there were limited research outputs in this medical image processing field using deep learning techniques, based on reviews of this Chapter, different literatures including [18] recommend that there is a great scope of deep learning-based research in the area of TB detection and medical image analysis.

According to the related works reviewed in this Chapter, there have been very few small numbers of works that used deep learning to solve the problem of TB detection [16,17,15]. Those works share the same common drawbacks, most of the deep learning techniques proposed used data patches which are done by extracting a small portion from the image that contains the bacillus object and feeding the networks with that small portion of the microscopic image. While this process generates a decent performance, the patches are generated manually and it requires a subject domain expert. Accordingly, the developed system can't work with full microscopic images, due to the fact that every time the system is used, the patches have to be constructed from the images.

On the other related works including [10, 74, 75,17], the proposed approach cannot classify touching or overlapped bacilli as true bacilli. In some works, they approach the touching and overlapping bacilli as negative or other classes while in other works the touching bacilli are removed from the dataset before the experimentation citing irregular shape and complexity as

the reason why it is removed. They concentrate on the segmentation of single bacilli objects only, all of the bacilli that are touching, occluded, and overlapping was either completely removed or considered non-bacillus and negative class as they focused only on classifying single bacilli.

Researches in [10, 73, 74, 75] used classical image processing and traditional machine learning techniques to solve the TB detection problem. The feature extractor combined with a classifier is intuitive but suffers from a lack of generality. Also, it demands a certain level of expertise in the field of the problem to pick the correct feature extractor.

These were the limitations observed so far in the related works that will be addressed by this thesis work. We will try to fill the gaps mentioned above (such as the need for patching of the sputum images and not being able to handle touching and overlapping bacillus) and solve the problems pointed out by proposing an automatic TB diagnosis and segmentation model for the microscopic image that enables better performance using regions with convolutional neural networks (R-CNN).

Chapter 4: Proposed solution

4.1 The proposed model

We have developed a model for TB detection by tailoring it to the specific gaps explored in the related work chapter, where each component can overcome a specific problem that riddles the current work.

The model as shown in Figure 4.1 has 3 components that are Mask R-CNN, Hungarian Algorithm, and Hard example mining. We used Mask R-CNN for instance segmentation. In this Mask RCNN model, we have components such as the FPN backbone module, RPN module, and the segmenter of a fully connected layer for mask representation. Then, we used a matching loss function that uniquely assigns a prediction to a ground truth object and to filter out misclassified predictions using the Hungarian algorithm.

We used hard example mining to provide some hard samples, the mining for hard examples was done by using the Hungarian algorithm. Therefore, the hard example mining method can optimize the model and improve the learning efficiency, each of the components are discussed in details in the next sections.

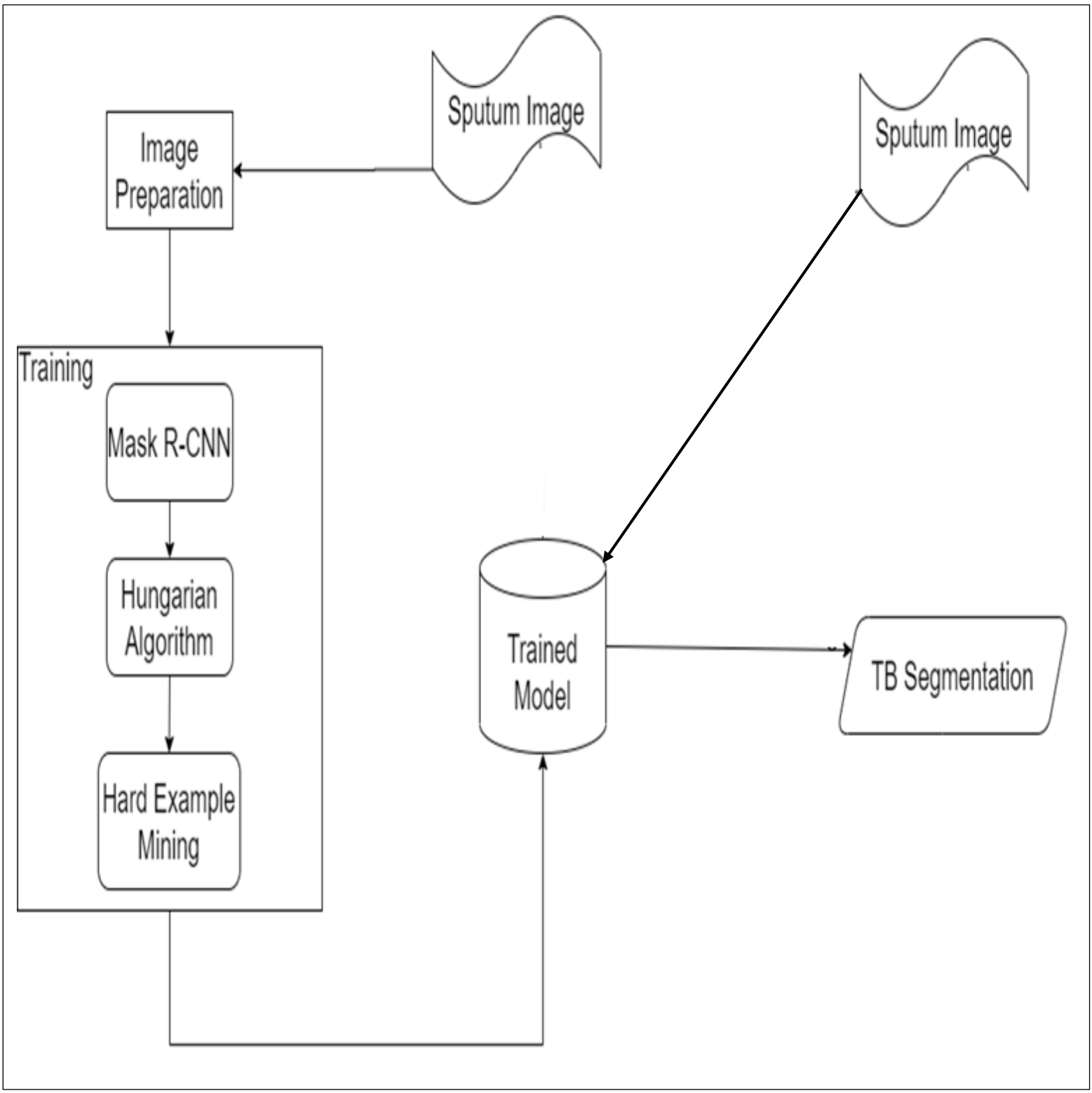


Figure 4.1: Proposed model

4.2 Image preparation

The efficiency of the deep learning model is affected by the input data, before fitting the model a uniform aspect ratio, image scaling, mean and standard deviation, normalization, and data augmentation is used to perform the final preparation of the data.

We cropped out around the image boundary and used the square shape image to maintain the uniform aspect ratio. This uniform aspect ratio facilitates the scaling up or down of input data to provide variation of image data. The normalization of the training dataset was performed by calculating the mean and standard deviation of the data and subtracting each pixel value from the mean and standard deviation.

Since the dataset doesn't come with ground truth mask data, we had to prepare ground truth masks for the dataset, the masks were prepared from the labels, annotation, and bounding box of the dataset. For every image in the dataset, we prepared ground truth of binary mask images using the bounding box data and we saved it in the form of PNG format with an identification name that matches the corresponding image. All the binary masks have the exact dimension as the original image. The mask images are encoded as grey 8-bit images which means each pixel represents 0 or 255 where 0 (black) is representing background and 255 foregrounds of the image (bacilli) as shown in Figure 4.2.

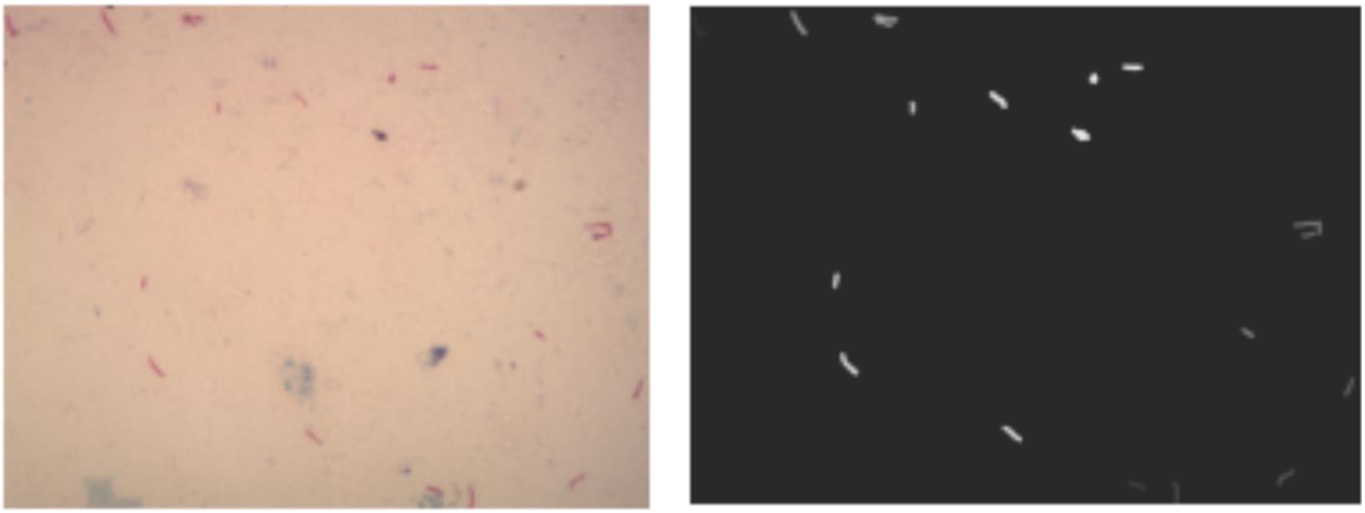


Figure 4.2: Data Preparation

4.3 Training and model construction

The training of the model has three consecutive components, Mask R-CNN, Hungarian algorithm and Hard Example mining. The training process is sequential, the model works by first taking an input of the sputum image and running all three components sequentially and finally outputting an image with all the bacillus object segmented and annotate. After the training process finishes the model is constructed and the trained model is saved to be used for other purpose such as testing and detection.

3.3.1 Mask R-CNN

The main functions of our Mask R-CNN are bacillus detection, bacillus classification, and bacillus instance segmentation. The model performs instance segmentation and produces a mask segmentation. Our Mask R-CNN model is composed of four parts: the feature extraction network (FPN), the region proposal network (RPN), ROIAlign, and the object classification and segmentation network.

Mask R-CNN has two stages, In the first stage, we scan the input and generate proposals i.e., areas that are likely to contain an object. And the second stage classifies the proposals generated earlier and predicts whether the object is bacillus or not then refine the bounding box, and generate a mask at the pixel level of the bacillus based on the first stage proposal. Both stages are connected to the backbone structure. Mask R-CNN has three outputs for each candidate bacillus, a class label, a bounding-box, and an object mask. The additional mask output is distinct from the class and box outputs, requiring extraction of a much finer spatial layout of an object.

Our Mask R-CNN network can be split into two heads. We used the first head to perform the task of classification of the object and bounding box prediction, and then we use the other to predict the masks. The classification and bounding box regression branch share two fully connected layers before splitting off from each other. The Mask head has four more convolutional layers before being up-sampled by a transpose convolutional slayer, bringing the

dimensions of our feature mask to 56x56x2, where 2 is the number of classes we are segmenting (TB or background). One more convolutional layer is followed by a sigmoid activation function to convert our per pixel regressions into the probability of this particular pixel belonging to the Background.

Mask R-CNN uses the multitask loss function defined in Chapter 2 Section 2.4.1. The mask head gives 2x2 dimensional output for each region of interest, where 2 are the binary masks we generated having the resolution of m x m for each class. Then we use per-pixel sigmoid and compute the average binary cross-entropy for the Lmask. Then we use L_{mask} to make a threshold for the classification and segmentation by computing the confidence score for each prediction, all the objects with a confidence score less than 0.8 are automatically removed.

As shown in Figure 4.3

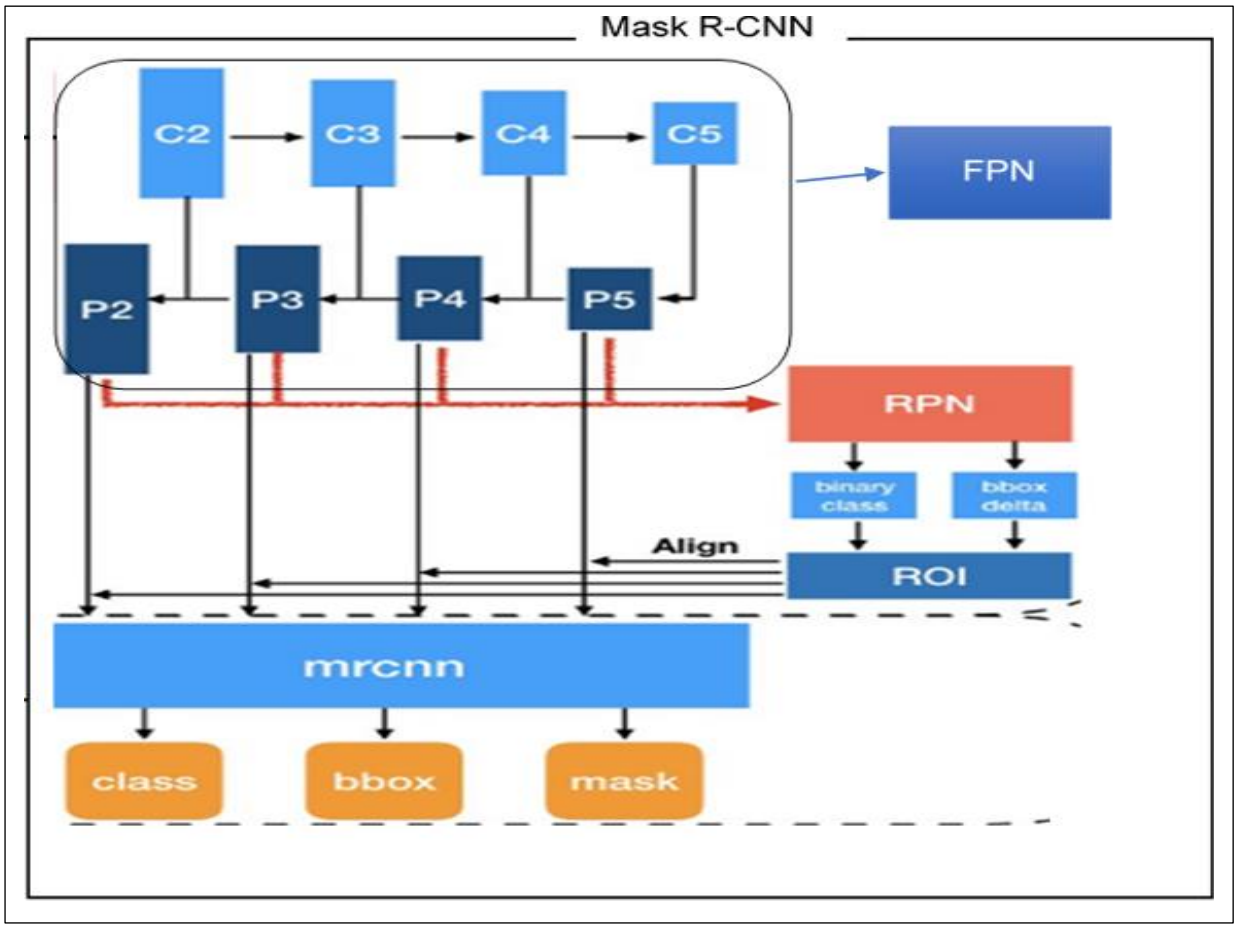


Figure 4.3: Mask R-CNN

➢ Feature extraction

One of the biggest problems in the sputum image for TB is the fact that the bacteria object has different scales, sizes, and features because of these, different objects are characterized by different features. Thus, it is difficult to use certain features to represent objects which make the feature extraction very difficult and leads to bad or low feature extraction and could ruin a model. Because the feature extraction has a direct effect on the overall performance of the model, we need to carry out multilevel convolution and pooling operations on the entire image to extract deep semantic features of the image so the detection can work for big object and as well for small objects that have low resolution. We chose to use FPN as the backbone for our network because it can overcome this problem.

After we fit the model, we first extract features using FPN. We use the bottom-up pathway as a convolutional network for feature extraction then we decrease the spatial resolution while we detect high-level structures and use those features to increase the semantics value of each layer. We use 5 convolutional layers with double the stride for each layer (2,4,8,16,32). We label the output of each convolutional layer as C and used it in the top-down pathway. In the top-down pathway, we first apply a 1 × 1 convolution filter to reduce C channel depth and we create Feature Map M. Then we merge all the layers and we reduce the aliasing effect after merging by applying a 3 × 3 convolution to all merged layers.

We do feature extraction at different convolutional levels of the object by using multiscale convolution to detect smaller objects. The model can detect objects at different levels of the pyramid thus allowing us to detect objects across a large range of scales successfully overcoming the problems of scalability.

➢ Region of interest proposal

As stated before in chapter 2 section 2.4.1, conventional CNN struggles with input that has multiple objects in the view field, to solve this issue the current works that used CNN in TB detection had to do manual intrusion in their model and get involved by manually extracting and cropping part of the image that contains the object exactly. Since we aiming for a fully automated process, we use Region proposal Network (RPN) to overcome this issue.

We use RPN to calculate and produce Regions of Interest (ROIs), which are boxes that may contain an object(bacillus) so we can feed those regions into our network's classifier head to classify as bacillus or not. This will allow us to find multiple regions in every input and be able to detect multiple bacilli in the image. The RPN will generate proposals or regions that may contain a bacillus object.

After the FPN step is finished, we will have the Feature Map that contains all the features learning by the FPN in a pyramid fashion. We give the feature map to the RPN.

In RPN we start by looping over the feature pyramid by using a 3x3 sliding window and we filter every output resolution in the pyramid using two fully connected layers. The first layer is used for box regression, it will create an anchor box over the regions that are suspected to have objects, then we use the second layer for box classification where it takes the boxes generated in the first layer and checks whether the box contains an object or not. We calculate the box offsets and a confidence score for the probability of the box containing an object.

➢ Classification and segmentation

After RPN finishes running, we use the output of RPN as input for RolAlign. We use the rolling to compute each sampling point using bilinear interpolation on the feature map. No quantization is performed on any coordinates involved in the RoI, its bins, or the sampling points. Rather properly aligning the extracted features with the input. We use bilinear interpolation to compute the exact values of the input features at four regularly sampled locations in each ROI bin and aggregate the result using max or average.

After the ROIalign layer locates all the relevant areas in the feature maps and generates ROI and Once the ROI has all the same dimensions, the ROI is used as the input for the last module of the Mask R-CNN the head of the network, where we parallelly perform the final tasks of bounding box recognition, regression, classification, and masks prediction.

We generate the masks by using 3 fully-convolutional neural and the mask size is 56 x 56, The generated masks are soft mini masks, represented by floating-point numbers, where each pixel in the mask denotes the probability of pixel belonging to the background or the TB class, thus holding more details even though they are small. We scale up the masks using 1 convolutional layer to fit the object size in the original image during inference.

4.3.2 Hungarian algorithm

We use the Hungarian algorithm to create and find a bipartite matching between the ground-truth and predictions generated from Mask R-CNN. This way permutation-invariance is enforced and each target element will have a unique match.

After we trained the Mask R-CNN we got a fixed-size set of N predictions, where N is larger than the actual number of objects in the image. Each element in the set of N contains (class, position, size) concerning the ground truth. We used the Hungarian algorithm to create an optimal bipartite matching between predicted and ground-truth objects, and then optimize object-specific losses.

Taken N which denotes the number of predictions, we then generate a set called Z from the ground truth images which are the set of objects containing the actual number of the objects in the image. Since the predicted objects will be more than the actual number of the objects in the image, we add an empty object with the same meta-data as the extra objects in N to Z, so that we can account for class imbalance and perform1 to 1 matching.

Elements in the set N, Z can be as (C, M) where C is the target class and M is a vector that contains the bounding box and ground truth masks center coordinates and its height and width relative to the image size e.g. (“TB”, (X, Y, size)). Then we find a bipartite matching for these

two sets by searching for a permutation of N elements and calculate a matching loss that takes into account the class prediction, the resemblance between the predicted masks and ground truth masks, and the bounding box position. we use bipartite matching to find one-to-one matching for predictions.

To calculate the matching loss we used log-probability, we down-weighted the log-probability when the class is empty whether in predicted or in the ground truth images, by factor 5 so we can handle the class imbalance. We make matching loss constant by making the matching between an object and empty not dependent on the prediction and uses the bounding boxes where it will focus on finding if they have the same positioning.

After the comparison is finished, we will have an output of two sets, one set containing all the correctly positive predicted objects (True positive) and another set of objects that were negatively predicted objects (False positives), both sets have meta-data such as bounding box, class, detection confidence score, and the probabilities.

4.3.3 Hard example mining

One of the biggest problems we faced during this thesis, was the imbalance and scalability and lack of uniformity in the dataset. Our training set has a large imbalance between the number of annotated objects(bacillus) and the number of background examples (image regions not belonging to the object class of interest i.e., non-bacillus). It comprises of an overwhelming number of easy examples and a small number of hard ones, which motivated us to use approaches to mine hard examples to enhance the performance of our model. Hard example mining is crucial for our task as we took advantage of it while we trained the model to enhance and increase the learning efficiency and mitigate overfitting.

We start the training using a set of images that contain few or multiple TB (bacilli), and bounding boxes for each, to train the network we need both positive training examples(bacilli) and negative examples(not-bacilli). For each positive, we created a positive training example

by looking inside that bounding box. And for negative examples, we included negative images that don't contain the required object(bacilli) to the training dataset.

After we run the Hungarian algorithm, we take the output of the algorithm and use hard example mining to harvest hard examples. we took the output of the algorithm, we make use of both sets of outputs where we use the correct positively predicted objects set as a positive hard example and we used the set of objects that were negatively predicted objects as the negative hard examples. We mine for both hard examples by using the loss, probabilities, and the confidence score from the Hungarian algorithm, we used the loss and probabilities generated by the algorithm as the threshold for filtrating hard examples. We considered the samples which the Mask R-CNN struggled with and were hard to match as hard examples, we took the positive pairs with a low detection confidence score and negatives pairs with a high detection confidence score. This will help the model account for every different kind of pattern in the data since the sputum images are by nature very different in scales and different features.

Soon after we train the Mask R-CNN model, we run it on the training images again with a sliding window. Then we mine for the hard examples and add it to our training set. Then we retrain the model with the new dataset and it should perform better with this extra knowledge, and not make as many false positives. In summary, we started with a training dataset that contains positive examples and a random set of negative examples. We then trained the model to convergence on that dataset and subsequently applied the Hungarian algorithm which helped us mine and harvest hard examples. We added those hard examples to the training dataset and train the model again.

Chapter 5: Experiment

This chapter explains the evaluation of TB segmentation using MASK R-CNN implementation explained in Chapter Four. In the next subsections, the data set used for implementation, the environments in which the model was tested, and the outcomes of the proposed model are discussed as taken from the experiments we have conducted.

5.1 Dataset

All images used for training and evaluation are taken from Ziehl–Neelsen Sputum smear Microscopy image DataBase (ZNSM-iDB) [27]. ZNSM-iDB is a repertoire of diverse smear microscopy digital images obtained from three different microscopes including one using Smartphone. This database assists to develop automated algorithms in the following domains:

1. Autofocusing of a view field
2. Auto stitching of view fields to get a panoramic view of Ziehl–Neelsen (ZN) stained slide
3. Detection and grading of Mycobacterium tuberculosis bacilli for automatic detection of tuberculosis (TB)

The database consists of multiple divisions, each acquired with different microscopes. The first using Labomed Digi 3 digital microscope with an iVu 5100 digital camera 5.0 megapixel (MS-1), and the second using Motic BA210 digital microscope with a Siedentopf type binocular. We took our dataset from these two divisions.

This database was developed for the sole purpose of facilitating more research towards automating the detection and diagnosis of TB detection so the disease can be managed or controlled by using efficient diagnosis.

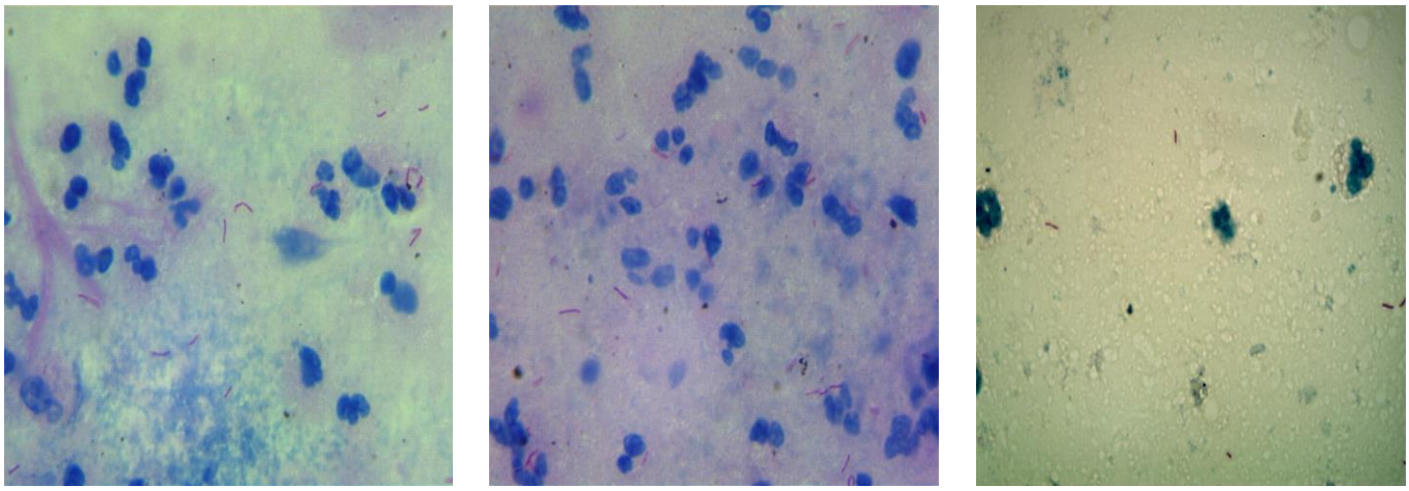


Figure 5.1: Sample of the positive images used

Figure 5.1 shows a sample of positive images that contain bacillus objects that are taken from our training dataset

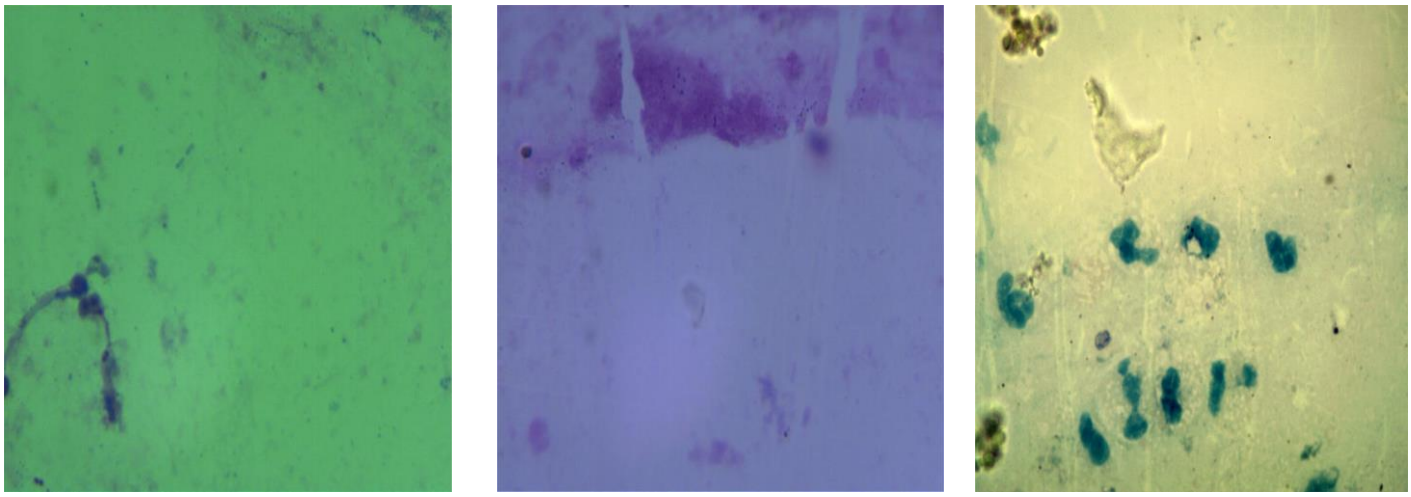


Figure 5.2: Sample of the negative images used

Figure 5.2 shows a sample of negative images that doesn’t contain bacillus and contain debris artifacts objects that are taken from our training dataset

5.2 Development tools and experimental environment

4.2.1 Tools and programming languages

For the development of the proposed model, we used python mainly because of python's simplicity, wide range of third-party libraries available that is tailored for the use in the deep learning process, outstanding readability, and compatibility with the major platforms and systems that make the process of coding simpler and bring portability.

All the steps of implementation are performed with the python back end TensorFlow framework for deep learning. we used python language deep learning libraries such as TensorFlow, Keras, and Sklearn and image processing libraries such as OpenCV, Matplotlib. TensorFlow is good for performing the high-performance calculation, and with very easy optimization of code. Keras is an open-source high-level API for neural networks. It runs on top of TensorFlow. Keras is user-friendly, it supports modularity and it is easily extensible and easy to use because it is written in python. The different module in this Mask R-CNN such as neural network, cost function, activation function, regularization is all done in TensorFlow and Keras.

We used the implementation of MASK R-CNN by the Facebook research team [81] in PyTorch as a baseline for our implementation.

4.2.2 Environment setup

All the experiments were done on a laptop with the following capabilities:

Table 5.1: Hardware Specifications

Category	description
Operating System	Ubuntu 20.4 x64
Processor	Intel® Core™ i7-9750H processor Hexa-core 2.60 GHz
Graphics	NVIDIA® GeForce RTX™ 2060 with 6 GB dedicated memory
Memory	16 GB, DDR4 SDRAM
Storage	512 GB SSD

5.3 Evaluation method

for validating our experimental results, we conducted quantitative evaluations using three commonly used metrics in previous studies: - Precision, Recall, and F1-Score. These metrics are widely used in the field of computer vision and medicine for determining performance.

To calculate these metrics, we first calculated True Positives, False Positives, and False Negatives.

We used the ground truth data to calculate the precision, recall, and F-Score according to the formulas defined in Chapter 2.

5.4 Training

To fit, train, and optimize deep learning models, a vast range of parameters are needed to be figured out. It is very complex and time-consuming to find the best and optimal parameter because they require attention to details and careful analysis of every batch result during training. Different combinations of hyperparameters should be assessed before selecting the optimal hyperparameters. Hyper-parameters are parameters that are not used for learning rather as an input to the model.

The proposed model explained in Chapter 4 is trained for segmenting the pixels into two classes which are TB and background. The input to the model is an image with an input size of 512×512×3, all images have 3 channels and are RGB. We scaled the images to size of 512 x 512 before being fed as input to the model. To resize the input images, we used the crop resize function in the python imaging library.

We further curb overfitting and improve the robustness of the model by using data augmentation techniques, because data augmentation techniques used for a training dataset must be chosen carefully and within the context of the training dataset, we experimented with different data augmentation techniques in isolation with small data portion to see how the augmentation affects the dataset, the model and the training in general. After we finished our little experimentation, we chose to augment the dataset by applying vertical and horizontal reflections, flipping up and down, rotation of 180 degrees, and gaussian blur to images in the training set and extend the number of training samples, we used the augmentation library in [82].

We train the network for 75 epochs with a mini-batch size of 8. Several key hyper-parameters have been tuned in the Mask R-CNN, such as anchor box scales (4,8,18,22,32), minimum detection confidence of 0.8, and mask shape of 56x56. The objective is to find the parameters that could minimize the loss function as much as possible. The validation is run after the end

of each training epoch, we used validation mainly to see if the model is starting to overfit, and if so from which epoch the overfitting starts from so we can take actions against it.

We used Stochastic gradient descent (SGD) as the optimizer of the network as it has been demonstrated by [75] that it generalizes well and converges faster compared to other adaptive optimization techniques. We experimented with the learning rate and weight decay and found that 0.01 and 0.0001 is the optimal learning rate and weight decay respectively to train the network with, we used a learning momentum of 0.9.

We applied Dropout to the model. we defined after the convolutional layers of the head of the network, as it proved experimentally to give small fluctuations in the masks predicted if the image is certain and big differences if it was uncertain. The dropout value we used is 0.5, as it proved to be the optimal one by [83].

We have used 600 images with 4200 (bacilli) objects, selected from all database categories. 80% of the dataset used for training and validation and 20% for testing.

We constructed our deep learning model with callbacks because they help us control, monitor, and improve the training process, we used Keras callbacks like ModelCheckpoint, ReduceLRonPlateau, EarlyStopping.

We used ModelCheckpoint callback to checkpoint our model and save the weights of our model in a checkpoint file in intervals, so the weights can be loaded later to continue the training from the last epoch we saved it in if interrupted or stopped for any reason, we only saved weights in the epochs where the loss function improves i.e. epochs with the lowest validation loss, only the epochs where the loss function stagnates and doesn't improve were ignored.

We also used EarlyStopping, as too many epochs can lead to overfitting on the training dataset and too few may result in underfitting model. We used EarlyStopping to train our model for a large number of epochs and stop the training once the model performance stops improving on

a holdout validation dataset. The Training stops automatically if the validation loss doesn't improve for 8 epochs.

We used ReduceLROnPlateau to reduce the learning rate when there is no improvement in the validation loss for 8 of epochs, as it has been proved that Models cam benefits from reduced learning rate once learning stagnates.

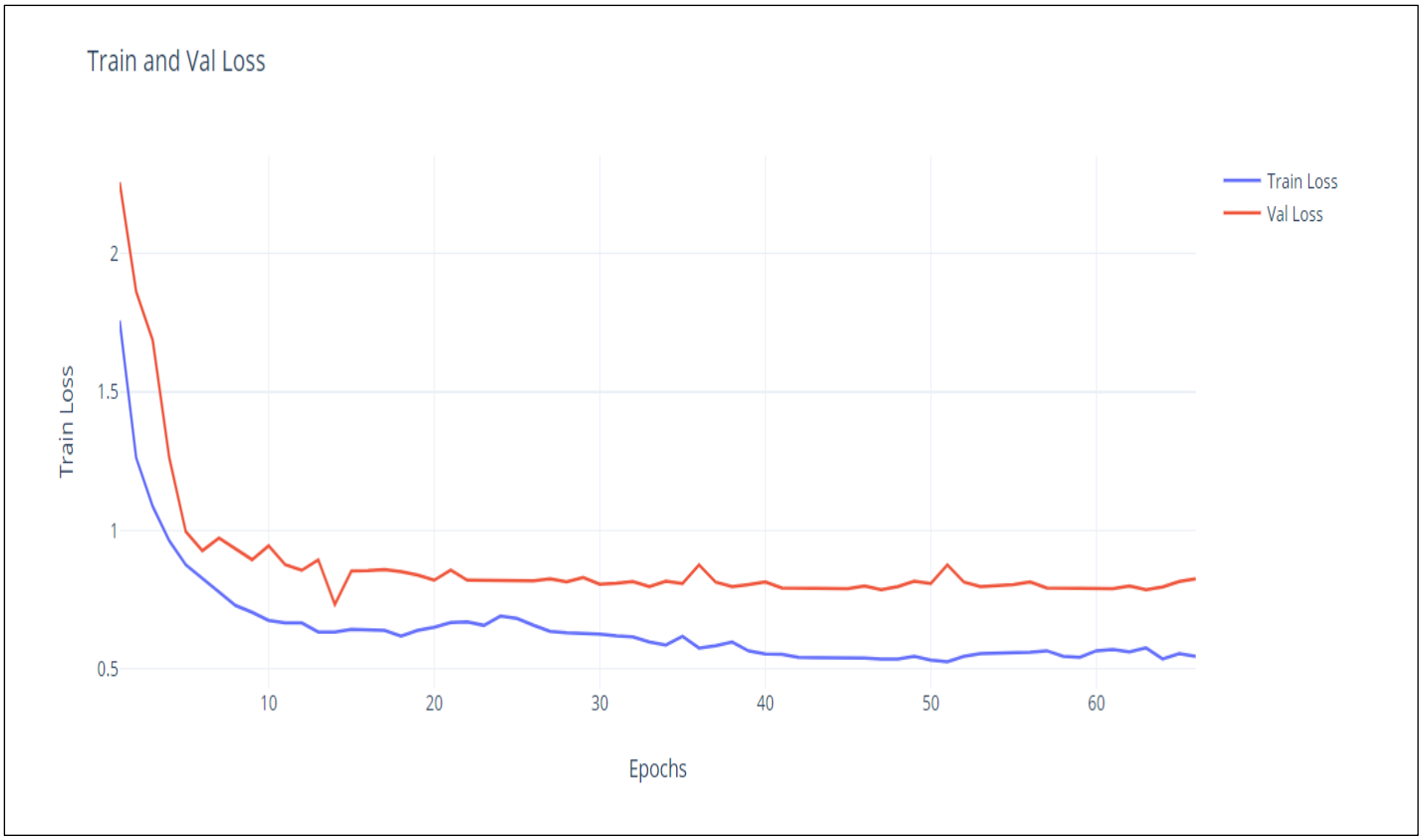


Figure 5.3: Training and Validation loss

Figure 5.4 Shows the training performance of the model over 70 epochs and how the model converges on the training dataset. The figure shows the training and validation loss.

5.5 Test result discussion

The proposed model was implemented and the performance of the model is evaluated. The capabilities of the model to effectively detect bacillus from the sputum and segment the image into bacillus and background is tested. Experiments are conducted to evaluate the performance of the proposed model.

We trained our model multiple times until the desired performance is reached using the training techniques presented in the training subsection. After we successfully train the model, we test the model on the test dataset to see if the model learned and can generalize well. There isn't any configuration required to test the model, we load the weights of the model saved from the training step and load the test dataset and start testing.

We calculated the recall, precision for every and each image in the testing dataset, after each image is predicted we calculated the recall and precision against the ground truth. After we finished running all the images, we get a list of recall and precision values, from those values we calculate the F-score. We take the average of the recall and precision as the total final value of the recall and precision.

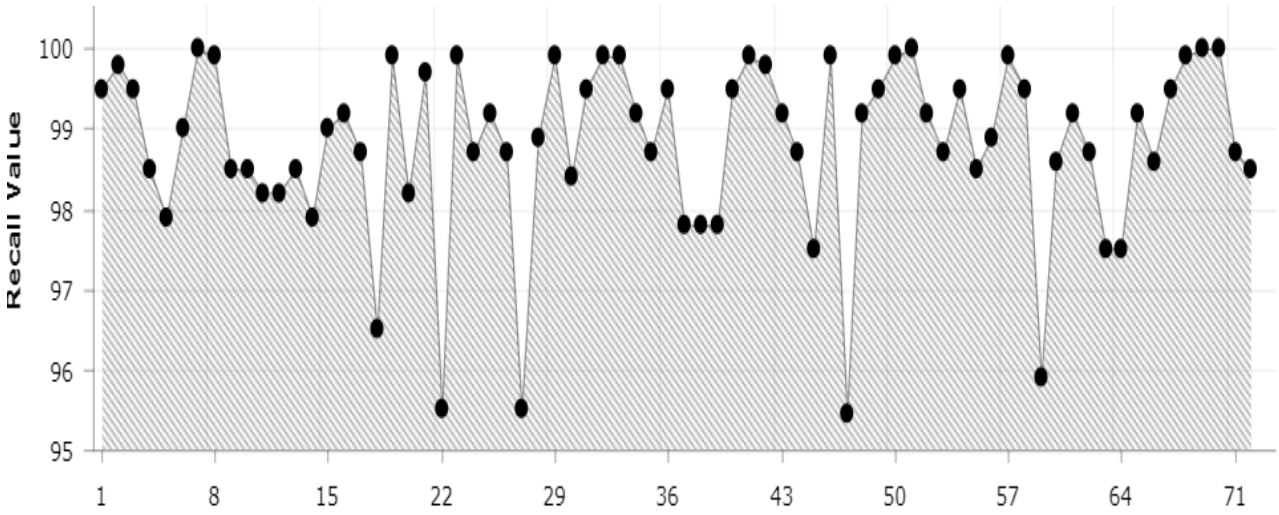


Figure 5.4: Recall values graph plot

Figure 5.5 shows all the recall values for 71 images from the test dataset. Each image has a recall value calculated for it, we take the average of all the images recall value as the model's recall value

we reported the test results obtained from the proposed model. Our model achieved 99.25%, 91.04%, 94.96% recall, precision, and f-score respectively in the detection of TB bacilli in test images on the sputum image dataset.

The recall of the model was obtained to be 99.25%, which affirms that the model is sensitive enough for detecting Tb bacillus, effectively detecting true positives while being effective and accurate in reducing the false positives. The precision values of the model were 91.04%, which suggests that the false positives produced by the models were minimum, small and we were able to enhance and reduce false positives greatly. And a score of 94.96% for F-Score, shows the model performs comprehensively well for the perspective of confidence for bacillus candidates detection, reduction in the number of false positives, and achieving suitable detection of TB bacillus. We can observe that the model demonstrated optimum overall performance.

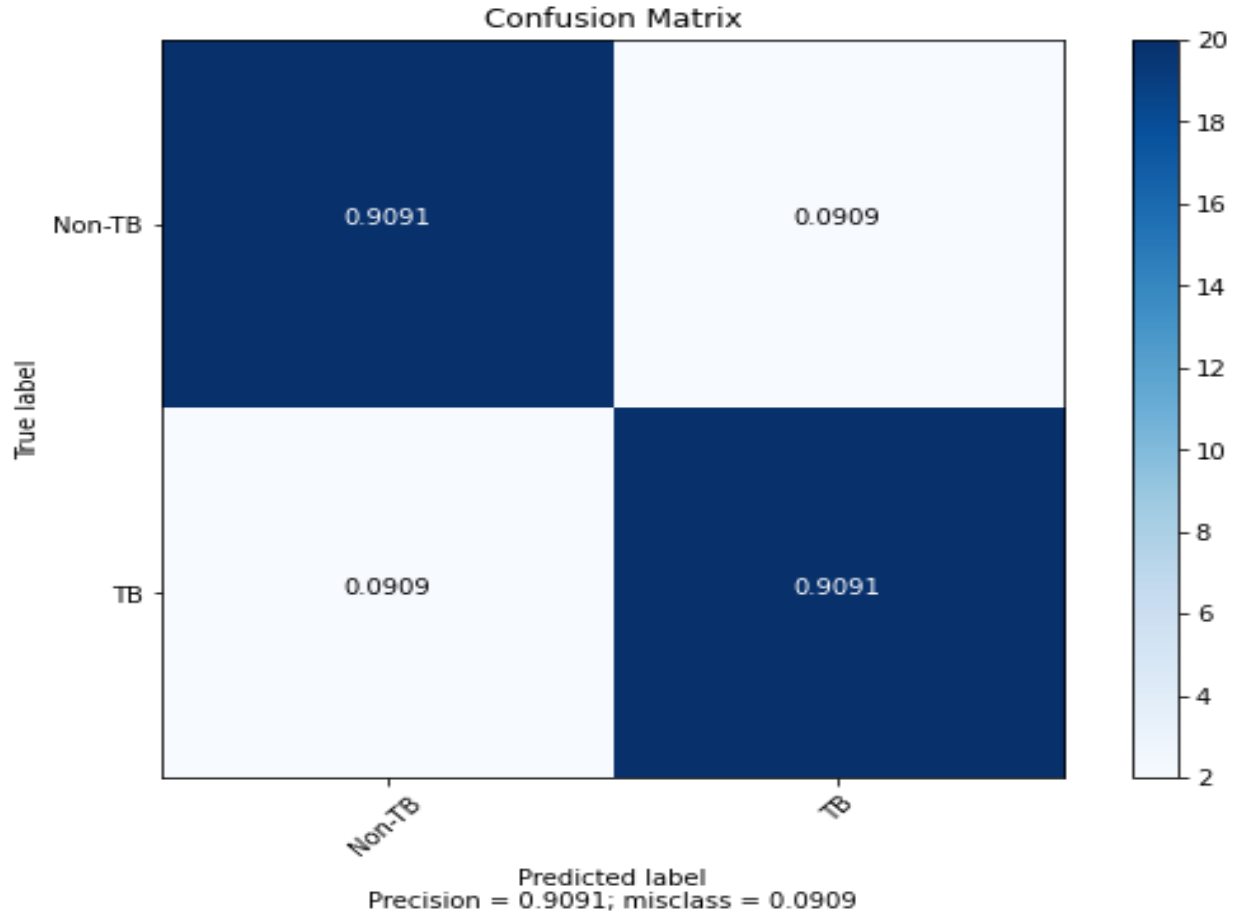


Figure 5.5: Confusion Matrix for one image

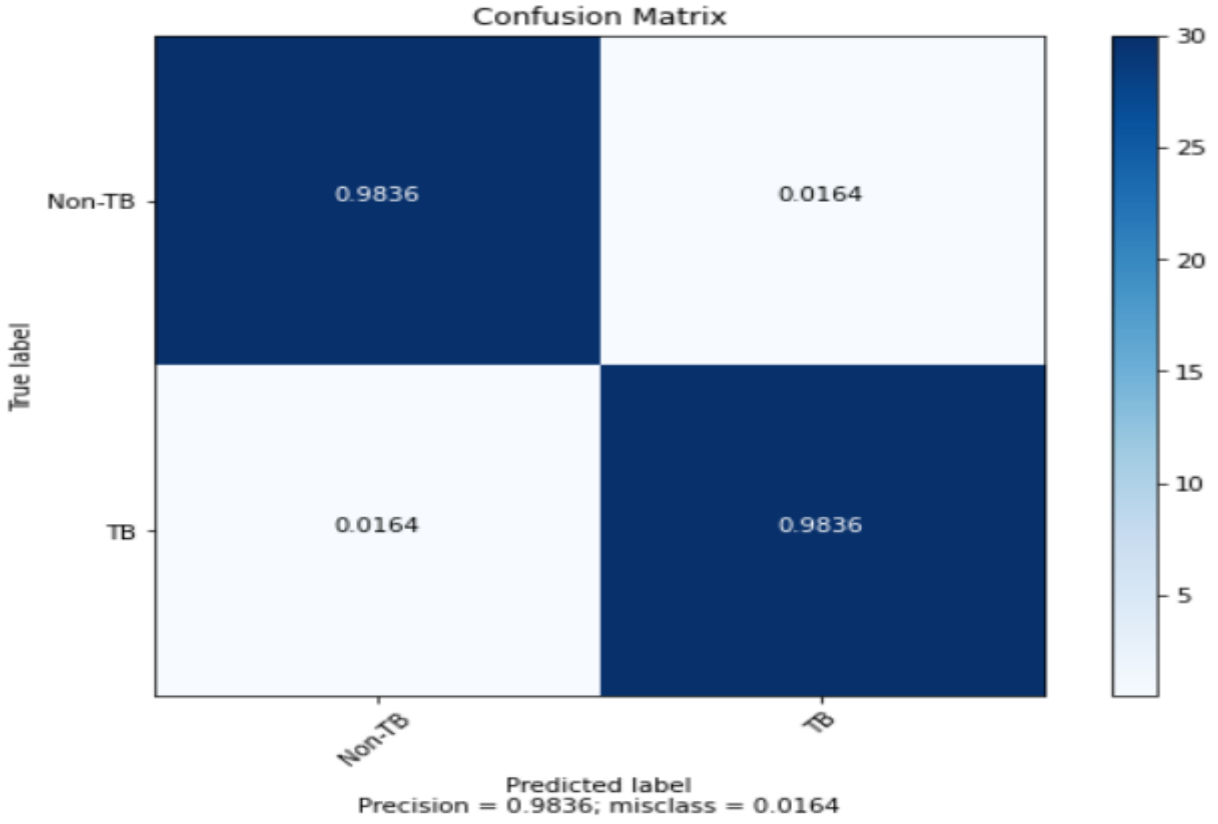


Figure 5.6: Confusion Matrix for one image

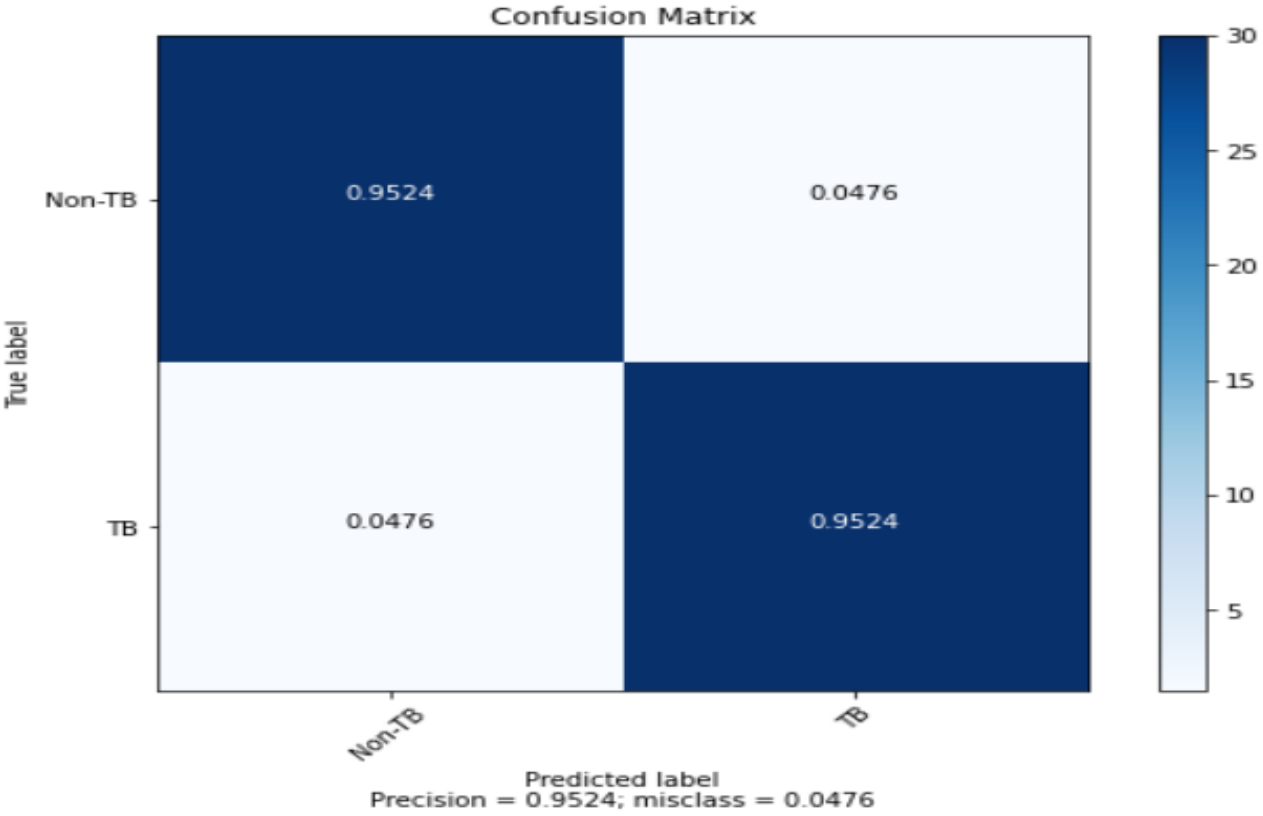


Figure 5.7: Confusion Matrix for one image

Figure 5.6, 5.7, and 5.8 show a confusion matrix for three images randomly selected from the test dataset, the figures show that the model only misclassified a very small class, and the majority of the classes were correctly classified meaning the false positives are minimum and very small.

Based on the results presented here, the proposed model provides a high result for the TB diagnosis system based on sputum images. We compared the proposed model with similarly existing other works.

We only compared against the works that have used either machine or deep learning, all the works that used traditional image processing are ignored. We also ignored all the works that collected their own dataset by themselves, the comparison was done against only the works that have used the same dataset as us, the quantitative comparison results are listed in Table 5.2.

Table 5.2: Comparison between our model and existing works

Algorithm	Recall	Precision	F-score
Faster CNN [84]	98.3%	82.6%	89.7%
Faster CNN + CNN [85]	98.4%	85.1%	91.2%
Ours	**99.25%**	**91.04%**	**94.96%**

With the ability of simultaneous detection and segmentation, our method significantly outperforms all existing works. As can be seen in Table 5.2, our results significantly outperform most of the leading methods. Comparatively, the proposed work is showing an improvement in all three metrics on the sputum image dataset. From these results, it can be seen that in all three scenarios, the model generally had a way better performance than existing works.

the automated detection of tubercle bacilli in sputum stained by auramine stain and screened using a fluorescence microscope appears to be feasible and practical. It could enhance current TB screening programs without dismantling existing infrastructures. From the results shown in Table 5.2, it can be seen that the developed technique appears as a viable solution for the identification of bacilli in sputum samples. This method provides overall better performance in all areas from sensitivity rate, recall, and F-score in comparison with the results reported of the current works. In summary, this preliminary investigation has shown that deep learning image techniques can be successfully used to overcome many issues in the identification of tubercle bacilli in sputum smears.

Chapter Six: Conclusion and recommendation

The final chapter of the document has three sections. In the first section, we provide a brief summary of the activities undertaken in the course of this research work. In the second section of the chapter, the contributions, and achievements of the research work are outlined to show the importance of the work we have achieved so far. In the final section of the chapter, future works that we were not able to extend because of different factors are specified that can lead to the improvement of the work's different aspects.

6.1 Conclusion

Tuberculosis is an airborne infectious disease caused by a bacterial infection. Tuberculosis poses a large problem in low-income countries and developing countries, it is the single culprit behind most of the death among individuals aged fifteen to forty-nine years. In this thesis, we proposed an M. tuberculosis identification model using deep learning. The model has 3 components namely Mask R-CNN, Hungarian algorithm, and Hard example mining. The model first detects plenty of candidates and further classifies them precisely.

Because we were aspiring for minimum manual intervention, we used convolutional neural network methods in our proposed model to construct the training and testing. In our experimentation, we compared the detection performance of our model among the best of the current work. The presented automatic TB detection method is based on Mask R-CNN using microscopic sputum smear images. These methods can be incorporated into an automated microscope for detecting TB disease more accurately within a short duration than manual detection.

The proposed method was experimentally evaluated and quantitative evaluation of segmentation and classification results for bacilli in ZN-stained sputum smear images is reported, we obtained 99.25%, 91.04%, 94.96% for recall, precision, and f-score respectively.

This automatic TB detection can act as a companion to clinicians in the rural and urban areas of high TB burden countries where there is a lack of properly trained technicians.

The proposed model can classify and segment TB bacillus, Furthermore, it can not only localize each bacillus but it can also create affected pixels instance segmentation, generating both bounding box and mask of the bacillus that gives the specific and exact structure of each bacteria in the image.

The model can be used in real-time as it doesn't require any manual interference and doesn't need a computer technical expert, the model can simply take an image, classify, and segment TB bacillus. Furthermore, each bacillus is localized and affected pixels are instance segmented, generating both bounding box and mask of the bacillus that gives the specific and exact structure of each bacteria in the image.

6.2 Contribution of the thesis

The contributions of this thesis work are summarized as follows:

- We proposed a model for diagnosis and instance segmentation of TB bacillus from sputum microscopic images.
- We adopted and modified MASK-RCNN to work efficiently for TB bacillus detection and segmentation from sputum microscopic images.
- We automated the hard example mining algorithm.
- We assess and compare our work against the current best existing methods.

6.3 Future work

We have adopted and designed a deep learning model for the diagnosis of TB from sputum microscopic images and achieved an encouraging result. In this thesis, although we have achieved good results, it could still be improved. There are still several issues regarding the diagnosis of TB that warrant further research.

In the future, underway directions are to increase the size of the data used for the deep network training and preparing it well, developing a specific algorithm that is only used for the overlapping bacillus in such a way that it should be able to handle it better while still handling

single bacillus, and using some sort of morphological operation such as Grabcut on the data to enhance the segmentation.

References

[1] E. Purwanti and P. Widiyanti, "USING LEARNING VECTOR QUANTIZATION METHOD FOR AUTOMATED IDENTIFICATION OF MYCOBACTERIUM TUBERCULOSIS," *Indones. J. Trop. Infect. Dis.*, vol. 3, no. 1, p. 26, Jul. 2015, doi: 10.20473/ijtid.v3i1.198.

[2] M. M. Johnson and J. A. Odell, "Nontuberculous mycobacterial pulmonary infections," *Journal of Thoracic Disease*, vol. 6, no. 3. Pioneer Bioscience Publishing, pp. 210–220, Mar. 01, 2014, doi: 10.3978/j.issn.2072-1439.2013.12.24.

[3] A. Konstantinos, "Testing for tuberculosis," *Australian Prescriber*, vol. 33, no. 1. Australian Government Publishing Service, pp. 12–18, 2010, doi: 10.18773/austprescr.2010.005.

[4] W. H. O. (WHO), "Global tuberculosis report 2016," 2016.

[5] K. Rawat and K. Burse, "A Soft Computing Genetic-Neuro fuzzy Approach for Data Mining and Its Application to Medical Diagnosis," *undefined*, no. 1, pp. 409–411, 2013.

[6] W. H. O. (WHO), "Global Tuberculosis Report 2019," 2018.

[7] Y. Payasi and S. Patidar, "Diagnosis and counting of tuberculosis bacilli using digital image processing," in *IEEE International Conference on Information, Communication, Instrumentation and Control, ICICIC 2017*, 2018, vol. 2018-Janua, pp. 1–5, doi: 10.1109/ICOMICON.2017.8279128.

[8] M. C. Raviglione, *Reichman and Hershfield's tuberculosis: A comprehensive, international approach, third edition*. 2006.

[9] "Tuberculosis." retrieved from https://www.who.int/news-room/fact-sheets/detail/tuberculosis, (Last accessed on September 03, 2020).

[10] S. Hirpa, G. Medhin, B. Girma, M. Melese, A. Mekonen, P. Suarez, and G. Ameni, "Determinants of multidrug-resistant tuberculosis in patients who underwent first-line treatment in Addis Ababa: A case control study," *BMC Public Health*, vol. 13, no. 1, p. 782, Dec. 2013, doi: 10.1186/1471-2458-13-782.

[11] C. F. F. Costa Filho, P. C. Levy, C. de Matos Xavier, L. B. Mendonça Fujimoto, and M. G. Fernandes Costa, "Automatic identification of tuberculosis mycobacterium," *Rev. Bras. Eng. Biomed.*, vol. 31, no. 1, pp. 33–43, 2015, doi: 10.1590/2446-4740.0524.

[12] E. Priya and S. Srinivasan, "Separation of overlapping bacilli in microscopic digital TB images," *Biocybern. Biomed. Eng.*, vol. 35, no. 2, pp. 87–99, 2015, doi:

10.1016/j.bbe.2014.08.002.

[13] M. G. Forero and G. Crist, “Automatic identification techniques of tuberculosis bacteria,” no. May 2014, 2003, doi: 10.1117/12.506800.

[14] K. Veropoulos, G. Learmonth, C. Campbell, B. Knight, and J. Simpson, “Automated identification of tubercle bacilli in sputum: A preliminary investigation,” *Anal. Quant. Cytol. Histol.*, vol. 21, no. 4, pp. 277–282, 1999.

[15] J. A. Quinn, R. Nakasi, P. K. B. Mugagga, P. Byanyima, W. Lubega, and A. Andama, “Deep Convolutional Neural Networks for Microscopy-Based Point of Care Diagnostics,” *jmlr.org*, 2016, Accessed: Sep. 14, 2020. [Online]. Available: http://www.jmlr.org/proceedings/papers/v56/Quinn16.pdf.

[16] Y. P. López, C. F. F. Costa Filho, L. M. R. Aguilera, and M. G. F. Costa, “Automatic classification of light field smear microscopy patches using Convolutional Neural Networks for identifying Mycobacterium Tuberculosis,” in *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies, CHILECON 2017 - Proceedings*, 2017, vol. 2017-Janua, pp. 1–5, doi: 10.1109/CHILECON.2017.8229512.

[17] S. Kant and M. M. Srivastava, “Towards Automated Tuberculosis detection using Deep Learning,” in *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018*, Jan. 2019, pp. 1250–1253, doi: 10.1109/SSCI.2018.8628800.

[18] R. O. Panicker, K. S. Kalmady, J. Rajan, and M. K. Sabu, “Automatic detection of tuberculosis bacilli from microscopic sputum smear images using deep learning methods,” *Biocybern. Biomed. Eng.*, vol. 38, no. 3, pp. 691–699, 2018, doi: 10.1016/j.bbe.2018.05.007.

[19] S. Godreuil, G. Torrea, D. Terru, F. Chevenet, S. Diagbouga, P. Supply, P. Van De Perre, C. Carriere, and A. L. Bañuls, “First Molecular Epidemiology Study of Mycobacterium tuberculosis in Burkina Faso,” *J. Clin. Microbiol.*, vol. 45, no. 3, pp. 921–927, 2007, doi: 10.1128/JCM.01918-06.

[20] R. Khutlang, S. Krishnan, R. Dendere, A. Whitelaw, K. Veropoulos, G. Learmonth, and T. S. Douglas, “Classification of mycobacterium tuberculosis in images of ZN-stained sputum smears,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 4, pp. 949–957, 2010, doi: 10.1109/TITB.2009.2028339.

[21] S. R. Reshma and T. Rehannara Beegum, “Microscope image processing for TB

diagnosis using shape features and ellipse fitting," in *2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems, SPICES 2017*, 2017, pp. 1–7, doi: 10.1109/SPICES.2017.8091342.

[22] FMOH, "Implementation Guideline for TB / HIV Collaborative Activities in Ethiopia," no. December, 2007.

[23] J. M. Grange, N. Kapata, D. Chanda, P. Mwaba, and A. Zumla, "The biosocial dynamics of tuberculosis," *Tropical Medicine and International Health*, vol. 14, no. 2. pp. 124–130, 2009, doi: 10.1111/j.1365-3156.2008.02205.x.

[24] F. J. Yammarino and B. M. Bass, "Transformational Leadership and Multiple Levels of Analysis," *Hum. Relations*, vol. 43, no. 10, pp. 975–995, Oct. 1990, doi: 10.1177/001872679004301003.

[25] H. F. Swai, F. M. Mugusi, and J. K. Mbwambo, "Sputum smear negative pulmonary tuberculosis: Sensitivity and specificity of diagnostic algorithm," *BMC Res. Notes*, vol. 4, pp. 2–7, 2011, doi: 10.1186/1756-0500-4-475.

[26] R. A. KAUR, "Tuberculosis control in India.," *J. Christ. Med. Assoc. India*, vol. 25, no. 3, pp. 156–160, 1950, Accessed: Sep. 03, 2020. [Online]. Available: https://pdfs.semanticscholar.org/0eaf/20ddbffe4140add6ed37ba913e27d196e7c0.pdf#page=142.

[27] M. I. Shah, S. Mishra, V. K. Yadav, A. Chauhan, M. Sarkar, S. K. Sharma, and C. Rout, "Ziehl–Neelsen sputum smear microscopy image database: a resource to facilitate automated bacilli detection for tuberculosis diagnosis," *J. Med. Imaging*, vol. 4, no. 2, p. 027503, Jun. 2017, doi: 10.1117/1.jmi.4.2.027503.

[28] D. Dubey, S. Rath, M. C. Sahu, N. K. Debata, and R. N. Padhy, "Antimicrobials of plant origin against TB and other infections and economics of plant drugs -Introspection," *Indian J. Tradit. Knowl.*, vol. 11, no. 2, pp. 225–233, 2012.

[29] Z. Taylor, C. M. Nolan, and H. M. Blumberg, "Controlling tuberculosis in the United States. Recommendations from the American Thoracic Society, CDC, and the Infectious Diseases Society of America.," *MMWR. Recomm. Rep.*, vol. 54, no. RR-12, pp. 1–81, 2005.

[30] M. L. Giger, H. P. Chan, and J. Boone, "Anniversary paper: History and status of CAD and quantitative image analysis: The role of Medical Physics and AAPM," *Med. Phys.*, vol. 35, no. 12, pp. 5799–5820, 2008, doi: 10.1118/1.3013555.

[31] J. Yanase and E. Triantaphyllou, “A systematic survey of computer-aided diagnosis in medicine: Past and present developments,” *Expert Systems with Applications*, vol. 138, no. February 2020. 2019, doi: 10.1016/j.eswa.2019.112821.

[32] B. J. Erickson and B. Bartholmai, “Computer-aided detection and diagnosis at the start of the third millennium,” *J. Digit. Imaging*, vol. 15, no. 2, pp. 59–68, 2002, doi: 10.1007/s10278-002-0011-x.

[33] A. Kumar, “An Overview on Deep Learnings and its Accomplishments in Computer Vision,” *Res. Rev. Int. J. Multidiscip.*, vol. 3085, no. 05, pp. 1182–1185, 2019.

[34] J. Gao, Y. Yang, P. Lin, and D. S. Park, “Editorial Computer Vision in Healthcare Applications,” vol. 2018, 2018.

[35] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep Learning for Computer Vision: A Brief Review,” *Computational Intelligence and Neuroscience*, vol. 2018. 2018, doi: 10.1155/2018/7068349.

[36] E. Cetinic, T. Lipic, and S. Grgic, “Learning the Principles of Art History with convolutional neural networks,” *Pattern Recognit. Lett.*, vol. 129, pp. 56–62, 2020, doi: 10.1016/j.patrec.2019.11.008.

[37] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, “A survey on deep learning techniques for image and video semantic segmentation,” *Applied Soft Computing Journal*, vol. 70. Elsevier B.V., pp. 41–65, 2018, doi: 10.1016/j.asoc.2018.05.018.

[38] Á. Casado-García, C. Domínguez, M. García-Domínguez, J. Heras, A. Inés, E. Mata, and V. Pascual, “Clodsa: A tool for augmentation in classification, localization, detection, semantic segmentation and instance segmentation tasks,” *BMC Bioinformatics*, vol. 20, no. 1, pp. 1–14, 2019, doi: 10.1186/s12859-019-2931-1.

[39] Z. Zhao and H. Liu, “Spectral feature selection for supervised and unsupervised learning,” *ACM Int. Conf. Proceeding Ser.*, vol. 227, pp. 1151–1157, 2007, doi: 10.1145/1273496.1273641.

[40] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553. pp. 436–444, 2015, doi: 10.1038/nature14539.

[41] A. Shahkarami, S. D. Mohaghegh, V. Gholami, and S. A. Haghighat, “Artificial intelligence (AI) assisted history matching,” Apr. 2014, doi: 10.2118/169507-ms.

[42] F. Xing, Y. Xie, H. Su, F. Liu, and L. Yang, “Deep Learning in Microscopy Image

Analysis : A Survey," pp. 1–19, 2017.

[43] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958, doi: 10.1037/h0042519.

[44] A. C. Ian Goodfellow, Yoshua Bengio, *Deep Learning - Ian Goodfellow, Yoshua Bengio, Aaron Courville - Google Books*. 2016.

[45] M. A. Ponti, L. S. F. Ribeiro, T. S. Nazare, T. Bui, and J. Collomosse, "Everything You Wanted to Know about Deep Learning for Computer Vision but Were Afraid to Ask," *Proc. - 2017 30th SIBGRAPI Conf. Graph. Patterns Images Tutorials SIBGRAPI-T 2017*, vol. 2018-Janua, pp. 17–41, 2017, doi: 10.1109/SIBGRAPI-T.2017.12.

[46] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," Mar. 2016, Accessed: Sep. 03, 2020. [Online]. Available: http://arxiv.org/abs/1603.04467.

[47] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, "Theano: new features and speed improvements," Nov. 2012, Accessed: Sep. 03, 2020. [Online]. Available: http://arxiv.org/abs/1211.5590.

[48] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems," Dec. 2015, Accessed: Sep. 03, 2020. [Online]. Available: http://arxiv.org/abs/1512.01274.

[49] T. Panch, P. Szolovits, and R. Atun, "Artificial intelligence, machine learning and health systems," *J. Glob. Health*, vol. 8, no. 2, pp. 1–8, 2018, doi: 10.7189/jogh.08.020303.

[50] L. Shao, H. P. H. Shum, and T. Hospedales, "Editorial: Special Issue on Machine Vision with Deep Learning," *International Journal of Computer Vision*, vol. 128, no. 4. Springer US, pp. 771–772, 2020, doi: 10.1007/s11263-020-01317-y.

[51] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *J. Physiol.*, vol. 195, no. 1, pp. 215–243, Mar. 1968, doi: 10.1113/jphysiol.1968.sp008455.

[52] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, Apr. 1980, doi: 10.1007/BF00344251.

[53] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.

[54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2012, doi: 10.1145/3065386.

[55] S. Liu and W. Deng, “Very deep convolutional neural network based image classification using small training sample size,” in *Proceedings - 3rd IAPR Asian Conference on Pattern Recognition, ACPR 2015*, Jun. 2016, pp. 730–734, doi: 10.1109/ACPR.2015.7486599.

[56] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[57] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Oct. 2015, vol. 07-12-June, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[58] Y. Bengio, “Learning deep architectures for AI,” *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–27, 2009, doi: 10.1561/2200000006.

[59] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Sep. 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[60] M. Raghu and E. Schmidt, “A Survey of Deep Learning for Scientific Discovery,” pp. 1–48, 2020, [Online]. Available: http://arxiv.org/abs/2003.11755.

[61] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, Dec. 2017, vol. 2017-Octob, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.

[62] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[63] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L.

Zitnick, “Microsoft COCO: Common objects in context,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8693 LNCS, no. PART 5, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[64] P. O. Pinheiro, T. Y. Lin, R. Collobert, and P. Dollár, “Learning to refine object segments,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Mar. 2016, vol. 9905 LNCS, pp. 75–91, doi: 10.1007/978-3-319-46448-0_5.

[65] D. M. Hawkins, “The Problem of Overfitting,” *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 1, pp. 1–12, 2004, doi: 10.1021/ci0342472.

[66] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014, Accessed: Sep. 04, 2020. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html.

[67] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *32nd International Conference on Machine Learning, ICML 2015*, Feb. 2015, vol. 1, pp. 448–456, Accessed: Sep. 04, 2020. [Online]. Available: https://arxiv.org/abs/1502.03167v3.

[68] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, “Efficient object localization using Convolutional Networks,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Oct. 2015, vol. 07-12-June, pp. 648–656, doi: 10.1109/CVPR.2015.7298664.

[69] C. Neubauer, “Evaluation of convolutional neural networks for visual recognition,” *IEEE Transactions on Neural Networks*, vol. 9, no. 4. pp. 685–696, 1998, doi: 10.1109/72.701181.

[70] T. Saito and M. Rehmsmeier, “The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets,” *PLoS One*, vol. 10, no. 3, pp. 1–21, 2015, doi: 10.1371/journal.pone.0118432.

[71] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Nav. Res. Logist. Q.*, vol. 2, no. 1–2, pp. 83–97, Mar. 1955, doi: 10.1002/nav.3800020109.

[72] H. W. Kuhn, “Variants of the hungarian method for assignment problems,” *Nav. Res. Logist. Q.*, vol. 3, no. 4, pp. 253–258, Dec. 1956, doi: 10.1002/nav.3800030404.

[73] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-End Object Detection with Transformers,” 2020, [Online]. Available: http://arxiv.org/abs/2005.12872.

[74] K. Sung, “Learning and Example Selection for Object and Pattern Detection,” *PhD thesis*, p. 195, 1996, doi: https://doi.org/10.1016/j.comnet.2014.12.002.

[75] Pedro F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object Detection with Discriminatively Trained Part Based Models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 2, pp. 6–7, 2010, doi: 10.1109/MC.2014.42.

[76] J. J. Yeh, “Validation of a model for predicting smear-positive active pulmonary tuberculosis in patients with initial acid-fast bacilli smear-negative sputum,” *Eur. Radiol.*, vol. 28, no. 1, pp. 243–256, 2018, doi: 10.1007/s00330-017-4959-9.

[77] J. B. João, J. M. de Seixas, R. Galliez, B. de Bragança Pereira, F. C. de Q Mello, A. M. dos Santos, and A. L. Kritski, “A screening system for smear-negative pulmonary tuberculosis using artificial neural networks,” *Int. J. Infect. Dis.*, vol. 49, pp. 33–39, 2016, doi: 10.1016/j.ijid.2016.05.019.

[78] E. Priya and S. Srinivasan, “Automated object and image level classification of TB images using support vector neural network classifier,” *Biocybern. Biomed. Eng.*, vol. 36, no. 4, pp. 670–678, 2016, doi: 10.1016/j.bbe.2016.06.008.

[79] P. Ghosh, D. Bhattacharjee, and M. Nasipuri, “A hybrid approach to diagnosis of tuberculosis from sputum,” in *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*, 2016, pp. 771–776, doi: 10.1109/ICEEOT.2016.7754790.

[80] C. Xu, D. Zhou, and Y. Liu, “Segmentation of touching mycobacterium tuberculosis from Ziehl-Neelsen stained sputum smear images,” in *MIPPR 2015: Automatic Target Recognition and Navigation*, 2015, vol. 9812, p. 981210, doi: 10.1117/12.2209226.

[81] R. Massa, Francisco and Girshick, “maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch,” 2018. retrieved from https://github.com/facebookresearch/maskrcnn-benchmark, (Last accessed on September 16, 2020).

[82] A. B. Jung, K. Wada, J. Crall, S. Tanaka, J. Graving, C. Reinders, S. Yadav, J. Banerjee, G. Vecsei, A. Kraft, Z. Rui, J. Borovec, C. Vallentin, S. Zhydenko, K. Pfeiffer, B. Cook, and Others, “imgaug: Image augmentation for machine learning experiments.,” 2020.

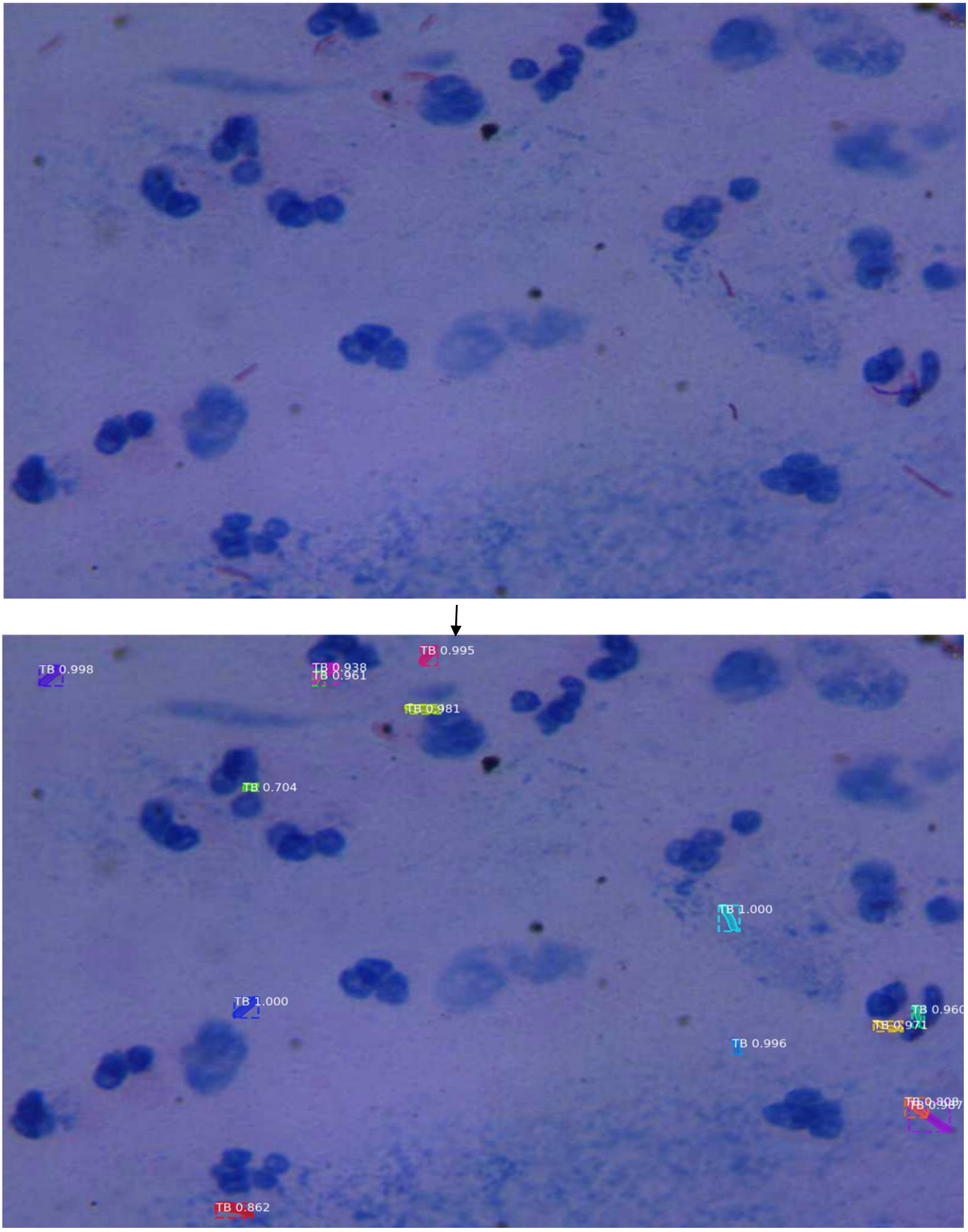
retrieved from https://github.com/aleju/imgaug, (Last accessed on September 23, 2020).

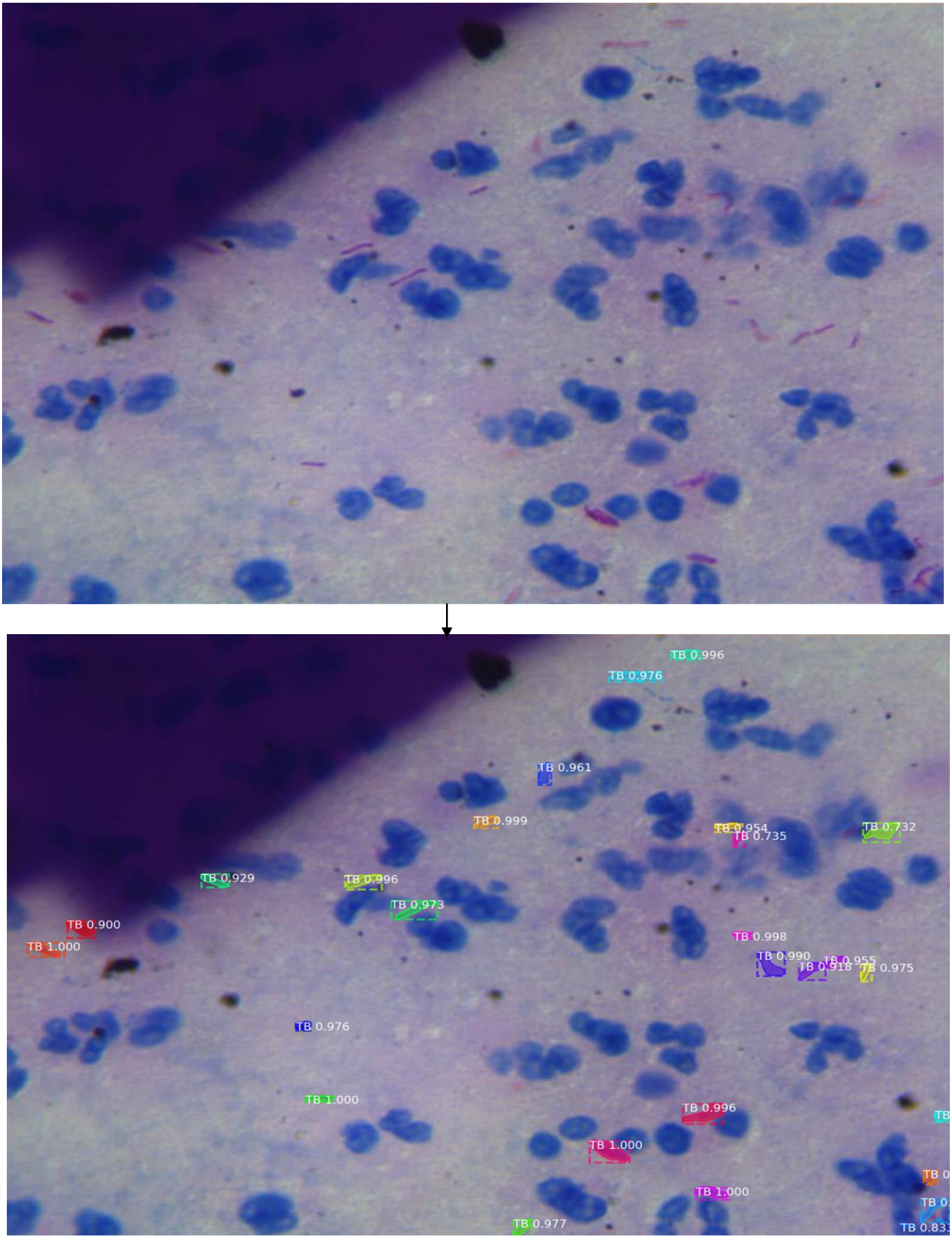
[83] A. Kendall, V. Badrinarayanan, and R. Cipolla, “Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding,” 2017, doi: 10.5244/c.31.57.

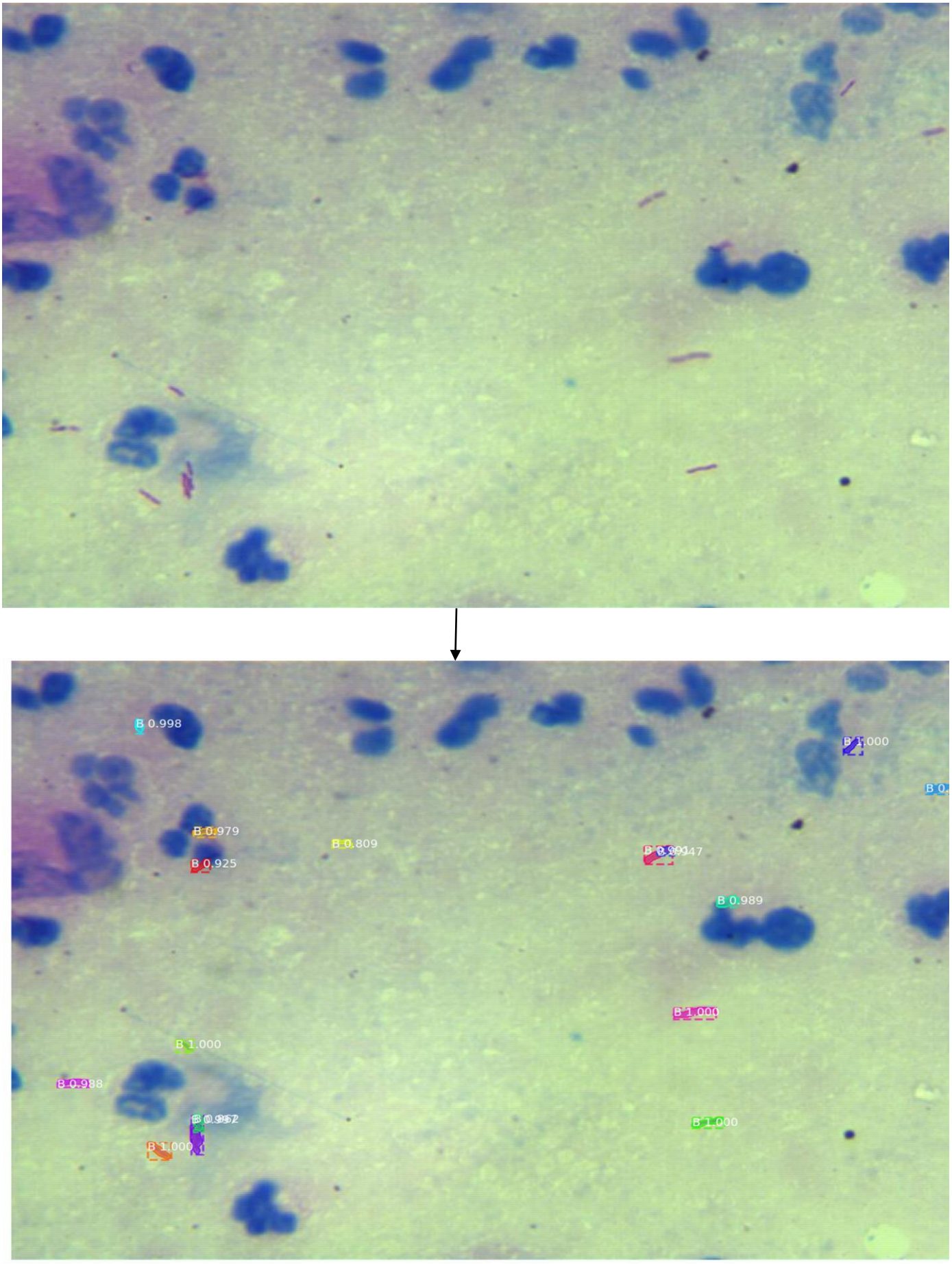
[84] M. El-Melegy, D. Mohamed, T. Elmelegy, and M. Abdelrahman, “Identification of tuberculosis bacilli in ZN-stained sputum smear images: A deep learning approach,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 1131–1137.

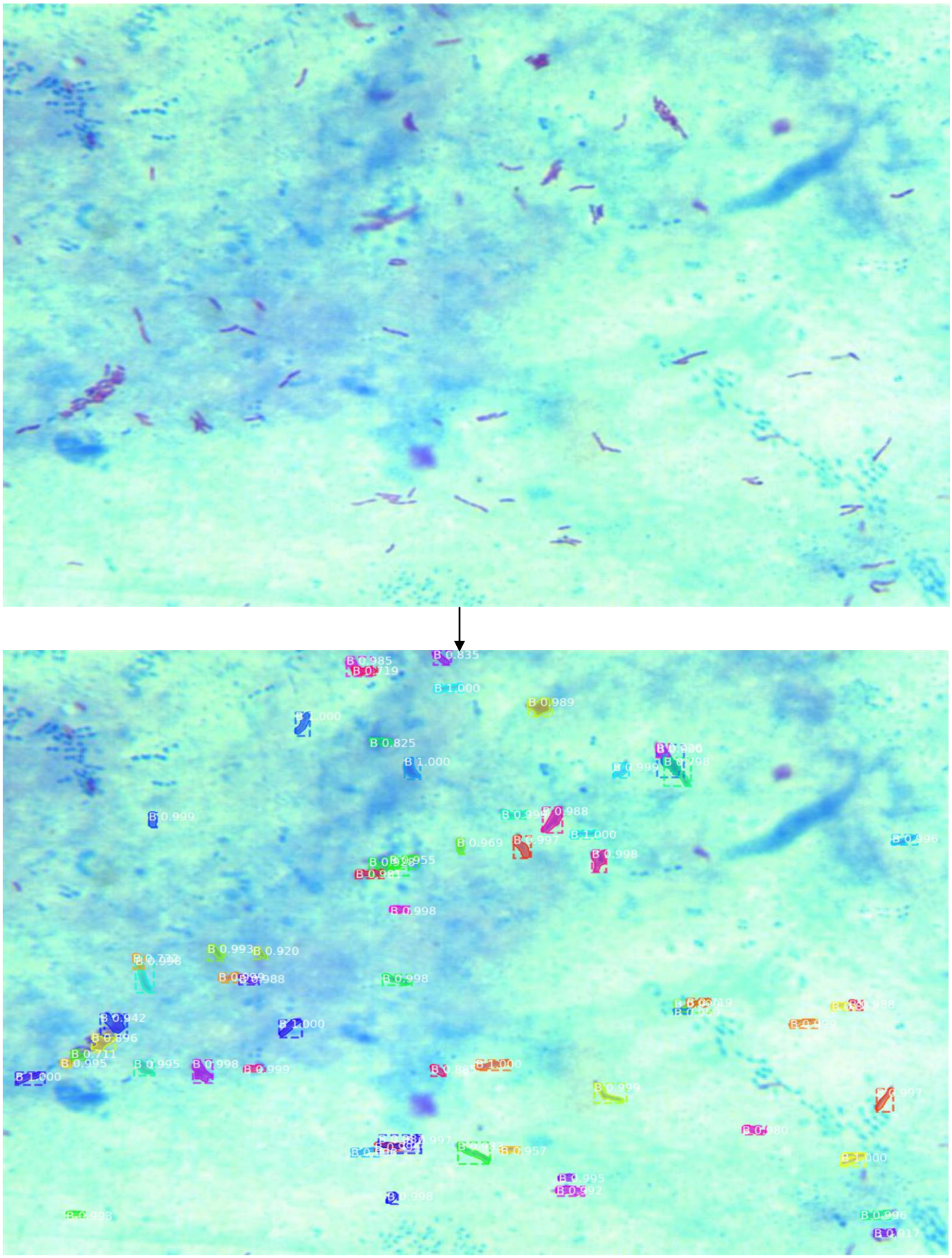
[85] M. El-Melegy, D. Mohamed, and T. ElMelegy, “Automatic Detection of Tuberculosis Bacilli from Microscopic Sputum Smear Images Using Faster R-CNN, Transfer Learning and Augmentation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11867 LNCS, pp. 270–278, doi: 10.1007/978-3-030-31332-6_24.

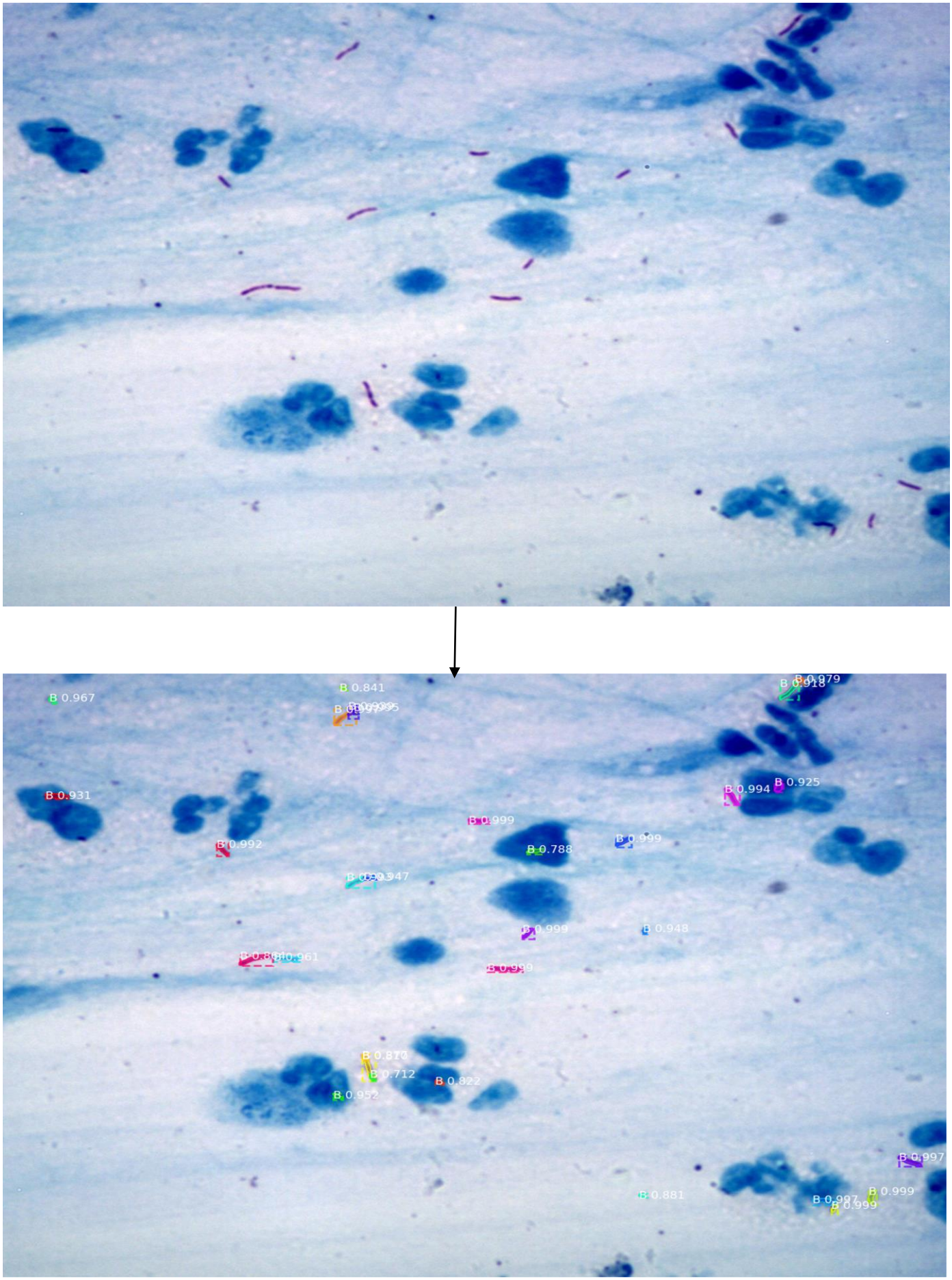
Appendix A: sample result of the model

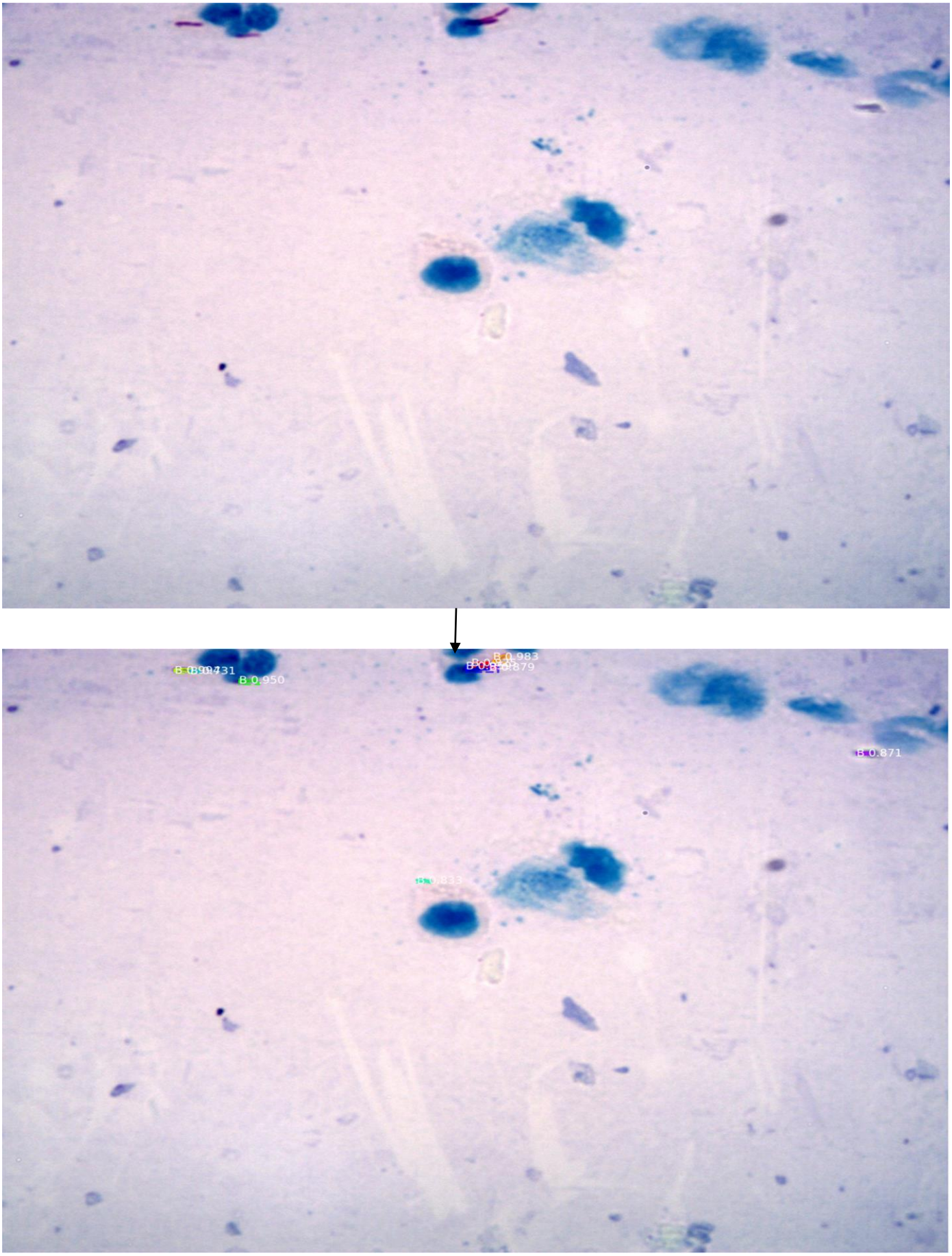



TB 0.996
TB 0.976
TB 0.961
TB 0.999
TB 0.954
TB 0.735
TB 0.732
TB 0.929
TB 0.996
TB 0.973
TB 0.900
TB 1.000
TB 0.998
TB 0.990
TB 0.955
TB 0.918
TB 0.975
TB 0.976
TB 1.000
TB 0.996
TB 1.000
TB 1.000
TB 0.977


B 0.998
B 0.979
B 0.809
B 0.925
B 0.989
B 1.000
B 1.000
B 1.000




B 0.967
B 0.841
B 0.931
B 0.992
B 0.788
B 0.994
B 0.925
B 0.999
B 0.948
B 0.712
B 0.822
B 0.952
B 0.881
B 0.997
B 0.999


B 0.950
B 0.983
B 0.871

Signed Declaration Sheet

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other universities and that all sources of materials used for the thesis have been duly acknowledged.

Declared by:

Name: Ibrahim Muse

Signature: _____

Date: November ,2020

Confirmed by Advisor:

Name: Ayalew Baley (PhD).

Signature: _____

Date: November, 2020