Spring 2020

Automatic Chest X-rays Analysis using Statistical Machine Learning Strategies

Hermann Yepdjio Nkouanga
yepdjionkoh@cwu.edu

AUTOMATIC CHEST X-RAYS ANALYSIS USING STATISTICAL MACHINE LEARNING STRATEGIES

A Thesis

Presented to

The Graduate Faculty

Central Washington University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Computational Science

---

by

Hermann Yepdjio Nkouanga

June 2020

CENTRAL WASHINGTON UNIVERSITY

Graduate Studies

We hereby approve the thesis of

Hermann Yepdjio Nkouanga

Candidate for the degree of Master of Science

APPROVED FOR THE GRADUATE FACULTY

_____	_____
	Dr. Szilárd Vajda
_____	_____
	Dr. Razvan Andonie
_____	_____
	Dr. Donald Davendra
_____	_____
	Dean of Graduate Studies

ABSTRACT

AUTOMATIC CHEST X-RAYS ANALYSIS USING
STATISTICAL MACHINE LEARNING
STRATEGIES

by

Hermann Yepdjio Nkouanga

June 2020

Tuberculosis (TB) is a disease responsible for the deaths of more than one million people worldwide every year. Even though it is preventable and curable, it remains a major threat to humanity that needs to be taken care of. It is often diagnosed in developed countries using approaches such as sputum smear microscopy and culture methods. However, since these approaches are rather expensive, they are not commonly used in poor regions of the globe such as India, Africa, and Bangladesh. Instead, the well known and affordable chest x-ray (CXR) interpretation by radiologists is the technique employed in those places. Nevertheless, if this method is obsolete in other parts of the world nowadays it is because of its many flaws including: i) it is a tedious task that requires experienced medical personnel --which is scarce given the high demand for it--, ii) it is manual and difficult when executed for a large population, and iii) it is prone to human error depending on the proficiency and aptitude of the interpreter. Researchers have thus been trying to overcome these challenges over the years by proposing software solutions that mainly involve computer vision, artificial intelligence, and machine learning. The problems with these existing solutions are that they are either complex or not reliable enough. The need for better solutions in this specific domain as well as my desire to bring my contribution to something meaningful are what led us to investigate in this direction.

In this manuscript, I propose a simple fully automatic software solution that uses only machine learning and image processing to analyze and detect anomalies related to TB in CXR scans. My system starts by extracting the region of interest from the incoming images, then performs a computationally inexpensive yet efficient feature extraction that involves edge detection using Laplacian of Gaussian and positional information retention. The extracted features are then fed to a regular random forest classifier for discrimination. I tested the system on two benchmark data collections --Montgomery and Shenzhen-- and obtained state-of-the-art results that reach up to 97% classification accuracy.

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor Dr. Szilárd Vajda for the continuous support of my research and thesis, for his patience, flexibility, and availability. His experience, vast knowledge, guidance, and advice helped me to quickly progress throughout my study. I am very grateful and I could never have imagined a better advisor.

I would also like to express my gratitude to the other members of my thesis committee: Prof. Razvan Andonie and Prof. Donald Davendra. I greatly value the time and effort that they put in to check my work and provide suggestions and meaningful comments.

I thank all my labmates for making these last two years enjoyable and memorable.

Last but not least, I would also like to thank my parents and siblings for their encouragement, financial and spiritual supports.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

Creating a visual representation of the internal structure of a body also known as medical imaging is a common process necessary to diagnose, monitor, and treat several medical conditions around the world. The term refers to a multitude of technologies that include X-ray, Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Ultrasound, Fluoroscopy, Bone densitometry (DEXA), etc. They all yield a different type of information and their importance depends on the condition and the part of the body that is being analyzed. Though such information is crucial to provide proper care to patients, it remains worthless unless interpreted properly and accurately by qualified medical personnel (see radiologists). However, the demand for the latter highly exceeds the offer in most places, a trend that is unlikely to change given the continuous growth and aging of the population. To adjust this tendency and assure that all body scans get interpretations, researchers have been trying for years to partially or completely replace the human factor with computer programs. Even if several software solutions mainly based on artificial intelligence, machine learning, and computer vision resulted along the way, only very few of them can actually reach or exceed the average performance of radiologists. Yet, these few systems are rather complicated and involve complex machine learning models (see Convolutional Neural Networks (CNN)) that require large numbers of training samples as well as expensive equipment to perform accurately. Consequently, the need for simpler solutions that can perform well with small datasets and simple features is still current, and since no such solution exists to the best of my knowledge, I have attempted to provide one.

In this study, I focus my attention on chest x-ray (CXR) and automatic detection of anomalies within the lungs caused by diseases such as Tuberculosis (TB). TB is a contagious disease caused by Mycobacterium tuberculosis which mainly affects the lungs but can also be found in other parts of the body such as the brain, kidneys, and spine. It usually transmits through the air when a person with active TB sneezes, coughs, or even talks [2]. Africa and Southeast Asia, where poverty and malnutrition have lowered the resistance of humans to the bacteria, are the regions most affected by the disease [3]. According to the World Health Organization (WHO) [4], about 1.5 million people died of TB worldwide in 2018. This shows that even though the disease is preventable and curable, it remains a major issue that needs to be taken care of. Nowadays, the most common and widely spread approach to detect TB is a technique called sputum smear microscopy which consists of looking for the bacteria in sputum samples using a microscope [3]. More developed countries employ culture methods [3]. Nonetheless, such diagnosis approaches are expensive and barely used in poor regions of the globe such as India, Bangladesh, and Africa where the traditional CXR interpretation by radiologists is still common practice.

Reading CXRs to diagnose TB is an economical approach which unfortunately has major inconveniences, hence its abandonment in developed parts of the world. For example, doing it manually is hard, time-consuming, and susceptible to human error depending on the interpreter's proficiency and aptitude. Finding an automatic affordable and reliable software solution would, therefore, be very handy to save more lives. Not only it would help speed up the whole process allowing patients to receive treatments on time, but it would also help to reduce the number of misinterpretations.

In this thesis, I propose an automatic method for detecting lung anomalies in CXRs using only image processing and statistical machine learning. My approach which was

elaborated after a careful analysis of recent similar works reported in the literature, is a very efficient one that consists in first extracting the region of interest (ROI) from incoming CXRs, second performing a simple feature extraction, and finally classifying the CXRs using the extracted features with a traditional machine learning model.

In the remaining sections of this thesis, I provide in chapter II a summary of similar attempts that have been reported in recent years. I layout a complete description of both my approach in chapter III and all the experiments that I have conducted throughout my study in chapter IV. In chapter V, I conclude by briefly going over the key points discussed in this report, highlighting the drawbacks of the existing solutions and my improvements along the way.

CHAPTER II

LITERATURE REVIEW

In this chapter, I discuss previously reported studies on CXR analysis. The chapter is broken into two sections. In the first one, I introduce the procedures that are most frequently applied to raw CXR scans before they are being analyzed. Section 2.2 provides descriptions of complete CXR analysis systems reported in the literature.

Pre-Processing Steps Frequently Applied to CXRs

Pre-processing medical images usually means extracting the region of interest (ROI) and applying additional procedures to denoise it in order to reduce the number of false positives during the classification phase. It is a very important step as it helps to improve the quality of the images and to reduce the complexity of the problem. In the case of CXRs, pre-processing often includes contrast enhancement, bones suppression, and lung boundary detection or lung segmentation [5].

Applying contrast enhancement can help increase the contrast at the lung boundaries and on the tissue surface [5]. The most commonly used technique is the histogram equalization which consists of spreading out the most frequent intensity levels to make low contrast regions more visible [6]. Other contrast enhancement techniques include energy normalization [7], and transformations using wavelets and piece-wise linear models [8, 9].

Bone suppression is an important step that can help during the lung segmentation and the features extraction phases. The edges produced by the rib cage and the clavicle bones create local minima similar to those created by lung boundaries and sometimes confuse lung boundary detection. On the other hand, ribs and clavicle bones may cover

some objects of interest such as nodules and linear opacities making the feature extraction phase meaningless. Different approaches for detecting and removing bones are discussed in [10, 11, 12, 13]. However, in [14] the authors conducted a study to observe whether or not bone suppression helps in the classification stage of a computer-aided diagnosis system (CAD). Therefore, they applied all the steps of a regular CAD to two separate data sets (one with regular CXR images and the other with bones suppressed CXR images). Those steps include automatic lung segmentation, features extraction, and classification. In the features extraction phase, the authors relied on moments of intensity distributions of Gaussian-derivative filtered images at each pixel and its relative location inside the lung fields. They used a K-nearest neighbor (KNN) classifier in the classification phase and were able to show that bones suppression only slightly increases the performance of a CAD.

Lung segmentation is the process of narrowing down the initial image to the lung regions which is the ROI. It is a necessary step to make sure that only information coming from inside the ROI is being processed in order to make predictions. The simplest approaches to detect lung boundaries are called rule-based and involve operations such as morphology detection and thresholding [15, 16]. Nonetheless, they have been shown to be the least efficient ones, hence their simplicity. Due to that, this method when used is often combined with other more powerful ones [5] such as pixel classification [17, 18, 19, 20], deformable models [21, 22, 23, 24], and hybrid approaches [25]. Pixel classification based segmentation consists of detecting the ROI by assigning labels to every pixel in the image using a machine learning model. On the other hand, deformable models segmentation comprises techniques that combine physics, geometry, and approximation theory to detect the ROI. They identify objects in an image by utilizing restrictions derived from the image in combination with knowledge about the position, size, and

shape of the objects. Hybrid approaches are just a combination of two or more of the other techniques.

To summarize, all three pre-processing techniques discussed in this section are important to improve the quality of the incoming information. First, bone suppression and contrast enhancement may reveal or render more visible objects of interest that are hidden or mingled with other parts of the image. Second, lung segmentation removes unwanted parts of the image leaving only information that is necessary. However, these procedures are complex and therefore, require a lot of time and effort to be implemented. On top of that, they all are computationally demanding. Including them all would considerably increase the complexity and processing time of the whole system. Nevertheless, some authors [3, 26] were able to obtain almost perfect results while omitting some of them. This shows that even though all three procedures are to be considered, they might not all be necessary as long as we can manage to provide meaningful features for the images and a good classifier. Still, I strongly believe that lung segmentation is necessary because I want to make sure that only data coming from inside the ROI is used to discriminate against the CXR scans.

Prior Attempts to Develop a Complete CXR Analysis System

In this section, I provide summaries of some automatic CXR analysis systems that have been proposed in recent years. The section is divided into two sub-sections respectively discussing attempts that make use of simple classifiers and attempts that make use of more complex ones (CNNs).

Approaches that Involve Simple Machine Learning Models

This sub-section goes over some of the attempts to automatically interpret CXR images using simple classifiers such as Multi-layer Perceptron (MLP), Random Forest (RF), and Support Vector Machines (SVM).

Melena et al. [27] proposed a CAD system that combines many subsystems each providing one or more sub-score(s) indicating the presence or absence of anomalies related to TB. All the sub-scores are then aggregated to a single one which is used to make a final decision on whether or not the patient has TB. The system goes through the typical steps of a CAD including lung segmentation using pixel classification, features extraction (Local Binary Pattern (LBP), Histogram of Oriented Gradient (HOG), and Tamura texture features) and classification using a support vector machine (SVM) classifier. Even though the idea of combining subsystems is good and can produce better results, the whole process can be time-consuming depending on how many subsystems are implemented, especially since they basically all have to go through all the different stages.

An approach using only lung segmentation, texture and shape features extraction, and classification using a binary classifier has been proposed by Jaeger et al. [28]. The authors used a graph cut method to segment the images, then computed a set of six feature descriptors including LBP, HOG, curvature descriptor histogram (CH), intensity histogram (IH), gradient magnitude histogram (GM) and shape descriptor histogram (SH). They computed 32 bins for each of theses feature descriptors and the bins' counts were appended to a final feature vector which ended with a total of 192 entries (32 x 6). An SVM classifier was used in the classification stage and the model was trained and tested on both the Montgomery and Shenzhen datasets [29] achieving 78.3% and 84.10% classification accuracy respectively. This approach, unfortunately, has two main issues

which are its rather low classification accuracy on both datasets and its complexity. For a problem of high importance such as diagnosing a deadly disease, one could expect such a critical system to perform with a very low error rate. So, correctly classifying only 84% of patients is just not enough. On top of that, feature extraction is an expensive operation, and applying it more than once is probably not necessary.

A similar approach to the one described in [28] was proposed by Vajda et al. [3]. In this study, the authors used an Atlas-based approach for lung segmentation, then performed a feature extraction similar to the one described in the previous paragraph, followed by a feature selection. The feature selection procedure consists of running an exhaustive search to find all the possible combinations of different features, applying a train/test procedure to each of them to see which one would produce the best accuracy or area under the curve (AUC). The winning combination is therefore retained and used to represent the corresponding CXR in the classification stage of the CAD. The classifier used to discriminate the scans is a multi-layer perceptron (MLP) classifier. The system was trained and tested on the Montgomery and Shenzhen collections separately and produced an impressive classification accuracy of 95.57% for the latter collection though it was only able to properly classify 78.3% of the images in the former set. This study shows that complex machine learning models are not necessary as long as we can provide good descriptors for the images. Even so, the system remains complex given the costly features extraction and features selection procedures.

Xu et al. [30] proposed a complex process to detect TB cavities, involving many steps during which textural and geometrical features are extracted from the images. They start by extracting possible cavities from CXRs with a feature classification model (SVM) using LBP, HOG, and Gaussian model-based template matching (GTM). Next, they enhance the quality of the cavities candidates with image enhancement techniques using

the eigenvalues of the Hessian matrix (HIE) and an active contours snake-based technique (ACS). Finally, they classify the resulting enhanced images with an SVM classifier using features such as circularity measure, gradient inverse coefficient of variation (GICOV), and Kullback-Leibler divergence (KLD). Using this approach on a small cavity data set of 35 CXR images obtained from the University of Alberta Hospital, the authors reported a classification accuracy of 82.8%. However, this system is just not good enough as it goes through a long and complicated process involving the extraction of multiple feature descriptors and application of many image processing techniques to only produce 83% classification accuracy in the end.

Approaches that Involve Deep Learning Models

In this sub-section I go over some of the attempts to automatically interpret CXR images using CNN networks.

Rajpurkar et al. [31] describe the implementation and performance of their CAD system called ChexNet. Chexnet is a convolutional neural network with 121 layers implemented to detect anomalies linked to pneumonia in CXRs. The model was trained on a large publicly available data set, the ChestX-ray14 which contains $112,120$ frontal-view images of $30,805$ patients. The model was then tested on 420 frontal images collected separately by the authors. The report shows that ChexNet was able to outperform the average performance of radiologists as it achieved a 0.435 F1 score compared to 0.387 on average for radiologists. However, the issue with this approach is that in practice, CNN classifiers require a large number of training samples in order to perform properly. The datasets that I am working with --the Montgomery and Shenzhen data sets--, are limited in the number of samples as they contain only 138 and 662 images respectively. More information about these two collections is given in Section 4.2.

Hwang et al. [26] proposed an approach using deep learning to detect active pulmonary tuberculosis on CXRs. The system is developed using a large dataset of 65,548 CXRs --57,481 normal and 8,067 abnormal-- obtained from the imaging database of Seoul National University Hospital (SNUH). They randomly divided the set into three sub-sets for training, tuning, and internal validation. The deep convolutional neural network they used comprised of 27 layers with 12 residual connections. Since only some of the training samples were annotated, they trained the model via a semi-supervised localization approach. They randomly re-scaled the images and applied data augmentation before inserting them into the network to respectively cover various lesion sizes and strengthen the model for inputs obtained from different machines and in different environments. After internally evaluating the network, they tested it on six different external data sets, including the Montgomery and Shenzhen collections. For the last two, the authors respectively reported achievement of 0.991 and 0.977 area under the receiver operating characteristic curve (AUC). Though effective, this approach uses a complex classifier which not only requires advanced and expensive computer architectures to operate effectively, but also thousands of images to train.

Pasa et al. [32] propose another approach using deep learning to automatically screen CXRs for TB. The network they used comprises five convolutional blocks of two convolution layers each. Each block is followed by a max pooling layer. The model also includes a global average pooling layer right after the convolutional blocks, followed by a fully connected softmax layer with two outputs. They trained and tested the model on the Montgomery and Shenzhen data collections and reported accuracies of 79.0% and 84.4% respectively. This study confirms that CNNs cannot produce good results unless they are provided with a large number of training samples and very complex structure (see more convolutional blocks).

To conclude this section, my analysis of previous works that have been done in the domain revealed that most if not all existing systems follow either of two specific patterns. In the first one, the images are pre-processed, then some high-level features are extracted from them. These features are fed to a simple machine learning model (MLP, RF, SVM etc.), and finally we obtain classification results. In the second pattern, the images are pre-processed, then fed to a deep learning model such as a CNN, and finally, we obtain classification results. Theses patterns are more clearly illustrated in Figure 1.

Even so, for my system, I have decided to go with the first model. My choice for this approach is due to i) the fact that authors such as Vajda et al. [3] used it and were able to develop a system that produces state-of-the-art results and ii) because of my inability to obtain a large CXR collection (that also includes segmentation masks for the lung regions) which is crucial to use the second approach.

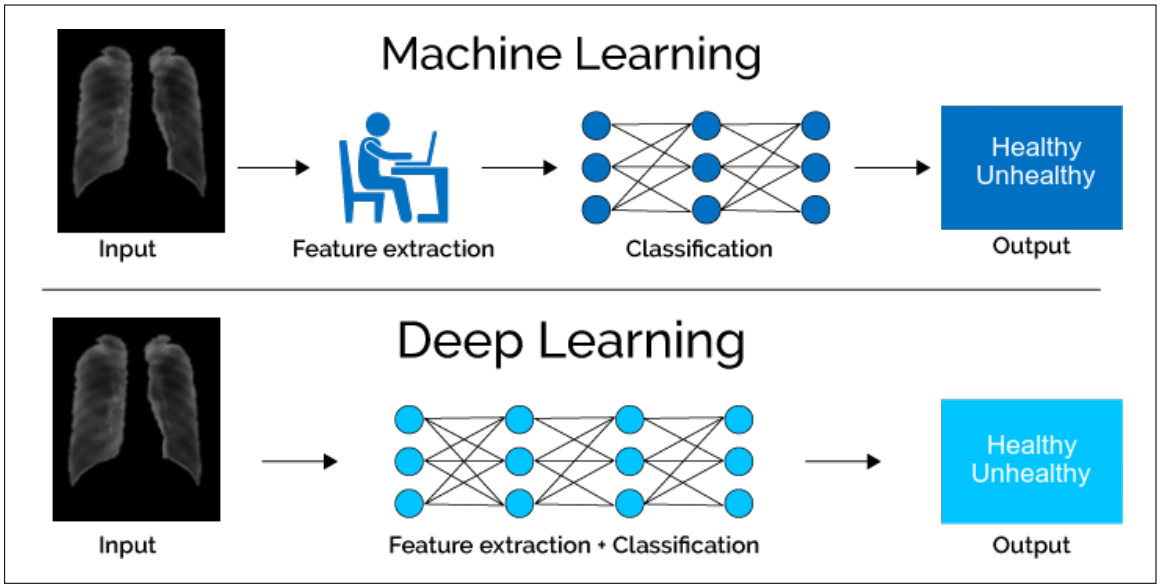


FIGURE 1: Two patterns often used in CAD systems. Image downloaded from [1] and modified by myself to adjust to the content of my study.

CHAPTER III

AN APPROACH THAT COMBINES A SINGLE FEATURE WITH POSITIONAL INFORMATION

This chapter presents my approach to detect anomalies in CXRs. It is split into three sections that respectively describe the image pre-processing, feature extraction, and classification stages of my system.

Image Pre-Processing

To make sure that information coming from outside the lungs does not impact on the results that I obtain, I limited my analysis only to the lung region. I extracted it from the original images using the segmentation masks that are provided with both the Montgomery and Shenzhen datasets. Each segmentation mask has the same resolution as its corresponding plain image and has all its pixels' values set to zero except for those that belong to the ROI which values are set to 255 (see Figure 2). Having this information, extracting the ROI was simple. Since the images are in fact 2D arrays (see grayscale images), I iterated through all its pixels and checked their values. For each pixel with a value equal to zero, knowing its exact position in the segmentation mask, I located the corresponding pixel in the original image and set its value to zero. All the other pixels remained unchanged. Some results of this procedure can be seen in Figure 3-a and Figure 3-c.

After extracting the ROI, the resulting images still had a lot of irrelevant information (see the black region around the lungs in Figure 3-a and Figure 3-c). Such information may increase similarities between the input images and impact on the classification results. So, I got rid of most of it as I thought it was necessary for my

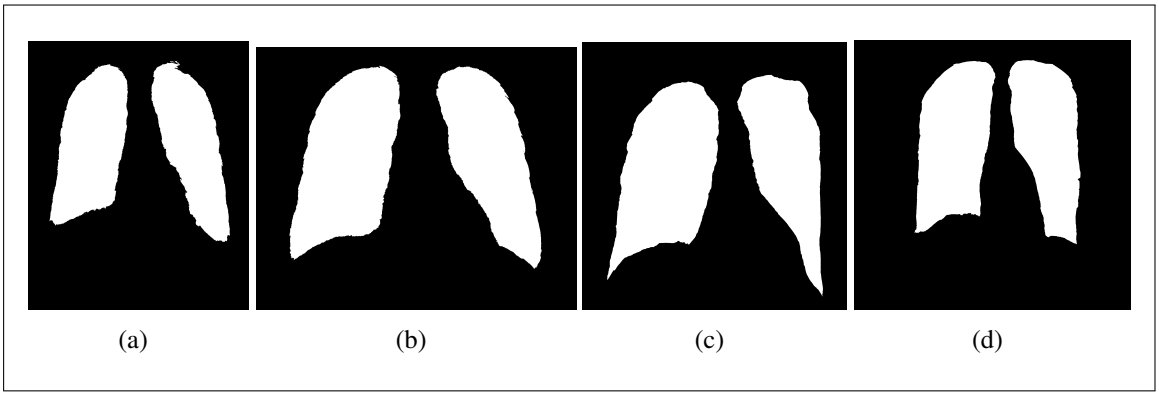


FIGURE 2: Examples of masks taken from the Montgomery (a, b) and Shenzhen (c, d) datasets.

system to operate as intended (see Figure 3-b and Figure 3-d). Doing so allowed me to have a more compact image with a continuous ROI. The resulting images are also of a smaller size than the original ones which means a reduced processing time for subsequent operations.

Feature Extraction

This phase of the system consists in reducing the dimensionality of the input data (i.e., pre-processed images). In fact, though the original images have been reduced in size during the pre-processing phase, the amount of information they contain is still extremely important. Therefore, it must be reduced in order to improve the performance (see processing time) of the whole system. To serve this purpose, after pre-processing the original images, I extracted Laplacian of Gaussian features (LoG) [33] from each resulting image. LoG is an operation that consists of first applying a Gaussian filter to an image to get rid of the noise, and second applying a Laplacian filter to detect edges or areas of rapid change [34]. Such high-level features have the ability to describe an image using only a fraction of the information contained in the original image. Nonetheless,

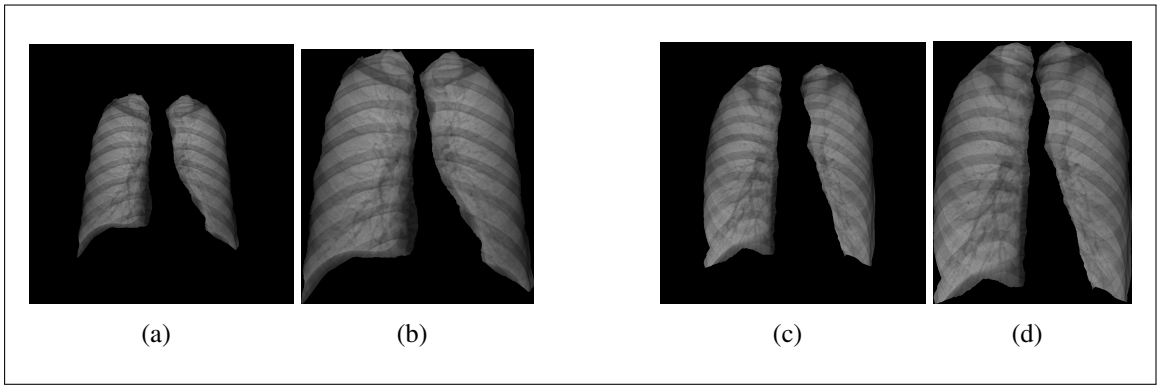


FIGURE 3: Examples of input images before (a, c) and after (b, d) reducing their sizes.

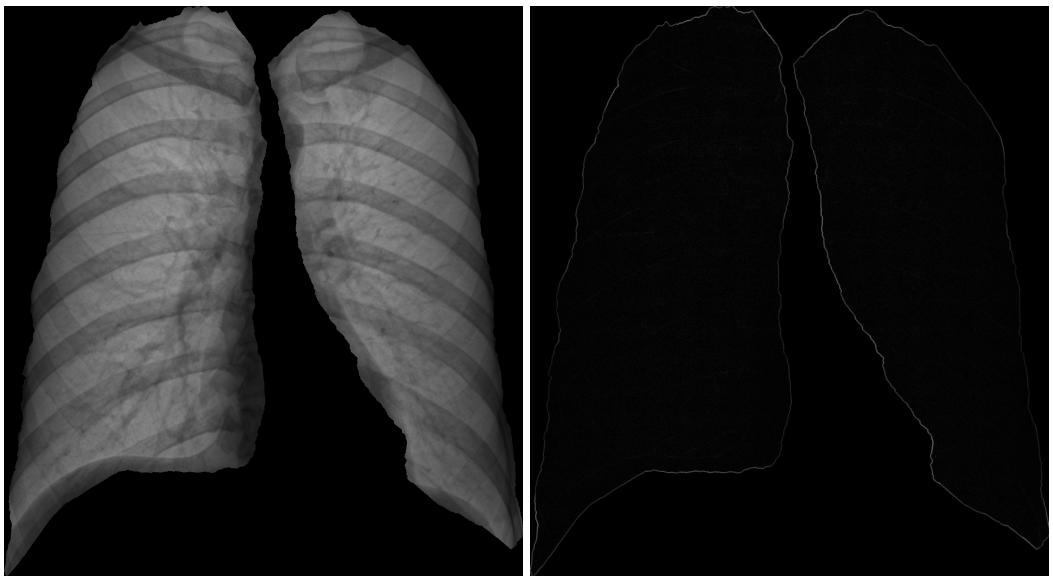
I have also considered LBP features for my system but LoG was retained because I was able to obtain better results with them. LBP [35] is a simple but very efficient texture operator that labels each pixel in an image by thresholding its neighborhood and considering the result as a binary number.

LBP and LoG were considered instead of other well know and widely used ones such as Pyramid Histogram of Oriented Gradient (PHOG), HOG, IH, GM, etc. [3, 28, 36, 37] for one specific reason. In contrast to LBP and LoG, with these other features positional information is lost during the extraction process because the results they provide are computed by combining data coming from diverse parts of the image. In other words, the information contained in these descriptors is not traceable to specific regions of the original image. However, being able to trace it back to its provenance, I believe, is very important given the nature of the task that I am trying to solve. In fact, anomalies such as those related to TB, when existing, usually manifest themselves in specific parts of the lungs. Providing classifiers with not only feature descriptors, but also insights about the regions of the image where they were extracted from surely will make it easier for them to distinguish normal CXRs from abnormal ones. Being able to pull out such enriched information constitutes the key of my approach.

For each input image, the process to extract LoG features combined with positional information consists of applying to it first a Gaussian blur and second a Laplacian operator. The result of these two procedures is an image of the same resolution as the input one (i.e., a 2D array see Figure 4). To extract simple LoG features, one would just compute the bins or intensity histogram using the values of this resulting 2D array. However, in order to incorporate positional information, I first divided the LoG image into multiple sub-images or regions with square or rectangle like shapes (i.e., n cuts row-wise and m cuts column-wise). The division was performed so that all the regions have about the same width and height. The next step consisted of computing the bins for each region of the LoG image and append them to a final feature vector. I processed each region one at the time and directly appended the results to the feature vector before moving to the next region. The order in which the regions were processed remained consistent across all the input images because this order is what makes the features traceable to their original emplacement. Specific details about the implementation of this phase such as the number of bins or the number of regions are provided in Section 4.8.

Classification

This phase consists of classifying the CXRs using a RF classifier. Even though I have experimented with other machine learning models such as KNN, MLP, SVM, and even CNN, RF appeared to be the most consistent across all the experiments that I have conducted. These four classifiers were selected for my experiments due to their capacity to successfully solve similar tasks as reported in [3, 28, 36, 26]. Despite having experimented with a CNN, it has never been my intention to incorporate such a model in my system. The reasons include the size of the CXR collections that I was working with (see their limited number of training samples) and also my confidence that a simple



(a) (b)

FIGURE 4: Example of an input image and its corresponding LoG image.

classifier, if provided with good feature descriptors for the images, would be enough to accurately sort them out.

KNN [38] is a supervised machine learning algorithm commonly used to solve classification and regression problems. The latter are those which have real numbers as output while the former are those which output discrete numbers. Some of the advantages of the KNN algorithm include its simplicity and easiness to implement. The base point of the algorithm is its assumption that similar objects are close to each other. Therefore, the process consists in computing the distances between the elements of the input set and relying on how far or close these elements are from each other to make predictions.

MLP [39] is a class of feedforward artificial neural network which comprises an input layer, an output layer, and multiple other layers in between the first two often called hidden layers. The term feedforward means that there are no cycles within the network, i.e., each layer feeds its output forward to a lower-level layer. However, the training is

performed using backpropagation to update the weights in the model. The whole concept was modeled around the functioning of the biological brain and neurons to solve complex tasks frequently seen in the field of machine learning. It is a model that can be used to solve regression and classification problems.

SVM [40] is a supervised machine learning model that uses a kernel method to transform the input data in order to find an optimal boundary or hyperplane between the possible outputs. It is an old model and though it isn't anymore as popular as it used to be, it is still considered as one of the best classifiers to solve classification and regression problems.

RF [41] is a very popular and widely used machine learning algorithm. Its flexibility, simplicity, diversity, and efficiency in solving classification and regression tasks make it one of the most frequently used algorithms. The algorithm consists in building a forest of decision trees and use their votes to make a final prediction. Even without tuning its hyper-parameters, it almost always produces good results which show how powerful it is compared to other machine learning models. On top of that, it is also well known for its capacity at avoiding over-fitting meaning that it performs well not only on the dataset that is used to train the model but also on any other set that it might be tested on.

All these four machine learning models (KNN, MLP, SVM, and RF) have the ability to solve classification and regression problems, they are widely used, and they have been shown to be very efficient.

CHAPTER IV

EXPERIMENTS

This chapter includes a description of the resources that were used during my study, the CXR collections that I considered, the experiments that I have conducted, and a discussion of the results that I obtained.

System

All the experiments conducted during my study were performed on a personal computer running Ubuntu 18.04 LTS. The computer has a 1 TB hard drive, 24 GB of memory, and an Intel processor with 2.80 GHz and 8 cores.

All my code was written in Python 3 using multiple libraries or packages. Numpy [42] was used to manage data in memory and Pickle [43] to manage data on the disc. OpenCV [44] was considered for image pre-processing, LoG and HOG features extraction and Scikit-image [45] for LBP features extraction. I used Scikit-learn [46] for the KNN, MLP, SVM, RF, and K-means models, PyMVPA [47] for the Self Organizing Maps (SOM) model, and Keras [48] (with Tensor-flow back-end) for the CNN classifier.

Data

For the experiments two different, publicly available CXR data collections[1] were considered. The images in these studies were de-identified by the data providers, and are exempted from IRB review at their institution.

The Montgomery dataset, --a representative subset of a larger image repository-- was collected over many years within the tuberculosis control program of the

[1]Download:`https://lhncbc.nlm.nih.gov/publication/pub9931`

Department of Health and Human Services of Montgomery County. The set contains 138 posteroanterior CXRs, among which 80 CXRs are normal ones, while the remaining 58 CXRs are abnormal cases, presenting some sort of abnormality indicating TB. Figure 5 shows some CXR examples taken from this collection with their corresponding segmentations.

The Shenzhen dataset is from Shenzhen No. 3 Hospital (Shenzhen, China), one of the largest hospitals in China for infectious diseases. The CXR images belong to outpatient clinics. The collection contains 342 normal, and 334 abnormal cases. Figure 6 shows some CXR examples taken from this set with their corresponding segmentations. For more details about the data, please refer to [29].

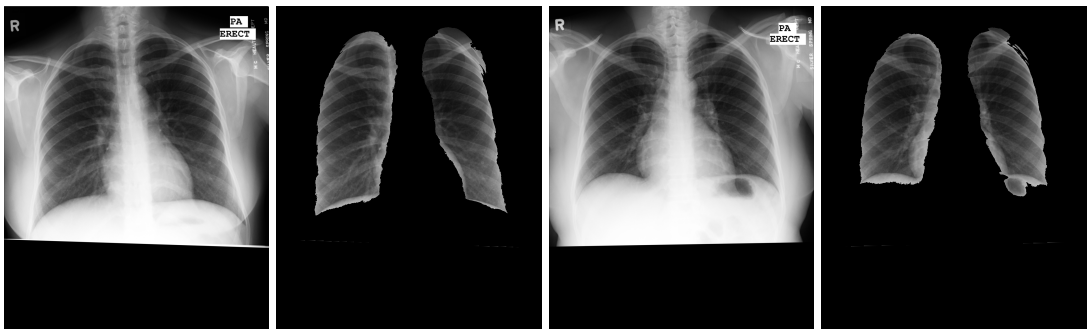
Lung Segmentation

My primary plan included a lung segmentation in the pre-processing stage of my system. For this, I considered two different approaches i) a hybrid one that combines pixels classification and rule-based techniques and ii) a simple rule-based one.

A Hybrid Approach

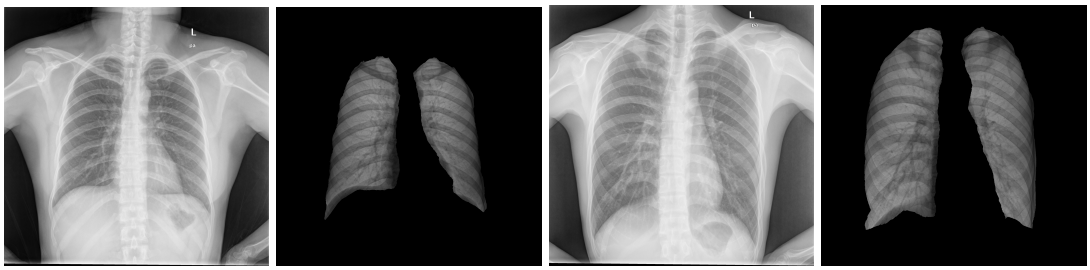
The first approach that I considered consisted of combining both pixel classification and rule-based techniques.

In the pixel classification step, I first created a tuple for each pixel in the image. The tuple contained the pixel value, its row index, and its column index. The row and column indexes were normalized --divided by the width and height of the image respectively-- before they were inserted into the tuples. The role of these two indexes is to help the classifier discriminate the pixels based not only on their values but also on their respective emplacements in the image. I classified the tuples using two clustering algorithms namely



(a) (b) (c) (d)

FIGURE 5: Examples of CXRs images and their corresponding segmentation from the Montgomery data set.

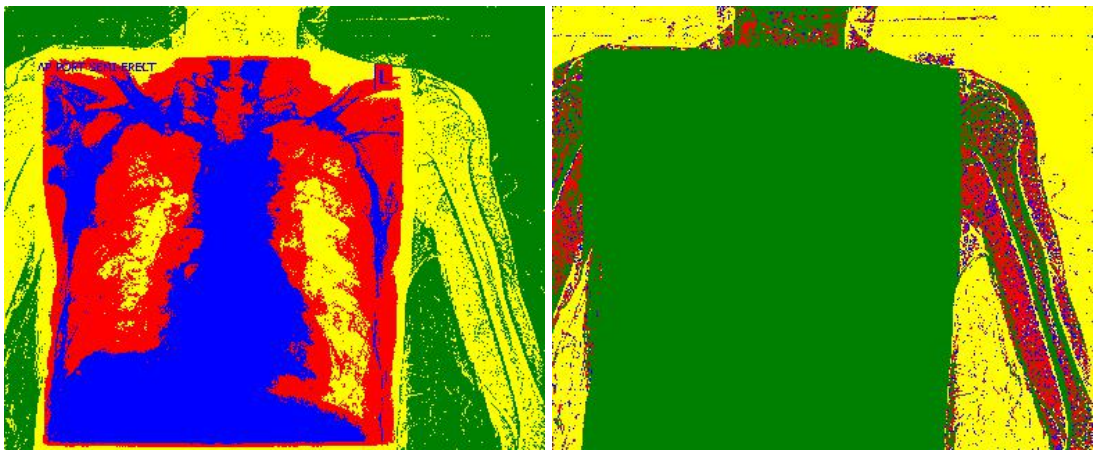


(a) (b) (c) (d)

FIGURE 6: Examples of CXRs images and their corresponding segmentation from the Shenzhen data set.

K-means and Self Organizing Maps. However, the former appeared to be the most efficient (see Figure 7). Having the clusters, the following step consisted in assigning different values (comprised between 0 and 255) to each of them and then replace the values in the original image with those that were assigned to the cluster they belong to. I saved the resulting image and proceeded to the rule-based step.

The rule-based process included sweeping the whole image looking for connected components using the function ”findContours” of the publicly available library OpenCV [44]. Next, out of the extracted connected components, I selected the two that comply



(a) (b)

FIGURE 7: Example of pixel classification (four clusters) results using K-means (a) and SOM (b).

with some predefined rules (see Figure 8). These rules basically set the requirements about the position, height, and width of the ROI relative to the full image. I then used the pixels' information of the two selected connected component to extract the ROI from the original image.

The major problem that I have faced with this approach was the processing time. Even though I had optimized my program to segment the images in parallel using all the eight cores available on my system system, it was still not able to segment an image (with a resolution of 1024 x 1024) in less than a minute on average. The high complexity of my program was mainly due to the pixel classification part since for some images, it had to run the clustering algorithm multiple time using a different number of clusters each time until it finds the one that would successfully separate the lungs from the rest of the image (see Figure 9).

(a) (b) (c)

FIGURE 8: A summary of the lung segmentation process. (a) is the original image, (b) shows all the connected components that were found, and (c) shows which two connected components were selected.

A Rule-Based Approach

In order to reduce the processing time of my segmentation algorithm, I suppressed the pixel classification phase and directly applied the rule-based step on the original images. The process remained the same as described in Sub-section 4.3. However, as can be observed from Figure 10, this approach ended up working very well for some images but not so well for others mainly due to the location of the ROI relatively to the edges of the image. Note that in (Figure 10-c) the left lung is too close to the right edge of the image which does not comply with one of the rules set in my algorithm.

I tested my implementation on the Montgomery and Shenzhen CXR collections and achieved 0.65 and 0.72 Jaccard similarity coefficient respectively. Nevertheless, these results are very low when compared to 0.94 reported in [3] for the Montgomery collection.

The Jaccard index or Jaccard similarity coefficient is a statistic used to measure the similarity between finite sample sets. It is computed using the following formula:

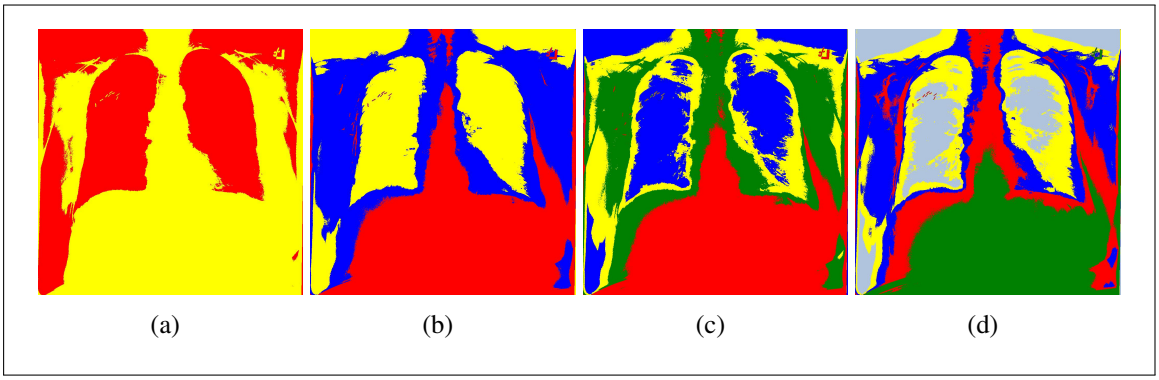


FIGURE 9: Results of pixel classification using different number of clusters (2, 3, 4, 5 clusters respectively).

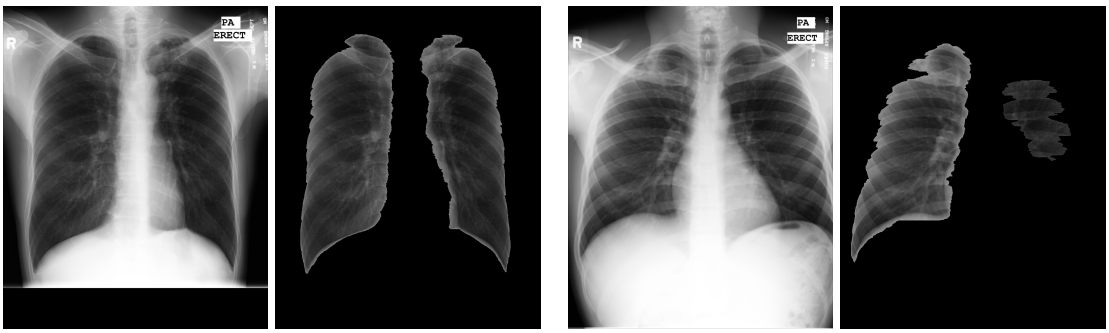
$\Omega = \frac{|TP|}{|FP|+|TP|+|FN|}$ where $TP = true\ positive$, $FP = false\ positive$, and $FN = false\ negative$.

As mentioned in Section 2.1, developing a lung segmentation program is a hard and time-consuming task which itself is another research topic. Even though my initial intention was to produce a complete diagnosis system by providing implementations for all its different phases, my main goal has always been to successfully be able to detect TB anomalies in CXRs. Therefore, after trying without success to come up with a fast and accurate lung segmentation algorithm, I have decided to move on. Instead, I used the masks provided with the Montgomery and Shenzhen datasets to extract the ROI, then I focused my attention on reaching my main goal.

Evaluation Metrics

To evaluate the performance of the proposed system, several metrics were considered but accuracy (ACC) and area under the receiver operational characteristic (ROC) curve (AUC) were retained [49].

The ACC was chosen because of its simplicity and also because it is the only one that anyone can understand easily even people who don't have prior machine learning



(a) (b) (c) (d)

FIGURE 10: (a) is an example of an image that was segmented properly using my algorithm and (b) is the result of the segmentation. (c) is an example of an image that was not segmented properly and (d) is the result of the segmentation.

knowledge. It is computed by dividing the number of correct predictions by the total number of predictions all multiplied by 100. Though it provides an easy understanding of how well a model is doing, the latter cannot be judged solely based on it. Indeed, in some cases that involve the classification of an unbalanced dataset, a classifier may get a good accuracy just by predicting that all the test samples belong to the category with more items. To show that this is not my case, I considered using a second metric.

AUC was chosen as the second metric due to the requirement of the problem in question. In fact, in my case where the issue is to decide whether or not a patient is sick, there is a certain need to control the true positive rate as none should be missed if his/her CXR shows signs of an anomaly that is been searched for [50]. AUC is a value between zero and one that is obtained by computing the area under the ROC curve. The latter is a plot that describes the performance of a binary classifier by showing its true positive rate (TPR) against its true negative rate (TNR) for various threshold values. Therefore, AUC can also be understood as the probability that a classifier ranks a random positive sample higher than a random negative sample.

K-Folds Cross-Validation

To assure the consistency of my results, a ten folds cross-validation was performed for all the experiments described below and the reported results are the averages of what was obtained. For each experiment, after randomly shuffling the dataset, I divided it into ten subsets of similar sizes each containing about the same amount of samples from both categories (healthy and non-healthy). Next, ten train/test procedures were run for each experiment, each time using a different subset for testing and the four remaining subsets for training. The ACC and AUC were recorded after each procedure and their averages were computed and reported at the end of the experiments.

CNN Classification

Given the incontestable success of convolutional neural networks for different tasks such as speech recognition, image segmentation, natural language processing, and also chest x-ray analysis, I have also considered such an approach. I implemented a CNN model which comprised two convolutional layers, two fully connected layers of 100 and one neuron(s) respectively, a ReLU function applied after each layer except for the last one where a softmax activation function is applied. The convolution layers were using 64 filters, a kernel size of three, and the regularization in use was ”dropout” with a value of 0.5. Nevertheless, I experimented with different values for the parameters and different numbers of hidden layers and neurons, but the one described in this section produced the best results.

After extracting the ROI from the original images, I created the ten folds for cross-validation as described in Section 4.5. For each of the ten runs, I performed a data augmentation by flipping around and resizing the images. In the end, I computed the average of the results obtained after each run and obtained 57% and 50% classification

accuracies for the Montgomery and Shenzhen collections respectively. Such low scores were to be expected given the insufficient amount of samples in both datasets since CNNs need more training data to learn efficiently. Certainly, I could have experimented with a more complex model to possibly increase the results. However, I didn't consider doing so because my goal was to develop a fast and efficient system. Adding a complex model to it would just have increased its processing time. After all, similar attempts to increase the complexity of the model have been made by others and still the results they obtained were not good enough [32], or the models still required large training sets in order to perform well [26].

Due to my incapacity to find larger CXR collections which also provide segmentation for the lung regions, I decided to step back from CNNs. My attention was since then focused on features extraction and classification using other machine learning algorithms. Eventually, other researchers working on the same data collections (Montgomery and Shenzhen) [3, 28] obtained promising results using this process (i.e., features extraction and classification).

Classifiers

As mentioned in Section 3.3, I experimented with KNN, MLP, SVM, and RF classifiers. In order to find the right topology and parameter set for each of the classifiers, I conducted several trial runs and retained those that produced the best ACC or AUC. Finally, for the KNN I considered using a KDTree neighbor search algorithm with Euclidean distance as the distance metric and three neighbors. My MLP comprised a sequential model, three hidden layers with 750, 400, and 400 neurons respectively, one output layer with one neuron, an Adam optimizer, and a rectified linear unit (ReLU) activation function in all the layers except for the last one where a sigmoid function

was considered. For the SVM, a linear kernel was retained after experimenting with few others including polynomial, Gaussian, and Gaussian Radian Basis Function (RBF) kernels. My RF classifier used 900 estimators and the Gini impurity criterion to measure the quality of the splits.

Results

In this section, I provide details and results of all the attempts that I have made to develop a system that does not involve a CNN classifier. I also provide a direct comparison of the best system that resulted from my experimentation with the best ones reported in the literature.

To show that saving positional information during the feature extraction phase of my best system is really what makes the difference, I conducted two types of experiments. In *Experiment 1* I developed baseline systems involving classical features, while in *Experiment 2* I developed a system that combines LoG features with positional information. For the sake of equity, the classifiers and parameters used for the feature extraction remained the same across both experiments.

Experiment I

The main point of this experiment was to observe how well a system would perform without the implication of any special procedure. I also wanted to confirm that my classifiers were properly configured and discover which type of feature would best describe images such as CXRs. Therefore, four types of feature descriptors commonly used in this field were extracted and used separately: HOG, PHOG, LBP, and LoG.

HOG features are widely used for object detection and the process to extract them consists of splitting an image into small squared cells and counting the occurrences of

gradient orientation in each cell. I extracted 600 of such features from each image using the functions *HOGDescriptor* and *HOGDescriptor.compute* of the OpenCV Library [44].

PHOG features are spatial shape descriptors which include HOGs over each sub-region of an image at each pyramid resolution level. Using the implementation provided in [51], I extracted 1020 features from each CXR scan.

The LBP descriptors were computed using the *feature.local_binary_pattern* function of the scikit-image image processing package [52]. The parameters used for the function include radius = 300, number of points = 600 and method = "uniform". From the resulting image, I computed 600 bins.

The LoG features were computed using two functions from the OpenCV library [44]. First, I applied the GaussianBlur function with *k_size* (kernel size) = 3 to the input image, and second, I applied the Laplacian function with *ddepth* (depth of the destination image) = CV_64F. From the resulting image, I computed 500 bins.

Though the specific parameters and numbers of extracted features or bins used in this experiment may seem arbitrary, I can assure that I have run several trials using different values before establishing these exact configurations.

The results reported in Table 1 for the Montgomery data and in Table 2 for the Shenzen data are completely aligned with those obtained by other researchers using the same type of features and same or similar classifiers [3, 28, 36]. Though they are not the best, the fact that they are close to what other researchers obtained shows that my classifiers have good configurations. After all, in contrast to them, I used single features and less processing steps to obtain these results.

While for the Montgomery data each classifier performs better for a different feature, in the case of the Shenzhen collection I can observe that RF is providing the best accuracy for each feature over the rest of the classifiers. This shows to some extent the

TABLE 1: Classification accuracies for the Montgomery dataset using different classifiers and feature descriptors.

Classifier	HOG (%)	PHOG (%)	LBP (%)	LoG (%)
KNN	**69.04**	73.81	42.43	59.52
MLP	69.04	73.81	45.27	55.36
SVM	69.04	**76.19**	48.89	64.29
RF	67.19	64.29	**73.81**	**71.43**

Note: Boldfaced numbers indicate the highest values in their corresponding columns.

TABLE 2: Classification accuracies for the Shenzhen dataset using different classifiers and feature descriptors.

Classifier	HOG (%)	PHOG (%)	LBP (%)	LoG (%)
KNN	64.29	79.91	72.24	74.89
MLP	65.47	78.99	77.84	79.91
SVM	76.19	77.84	59.17	**84.93**
RF	**77.84**	**80.95**	**77.84**	84.01

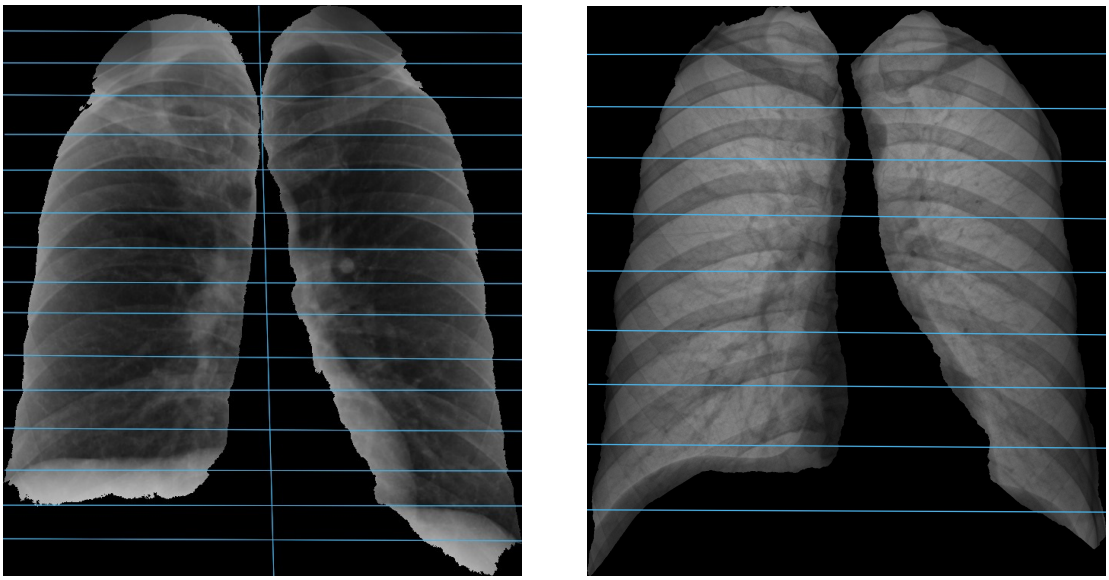
Note: Boldfaced numbers indicate the highest values in their corresponding columns.

strength of RF. On the other hand, Table 1 and Table 2 also show that the system tends to perform better when using LoG features rather than LBP. This tendency helped us decide on what type of feature to use for my next experiment.

Experiment II

As described in Section 3.2, the main point of this experiment was to add a procedure to the system that would allow us to retain some positional information during the feature extraction phase. I wanted to see if such information could indeed help discriminate against CXRs.

In this test, I revised my feature extraction phase so that features are drawn out from regions of the image rather than the whole image directly. In fact, as depicted in Figure 11, I divided the pre-processed input images into multiple regions after applying the LoG procedure to each of them. For each region, I built a local histogram based on



(a) (b)

FIGURE 11: A visual representation of how the images were splitted into regions. (a) shows an image from the Montgomery collection and (b) shows an image from the Shenzhen collection.

pixel densities. It is worth mentioning that unlike *Experiment I*, all pixel values were not considered to create the bins or histogram in this experiment and the reason is that I have noticed that I could slightly improve the results by doing so. However, even by considering all the values, the results were still much better than those reported in Table 1 and Table 2.

The range of pixel values considered as well as the number of bins and the number of regions used were different for both datasets and are recorded in Table 3. All these numbers in Table 3 were retained after experimenting with many others as they produced the best ACCs or AUCs. For the number of rows and number of columns I tried all integers from 1 up to 25. For the ranges of values to be considered when computing the intensity histograms (i.e., bins), I tried all integers from 0 up to 10 and 40 up to 60 for

TABLE 3: Details about the features extraction phase.

Dataset	# rows	# columns	# bins	values range
Montgomery	16	2	44	1 - 45
Shenzhen	10	1	40	5 - 45

the left and right bounds respectively. The number of bins was obtained each time by computing the difference between the left and right bounds of considered pixel values.

As to be observed in the results (see Table 4), the RF is providing the best ACC of 83.33% with the corresponding AUC of 0.815 for the Montgomery data collection. For the Shenzhen set, the KNN classifier is reaching an impressive ACC of 97.26% and a very promising AUC of 0.973. When compared to the best results obtained during my first experiment considering LoG features and regardless of the type of classifier used, we notice a net accuracy improvement of 11.9% and 12.33% for the Montgomery and Shenzhen sets respectively. These substantial improvements show how much the additional information saved during the features extraction procedures were helpful for the different screening processes. These numbers also show that this system can reliably be used to identify TB abnormalities in different types and sizes of CXRs.

To put my results in a larger context, I have also provided a direct comparison (see Table 5) with the current state-of-the-art results reported in the literature on the same data collections. As is to be observed, my system outperforms all others mentioned in the comparison if considering ACCs only, while only Vajda et al. [3] and Huang et al. [26] obtained better results than us on both datasets when looking at AUCs. However, it is worth mentioning that their results were optimized. Vajda et al. [3] incorporated a computationally expensive feature selection in their system while Huang et al. [26] used a complex CNN classifier pre-trained with 65,548 images. My system does not have any such optimizing procedure and relies solely on the data that it's been provided with in order to make the predictions. It relies on a single type of feature (see LoG) unlike

TABLE 4: Classification results using LoG features enriched with positional information.

Classifier	Montgomery ACC (%)	Montgomery AUC	Shenzhen ACC (%)	Shenzhen AUC
KNN	69.05	0.681	**97.26**	**0.973**
MLP	71.43	0.707	91.32	0.914
SVM	54.76	0.500	74.89	0.749
RF	**83.33**	**0.815**	96.35	0.964

Note: Boldfaced numbers indicate the highest values in their corresponding columns.

others [3, 28] that rely on six or more though the process to extract every single one of them is costly in time and memory space. My system uses a basic RF classifier unlike others [26, 53, 54, 55, 32] that use more advanced machine learning models (see CNN). Altogether, the results in Table 5 show that extracting good quality information from the images can equally serve to design a reliable system just like other approaches --making use of complicated procedures, complex classifiers, and large datasets--.

TABLE 5: A fair comparison of my results with those reported by other researchers on the same CXR collections (Montgomery and Shenzhen).

Method	Montgom. ACC (%)	Montgom. AUC	Shenzhen ACC (%)	Shenzhen AUC	Classifier Used
Our method	**83.33**	0.815	**96.35**	.964	RF
Vajda et al. [3]	78.3	0.87	95.57	**0.99**	MLP
Jaeger et al. [28]	78.3	0.86	84.10	0.88	SVM
Hwang et al. [26]	–	**0.991**	–	0.977	CNN
Hwang et al. [53]	67.4	0.884	83.7	0.926	CNN
Islam et al. [54]	–	–	88.0	0.91	CNN
Rajaraman et al. [55]	–	–	89.9	0.948	CNN
Pasa et al. [32]	79.0	0.811	84.4	0.90	CNN

Note: Boldfaced numbers indicate the highest values in their corresponding columns.

CHAPTER V

CONCLUSION

Tuberculosis is a dangerous and easily transmissible disease that kills more than one million people around the world each year. Despite the efforts put in place to prevent and cure the disease, the toll remains high partly due to the lack of efficient diagnosis systems especially in the most affected regions (see Africa and Southeastern Asia). In these places, radiologists diagnose patients by visually inspecting their CXRs. Though economical, this approach is difficult, slow, and unreliable compared to other more expensive ones commonly employed in wealthy countries (see sputum smear microscopy and culture methods). Since these flaws are mainly due to the fact that the task is performed manually, researches aiming to successfully replace the human interpreters with software analysis systems have been conducted over the years. These attempts were made in the areas of computer vision, artificial intelligence, and machine learning. Numbers of solutions resulted along the way, some of which are good to the point they can even interpret CXRs better than the average radiologist. However, all of these proposed automatic solutions have their own flaws. Only a few of them can accurately interpret CXRs and they do so at the price of complexity --they need both to be operated on powerful and expensive equipment and a lot of training data--. For the rest, they are just not reliable enough as they can't even reach the average performance of radiologists.

My goal with this study was to develop an automatic analysis system that can efficiently and accurately classify CXRs even if the resource available for it is limited, --a small amount of training data and inexpensive equipment--. After reading about what others have done in the field, I realized that there are two common approaches to develop

such a system. The first consists of using image pre-processing techniques to enhance the quality of the input images and/or extract the ROIs. Following, extracting some high level features descriptive of the input images and feed them to a generic machine learning model (see KNN, MLP, SVM, RF, etc.) for predictions. The second approach consists in pre-processing the input images, then feed them directly to a more complex deep learning model such as a CNN for predictions. Since, the second approach requires a lot of training data which I didn't have, I opted for the first one which is less demanding and produced state-of-the-art results when used by Vajda et al. [3]. Even though the results obtained by Vajda et al. [3] (see Table 5) are impressive given the number of training samples that was used (see Section 4.2), the authors rendered an approach, supposed to be simple, very complex. They repeated the computationally expensive feature extraction step many times and added another expensive step for feature selection to their system (see Section 2.2). I, therefore, tried to develop a system that would produce similar results while using the same approach, but with fewer computations.

Since anomalies such as those related to TB can only be detected in specific parts of the lungs, I believed that preserving some emplacement information across the feature extraction phase would be helpful to correctly classify the CXRs.

My system starts with an image pre-processing phase where I extract the ROIs from input CXR scans and crop out some other unwanted parts of the images (see Section 3.1). Following, I perform the extraction of LoG features while preserving information about their respective positions in the image. This phase, which constitutes the key of my approach, consists of applying first a Gaussian filter and second a Laplacian filter to each CXR. Next, I divide each resulting LoG image into regions and for each region, I compute local histograms or bins based on pixel densities. I append the bins' counts for each region to a feature vector --one vector per image-- as they are being computed.

The order in which these histograms are calculated and appended to the vectors is to be consistent across all the images as this is the information that indicates the region of the image where each number in the vector came from. Finally, the resulting feature vectors are passed on to a RF classifier for predictions.

My approach was tested on two publicly available now benchmark data collections, --Montgomery and Shenzhen (see Section 4.2)--. The results that I obtained are very promising as my system achieved an ACC of 83.33% with an AUC of 0.815 on the Montgomery collection and 96.35% ACC with 0.964 AUC on the Shenzhen set.

To the best of my knowledge, the system I propose outperforms all other reported ones that were trained and tested solely on either of the same two CXR collections [3, 28, 53, 32, 55, 54]. It is simple and economical --uses less memory space, processing time, and computing power-- compared to others that involve complicated procedures and/or complex machine learning models. This and the results that I have obtained show that my system can efficiently and successfully be used to automatically detect anomalies linked to TB in CXRs.

REFERENCES

[1] Lawtomated, "A.I. Technical: Machine vs Deep Learning." https://lawtomated.com/a-i-technical-machine-vs-deep-learning/, APR 2019. Accessed on 2020-21-04.

[2] Centers for Disease Control and Prevention, "Tuberculosis (TB)." https://www.cdc.gov/tb/publications/factsheets/general/tb.htm/, OCT 2011. Accessed on 2020-27-02.

[3] S. Vajda, A. Karargyris, S. Jäger, K. C. Santosh, S. Candemir, Z. Xue, S. K. Antani, and G. R. Thoma, "Feature selection for automatic tuberculosis screening in frontal chest radiographs," *J. Medical Systems*, vol. 42, no. 8, pp. 146:1–146:11, 2018.

[4] World Health Organization, "Tuberculosis." https://www.who.int/news-room/fact-sheets/detail/tuberculosis, OCT 2019. Accessed on 2020-27-02.

[5] S. Jaeger, A. Karargyris, S. Candemir, J. Siegelman, L. Folio, S. Antani, and G. Thoma, "Automatic screening for tuberculosis in chest radiographs: A survey," *Quantitative Imaging in Medicine and Surgery*, vol. 3, pp. 89–99, 04 2013.

[6] R. H. Sherrier and G. A. Johnson, "Regionally adaptive histogram equalization of the chest," *IEEE Transactions on Medical Imaging*, vol. 6, pp. 1–7, March 1987.

[7] R. Philipsen, P. Maduskar, L. Hogeweg, and B. Ginneken, "Normalization of chest radiographs," in *Medical Imaging 2013: Computer-Aided Diagnosis* (C. L. Novak and S. Aylward, eds.), vol. 8670, pp. 106–111, International Society for Optics and Photonics, SPIE, 2013.

[8] T. Matozaki, A. Tanishita, and T. Ikeguchi, "Image enhancement of chest radiography using wavelet analysis," in *Proceedings of 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 3, pp. 1109–1110, Oct 1996.

[9] C. Shuyue, H. Honghua, Z. Yanjun, and X. Xiaomin, "Study of automatic enhancement for chest radiograph," *Journal of Digital Imaging : the Official Journal of the Society for Computer Applications in Radiology*, vol. 19, pp. 371–375, 01 2007.

[10] L. Hogeweg, C. Sánchez, P. Jong, P. Maduskar, and B. Ginneken, "Clavicle segmentation in chest radiographs," *Medical Image Analysis*, vol. 16, pp. 1490–1502, 12 2012.

[11] A. Karargyris, S. Antani, and G. Thoma, "Segmenting anatomy in chest x-rays for tuberculosis screening," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 7779–7782, Aug 2011.

[12] M. Loog and B. van Ginneken, "Bony structure suppression in chest radiographs," in *Computer Vision Approaches to Medical Image Analysis* (R. R. Beichel and M. Sonka, eds.), (Berlin, Heidelberg), pp. 166–177, Springer Berlin Heidelberg, 2006.

[13] K. Suzuki, H. Abe, H. Macmahon, and K. Doi, "Image-processing technique for suppressing ribs in chest radiographs by means of massive training artificial neural network (mtann)," *IEEE Transactions on Medical Imaging*, vol. 25, pp. 406–416, 05 2006.

[14] P. Maduskar, L. Hogeweg, R. Philipsen, S. Schalekamp, and B. van Ginneken, "Improved texture analysis for automatic detection of tuberculosis (TB) on chest radiographs with bone suppression images," in *Medical Imaging 2013: Computer-Aided Diagnosis* (C. L. Novak and S. Aylward, eds.), vol. 8670, pp. 112–117, International Society for Optics and Photonics, SPIE, 2013.

[15] S. G. Armato, M. L. Giger, and H. MacMahon, "Automated lung segmentation in digitized posteroanterior chest radiographs," *Academic Radiology*, vol. 5, no. 4, pp. 245–255, 1998.

[16] L. Li, Y. Zheng, M. Kallergi, and R. Clark, "Improved method for automatic identification of lung regions on chest radiographs," *Academic Radiology*, vol. 8, pp. 629–638, 08 2001.

[17] B. Ginneken and B. ter Haar Romeny, "Automatic segmentation of lung fields in chest radiographs," *Medical Physics*, vol. 27, pp. 2445–2455, 11 2000.

[18] M. Loog and B. van Ginneken, "Supervised segmentation by iterated contextual pixel classification," in *Object Recognition Supported by User Interaction for Service Robots*, vol. 2, pp. 925–928, Aug 2002.

[19] M. Loog and B. Ginneken, "Segmentation of the posterior ribs in chest radiographs using iterated contextual pixel classification," *IEEE Transactions on Medical Imaging*, vol. 25, pp. 602–611, 06 2006.

[20] Zhanjun Yue, A. Goshtasby, and L. V. Ackerman, "Automatic detection of rib borders in chest radiographs," *IEEE Transactions on Medical Imaging*, vol. 14, pp. 525–536, Sep. 1995.

[21] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[22] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 681–685, June 2001.

[23] B. van Ginneken, A. F. Frangi, J. J. Staal, B. M. ter Haar Romeny, and M. A. Viergever, "Active shape model segmentation with optimal features," *IEEE Transactions on Medical Imaging*, vol. 21, pp. 924–933, Aug 2002.

[24] B. Ginneken, S. Katsuragawa, B. ter Haar Romeny, K. Doi, and M. Viergever, "Automatic detection of abnormalities in chest radiographs using local texture analysis," *IEEE Transactions on Medical Imaging*, vol. 21, pp. 139–149, 03 2002.

[25] B. Ginneken, B. ter Haar Romeny, and M. Viergever, "Computer-aided diagnosis in chest radiography: A survey," *IEEE Transactions on Medical Imaging*, vol. 20, pp. 1228–1241, 01 2002.

[26] E. J. Hwang, S. Park, K.-N. Jin, J. Kim, S. Choi, J. Lee, J. M. Goo, J. Aum, J.-J. Yim, and C. M. Park, "Development and validation of a deep learning-based automatic detection algorithm for active pulmonary tuberculosis on chest radiographs," *Clinical infectious diseases : an official publication of the Infectious Diseases Society of America*, vol. 69, 11 2018.

[27] N. N. Melena, S. Anusha, G. Chitra, and C. Sujatha, "Computer aided automatic detection of tuberculosis in chest radiographs," *International Journal of Science Technology & Engineering*, vol. 2, pp. 106–109, 2016.

[28] S. Jaeger, A. Karargyris, S. Candemir, L. R. Folio, J. Siegelman, F. M. Callaghan, Z. Xue, K. Palaniappan, R. K. Singh, S. K. Antani, G. R. Thoma, Y. Wang, P. Lu, and C. J. McDonald, "Automatic tuberculosis screening using chest radiographs," *IEEE Trans. Med. Imaging*, vol. 33, no. 2, pp. 233–245, 2014.

[29] S. Jaeger, S. Candemir, S. Antani, Y.-X. J. Wang, P.-X. Lu, and G. Thoma, "Two public chest x-ray datasets for computer-aided screening of pulmonary diseases," *Quantitative Imaging in Medicine and Surgery*, vol. 4, no. 6, 2014.

[30] T. Xu, I. Cheng, R. Long, and M. Mandal, "Novel coarse-to-fine dual scale technique for tuberculosis cavity detection in chest radiographs," *EURASIP Journal on Image and Video Processing*, vol. 3, pp. 1–18, 12 2013.

[31] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Y. Ding, A. Bagul, C. Langlotz, K. S. Shpanskaya, M. P. Lungren, and A. Y. Ng, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *CoRR*, vol. abs/1711.05225, 2017.

[32] F. Pasa, V. Golkov, F. Pfeiffer, D. Cremers, and D. Pfeiffer, "Efficient deep network architectures for fast chest x-ray tuberculosis screening and visualization," *Scientific Reports*, vol. 9, 12 2019.

[33] G. Wang, C. Lopez-Molina, and B. D. Baets, "Automated blob detection using iterative laplacian of gaussian filtering and unilateral second-order gaussian kernels," *Digit. Signal Process.*, vol. 96, 2020.

[34] Team OpenCV , "Laplace operator." https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/laplace_operator/laplace_operator.html/. Accessed on 2020-27-02.

[35] T. Ojala, M. Pietikäinen, and T. Mäenpää, "A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification," in *Advances in Pattern Recognition - ICAPR 2001, Second International Conference Rio de Janeiro, Brazil, March 11-14, 2001, Proceedings* (S. Singh, N. A. Murshed, and W. G. Kropatsch, eds.), vol. 2013 of *Lecture Notes in Computer Science*, pp. 397–406, Springer, 2001.

[36] A. Karargyris, J. Siegelman, D. Tzortzis, S. Jaeger, S. Candemir, Z. Xue, K. C. Santosh, S. Vajda, S. K. Antani, L. R. Folio, and G. R. Thoma, "Combination of texture and shape features to detect pulmonary abnormalities in digital chest x-rays," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 11, no. 1, pp. 99–106, 2016.

[37] K. C. Santosh, S. Vajda, S. K. Antani, and G. R. Thoma, "Edge map analysis in chest x-rays for automatic pulmonary abnormality screening," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 11, no. 9, pp. 1637–1646, 2016.

[38] Java T Point, "K-nearest neighbor(knn) algorithm for machine learning." https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning. Accessed on 2020-07-05.

[39] Deep Learning, "Multilayer perceptron." http://deeplearning.net/tutorial/mlp.html, JUN 2018. Accessed on 2020-07-05.

[40] Scikit Learn, "Support vector machines." https://scikit-learn.org/stable/modules/svm.html. Accessed on 2020-07-05.

[41] N. Donges, "A complete guide to the random forest algorithm." https://builtin.com/data-science/random-forest-algorithm, JUN 2019. Accessed on 2020-07-05.

[42] NumPy, "Numpy." https://numpy.org/. Accessed on 2020-14-05.

[43] Python Software Foundation [US], "Pickle – python object serialization." https://docs.python.org/3/library/pickle.html/. Accessed on 2020-14-05.

[44] Team OpenCV , "About opencv." https://opencv.org/about/. Accessed on 2020-27-02.

[45] Scikit-Image, "Image processing in python." https://scikit-image.org/. Accessed on 2020-14-05.

[46] Scikit-Learn, "Machine learning in python." https://scikit-learn.org/stable/. Accessed on 2020-14-05.

[47] PyMVPA, "Multivariate pattern analysis in python." http://www.pymvpa.org/installation.html/. Accessed on 2020-14-05.

[48] Keras, "About keras." https://keras.io/about/. Accessed on 2020-14-05.

[49] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861 – 874, 2006.

[50] N. A. Obuchowski, "Roc analysis," *Fundamentals of Clinical Research for Radiologists*, vol. 184, no. 2, pp. 364–372, 2005.

[51] A. Bosch and A. Zisserman, "Pyramid histogram of oriented gradients (phog)." https://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html. Accessed on 2020-27-02.

[52] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, 6 2014.

[53] S. Hwang, H.-E. Kim, J. J. M.D., and H.-J. Kim, "A novel approach for tuberculosis screening based on deep convolutional neural networks," in *Medical Imaging 2016: Computer-Aided Diagnosis* (G. D. Tourassi and S. G. A. III, eds.), vol. 9785, pp. 750 – 757, International Society for Optics and Photonics, SPIE, 2016.

[54] M. T. Islam, M. A. Aowal, A. T. Minhaz, and K. Ashraf, "Abnormality detection and localization in chest x-rays using deep convolutional neural networks," *CoRR*, vol. abs/1705.09850, 2017.

[55] S. Rajaraman and S. K. Antani, "Modality-specific deep learning model ensembles toward improving tb detection in chest radiographs," *IEEE Access*, vol. 8, pp. 27318–27326, 2020.