Spring 2020

Image Features for Tuberculosis Classification in Digital Chest Radiographs

Brian Hooper
Central Washington University, brian@brianhooper.org

IMAGE FEATURES FOR TUBERCULOSIS CLASSIFICATION IN DIGITAL CHEST RADIOGRAPHS

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

Brian Hooper

June 2020

CENTRAL WASHINGTON UNIVERSITY

Graduate Studies

We hearby approve the thesis of

Brian Hooper

Candidate for the degree of Master of Science

APPROVED FOR THE GRADUATE FACULTY

________________ ________________________________

Dr. Szilard Vajda

________________ ________________________________

Dr. Razvan Andonie

________________ ________________________________

Dr. Donald Davendra

________________ ________________________________

Dean of Graduate Studies

ABSTRACT

IMAGE FEATURES FOR TUBERCULOSIS CLASSIFICATION IN DIGITAL CHEST RADIOGRAPHS

by

Brian Hooper

June 2020

Tuberculosis (TB) is a respiratory disease which affects millions of people each year, accounting for the tenth leading cause of death worldwide, and is especially prevalent in underdeveloped regions where access to adequate medical care may be limited. Analysis of digital chest radiographs (CXRs) is a common and inexpensive method for the diagnosis of TB; however, a trained radiologist is required to interpret the results, and is subject to human error. Computer-Aided Detection (CAD) systems are a promising machine-learning based solution to automate the diagnosis of TB from CXR images. As the dimensionality of a high-resolution CXR image is very large, image features are used to describe the CXR image in a lower dimension while preserving the elements in the CXR necessary for the detection of TB. In this thesis, I present a set of image features using Pyramid Histogram of Oriented Gradients, Local Binary Patterns, and Principal Component Analysis which provides high classifier performance on two publicly available CXR datasets, and compare my results to current state-of-the-art research.

ACKNOWLEDGEMENTS

I'd like to thank my thesis advisor, Dr. Szilárd Vajda, for his support and valuable feedback during the research and development of this project, without whose encouragement I would not have persued higher education. I would also like to thank my thesis comittee, Dr. Donald Davendra and Dr. Razvan Andonie, for their valuable feedback and advice, both during the writing of this thesis and my graduate studies, as well as my colleagues in the Computer Science department, particulary Hermann Yepdjio, for their camaraderie and assistance during the graduate program. Finally, I would like to thank my partner Jodi Roush, and the rest of my friends and family, for their continuous patience and support.

TABLE OF CONTENTS

Chapter Page

TABLE OF CONTENTS (CONTINUED)

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

Tuberculosis (TB) is a respiratory disease caused by an infection of the bacteria *Mycobacterium Tuberculosis* in the lungs. TB can be contracted through the air by exposure to a person already infected with TB. In 2018, it was estimated that 10 million people contracted TB, and approximately 1.4 million people die from the disease every year. As of 2018, TB accounted for the tenth leading cause of death worldwide and the highest leading cause of death from a single infectious agent. TB is especially prevalent in underdeveloped regions, with eight countries accounting for two-thirds of new TB cases: India, China, Indonesia, The Philippines, Pakistan, Nigeria, Bangladesh, and South Africa [1].

Currently, the primary method for diagnosing tuberculosis is the detection of *Mycobacterium Tuberculosis* using sputum smear microscopy; however, this process can take several days or weeks for the sample to be identified, and the test can suffer from a high number of false positives. As such, it is frequently used in combination with the analysis of chest radiographs (CXRs), especially due to the wide availability and relative low cost of digital radiography machines. However, CXRs still require analysis by a trained radiologist, and are subject to human error and are dependent on the level of expertise of the radiologist. The difficulty in CXR analysis is compounded by the varying manifestations of TB on chest radiographs, with both the texture and geometry of the lungs affected. Overlapping tissue structures in the CXR increases the complexity of interpretation. Other methods, such as blood tests, can be more reliable than CXR diagnosis but are generally much more costly and time consuming, and so are much less commonly used than CXRs [2].

There is currently interest in applying Computer-Aided Detection (CAD) systems to the detection of tuberculosis and other respiratory diseases such as pneumonia. In regions lacking a sufficient number of trained radiologists, CAD systems could be used to help screen patients and highlight those with the greatest need for further treatment, and greatly reduce the time required to screen a large population [2]. However, much of the current research in developing CAD systems for use with CXRs is dedicated to early detection of lung cancer, with a comparatively small number of studies dedicated to TB and other similar pathologies [3]. Typically, CAD systems work by first pre-processing the CXR images, segmenting the region of interest (ROI), extracting image features, and classifying the disease [4]. Publicly available CXR datasets devoted to the diagnosis of TB and other pathologies have contributed to the increase in studies of CAD systems. Image feature descriptors can be used to reduce the dimensionality of CXR images, and increase the performance of a classifier system. The goal of this thesis is to develop a set of image features appropriate for the efficient and accurate classification of Tuberculosis in CXR images. While my primary focus will be on image features, I will study and compare different machine learning methods for image classification in order to effectively evaluate my results. I will test my feature descriptors and classifier models on two publicly available CXR datasets provided by Jaeger et al. [5]. The rest of this thesis is organized as follows: In Chapter II, I introduce current methods for TB diagnosis, CAD systems, and discuss current research in CAD systems for TB diagnosis. In Chapter III, I present the background of image feature descriptors. Chapter IV provides background information on Machine Learning classifier models. In Chapter V, I describe the descriptor and classifier models, experimental results, and analysis of results. Finally, in Chapter VI, I present my conclusions on the use of image feature descriptors for TB diagnosis in CXR images.

CHAPTER II

BACKGROUND

Tuberculosis Diagnosis

There are currently many methods for the diagnosis of Tuberculosis (TB), including sputum smear microscopy, and analysis of chest radiographs (CXRs). CXR diagnosis benefits from quick results, low cost, and ease of use. CXR images contain a wide range of information about a patients health, and can be used to detect various illnesses such as Pneumonia or lung cancer [3]. However, accurate assessment of a CXR is challenging, requiring a highly trained specialist to correctly interpret the image. Even expert analysis is not perfect, with one study from 1999 reporting that 19% of pulmonary nodules were undetected by expert radiologists [6].

Many factors contribute to the difficulty of analyzing a CXR for the presence of TB, including varying manifestations of TB, differences in image resolution and contrast, noise, and overlapping tissue structures. The manifestation of TB on a CXR image is complex, with a large number of abnormalities in the lung region that may or may not be present. These abnormalities include texture abnormalities, such as changes in appearance or structure, focal abnormalities, such as the presence of pulmonary nodules, and shape abnormalities, meaning changes in the lung contour [7].

Common CXR imaging manifestations of TB include lung cavitations, pulmonary consolidations, bilateral infiltrates, and pleural effusion, which can appear as blunted costophrenic angles [8]. The difficulty of TB diagnosis is compounded by differences in manifestations between active infections and inactive infections, meaning either patients who have been previously treated for TB, or patients who have been exposed to small colonies of TB bacteria which remains inactive in the body. Patients with inactive TB are at risk for TB infection if their immune system becomes compromised. Specifically, patients with Human Immunodeficiency Virus (HIV) are at significantly higher risk for TB, with TB accounting for one of the leading causes of death among people infected with HIV [1].

Computer-Aided Detection

In recent years, there has been an interest in the development of Computer-Aided Detection (CAD) systems for the diagnosis of TB. Such systems would reduce the time it takes to screen a large population, and more effectively filter patients with the highest need for further treatment. There are currently multiple commercial available CAD systems for the analysis of CXR images, including *CAD4TB* and *Riverain* [9]. However, because of the complexity of CXR images, the development of an effective CAD system is challenging, and the majority of commercially available CAD systems are dedicated to the detection of lung cancer, with the research towards detecting other types of pathologies relatively limited. Additionally, current commercial CAD systems do not match the performance of state of the art research systems, with one review of CAD4TB performance showing an AUC ranging from 0.71 to 0.84. A new version of CAD4TB, released in 2019, used a deep learning model trained on a dataset of 500 images from Pakistan, and achieved a specificity of 98% and a sensitivity of 90% [10].

Typically, CAD systems work in the following manner: pre-processing, segmentation, feature extraction, and classification. Segmentation, or region-of-interest extraction, isolates the lung regions within the image. This allows the feature extraction and classification steps to only act on those regions within the image that contribute to a positive or negative diagnosis, removing all other regions in the image that only act as noise. Automatic segmentation of lung regions is one of the most difficult aspect of CAD systems, and as such, there have been many studies that focus on lung segmentation methods. [5] [11].

Pre-processing includes any kind of image transformation that occurs prior to segmentation or feature extraction, such as resizing, cropping, rotation, equalization, or other image processing technique. Because contrast has a large influence on the detection of lung abnormalities, contrast enhancement can be applied to more effectively highlight these regions. Pre-processing steps can be an important factor in reducing the overall noise in a CXR image. Noise can be characterized in two categories: *radiographic noise*, resulting from variations in radiographic techniques and equipment, and *anatomical noise*, referring to the tissue structures, such as ribs or vascular structures, that surround and overlap the lungs. In CXR images, anatomic noise contributes significantly to the difficulty in detecting pulmonary nodules [12]. In addition to noise reduction, segmentation is important in defining the outer shape of the lung region. Deformations in lung shape, such as cavities, can contribute to the diagnosis of TB. In general, segmentation methods can be grouped into two categories: rule-based methods, and machine learning based methods. Rule-based methods include segmentation methods that use location, texture, and shape features to define regions of interest algorithmically. The category also includes deformable model based methods. Machine learning based methods use supervised or unsupervised learning to classify pixels as belonging to a particular anatomical structure.

As accurate classification of TB in CXR images requires high-resolution images, the dimensionality of the image causes challenges in training a classifier system. For example, a 1000 by 1000 pixel image contains one million dimensions. As such, methods for dimensionality reduction, such as image feature descriptors, are typically used to reduce the size of the dataset required to train a classifier. I describe various feature descriptors in Chapter III. In some systems, such as Convolutional Neural Networks, feature extraction and classification are combined into one step.

Related Work

In recent years, various CAD systems have been developed using feature extraction and image classification methods for CXR diagnosis. Vajda et al. [8] considered three feature sets for classification of segmented CXR images from the Montgomery and Shenzhen CXR datasets [5]. Set A consisted of shape, edge, and texture descriptors, with an overall vector length of 192. Set B consisted of 595 intensity, edge, texture, color, and shape moment features. Set C contained only shape measurements, with a much smaller set of only 12 features. Using a neural-network based classifier on the Montgomery dataset, the authors obtained an AUC of 0.87, 0.72, 0.71 on sets A, B, and C, respectively. With the Shenzhen dataset, an AUC of 0.99, 0.90, and 0.77 was achieved.

Jaeger et al. [5] created an effective algorithm for automatic lung boundary segmentation. Using a content-based image retrieval method combined with a set of manually segmented training images, the authors matched patient CXRs to the closest matching training images, and then warped the patient CXR image to the training set using a nonrigid registration algorithm. This work provided the segmentation used in the Montgomery and Shenzhen datasets.

Hogweg et al. [7] used lung sub-segmentation to extract images features from four sections for each lung: lower, middle, central, and upper. Using two datasets consisting of 200 CXR images each, the researchers achieved a best AUC for TB detection of 0.90. Automatic segmentation was achieved using a combination of pixel classification and shape model information. This method of incorporating spatial data is promising for the development of a CAD system because TB can show as present different anomalies on a CXR. However, the precise segmentation requirements of this method make its use on low-quality images challenging.

Xue et al. [13] proposed a CAD system to distinguish between frontal and lateral CXR images. Using a combination of image profile, body shape, Pyramid Histogram of Oriented Gradients (PHOG), and contour-based shape features, the authors achieved a very high accuracy of 99.9% on a CXR dataset containing 8300 images provided by University of Indiana School of Medicine .

Carrillo-de-Gea et al. [14] created a CAD system to classify healthy lungs from those with any form of non-normality or pathology present based on an ensemble of location-specific classifiers. For their training and test data, the researchers collected a private dataset of CXR images from 25 male patients and 23 female patients. By applying Local Binary Pattern (LBP) features to the image, they created an ensemble of classifiers by training individual classifiers on local lung regions. With this method, the they achieved a highest accuracy of 70% . While the overall accuracy is low, this result is significant given that the dataset consisted of only 48 samples.

More recently, deep convolutional neural networks have been applied to TB detection in CXR images. Hwang et al. [15] used transfer learning on a deep CNN with the AlexNet architecture, achieving an accuracy of 67% on the Montgomery dataset, and an accuracy of 83% on the Shenzhen dataset. Pasa et al. [16] developed a deep-learning model with significantly lower hardware requirements than previous CNN-based CAD systems. The authors trained the model using the Montgomery and Shenzhen datasets and achieved an AUC of 0.811 and 0.9, respectively. Compared to Hwang et al., the authors achieved similar classifier performance, but with a more efficient CNN model and without using transfer learning.

Similarly, CNNs have been used as feature extraction methods, while using a traditional model for classification. Allaouzi and Ahmed [17] used a pre-trained CNN as a feature extractor on CXR images from the ChestX-ray14 and CheXpert datasets. The authors used the DenseNet-121 CNN architecture with transfer learning from ImageNet as a feature extractor to give a feature vector of length 1024. For classification, they used a Logistic Regression model and to predict the probability that each sample belonged to each of the 14 labels in the ChestX-ray14 and CheXpert datasets. Metrics were calculated by taking an average of the binary classification accuracy across all labels. For the ChestX-ray14 dataset, the researchers obtained an AUC of 0.88, and an AUC of 0.81 on the CheXpert dataset. Lopes and Valiati [18] used pre-trained convolutional network as a feature extractor to train a support vector machine classifier, and achieved an ACC of 83% and an AUC of 0.92 on the Montgomery dataset and an ACC of 85% and an AUC of 0.93 on the Shenzhen dataset. Overall, feature descriptor based methods have been more effective for TB classification than CNN-based methods, with generally lower hardware requirements for both training and classification.

CHAPTER III

IMAGE FEATURE DESCRIPTORS

This chapter describes the image feature descriptors used for my experiments. Because the dimensionality of a CXR images is high (1 million data points for a 1000 by 1000 grayscale image), image features are extracted to attempt to describe the image in a lower dimension. I experimented with various feature extraction method in an attempt to find a set of feature descriptors that is able to effectively classify the presence of Tuberculosis in a CXR image with a minimum number of features.

Feature extraction is closely related to the problem of compression, that is, what is the minimum number of dimensions that can be used to represent the data, while still preserving some necessary element of the data (in this case, the presence or lack of TB manifestations on CXR images) [19]. In many cases, better classifier performance can be achieved with a selection of features than with the original data. This may be due the curse of dimensionality, the idea that as the dimensionality of your data increases, the number of samples required to effectively train a classifier increases exponentially.

A distinction should be made between feature selection, and feature extraction. In feature selection, a subset of features in a data is taken, and the rest of the features are discarded, with the idea to keep only the features that contribute the most to the correct classification and reduce unnecessary noise. Typically, feature selection involves some form of feature ranking, where the variables are ordered by some measure of their relevance for classification. In feature extraction, new features that describe some aspect of the data, such as texture or shape, are generated from the original data [20]. Filters, transformations, statistical measures, shape and texture analysis, and interest point detection are all forms of feature extraction methods.

Pyramid Histogram of Oriented Gradients

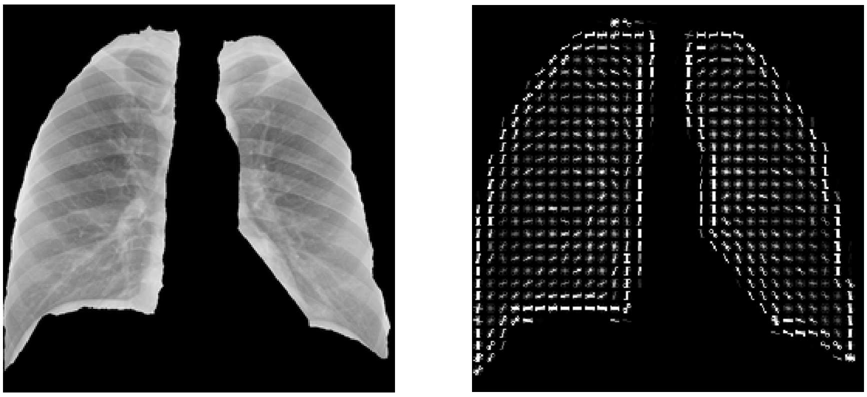
The Pyramid Histogram of Oriented Gradients (PHOG) feature descriptor is a variation of the Histogram of Oriented Gradients (HOG) image descriptor, which computes the occurrence of gradient orientation over a grid of cells in an image. The original HOG descriptor was developed in 2005 and has shown to be useful for pedestrian detection and handwriting recognition [21]. PHOG was first described by Bosch et al in 2007 [22]. The PHOG descriptor divides the image into sub-regions at various resolutions, and calculates the HOG descriptor for each spatial pyramid, which is then concatenated into one feature vector [23]. Both PHOG and HOG output a scale-invariant feature vector of fixed size depending on the input parameters, which makes it suitable for using as input to a classifier.

The HOG descriptor computes occurrences of gradient orientation over a dense grid of uniformly space cells. Across the whole image, the horizontal and vertical gradients are calculated for each pixel. This is achieved by first applying a Sobel filter to the image with a kernel size of 1, and then computing the magnitude and direction of the gradient for each pixel using equations 3.1 and 3.2, respectively, where g_x and g_y represent the Sobel filtered value for the pixel in the horizontal and vertical directions.

$$g = \sqrt{g_x^2 + g_y^2} \tag{3.1}$$

$$\theta = \arctan \frac{g_y}{g_x} \tag{3.2}$$

Figure 1 shows a subset of pixel gradients computed over a CXR image. For each 8x8 cell in the image, a single gradient is shown, with the magnitude represented by the length of the line and the line running perpendicular to the direction of the gradient. Numerically, I use unsigned gradients with a range of 0 to 180 degrees to represent the angle of the gradient, with a gradient and its negative represented by the same direction. In practice, this method has been shown to be more effective than using signed gradients [21].



(a) Original Image

(b) Gradient orientations

FIGURE 1: Visualization of Histogram of Oriented Gradients feature descriptor.

In order to encode location data into the feature descriptor, we divide the image into a uniform grid of fixed-size cells, and compute a histogram of gradient orientations for each cell. These histograms are concatenated together to provide the final HOG descriptor vector. For my experiments, I used a cell size of 8 by 8 pixels. While somewhat arbitrary, this cell size is sufficient to detect the smallest features necessary for the recognition of TB, provided that the resolution of the CXR images is sufficiently large, and my initial experimentation showed little change in classification accuracy with smaller or larger cell sizes.

As we have 2 values per pixel (gradient and magnitude), using an 8x8 grid gives us 128 pixel values per cell. Each histogram consists of 9 bins, corresponding to angles 0-19, 20-39, etc. Each pixel's magnitude is added to the respective bin based on its magnitude. For example, a pixel with angle 25 and magnitude 5 would have 5 added to the second bin in the histogram. Before each cell's histograms are concatenated together, the histograms are normalized relative to each other. The purpose of this step is to reduce the variance from lighting across the image. For normalization, we use a sliding window consisting of a 2x2 grid of cells, normalizing the block of four cells together, and moving the window by one cell across the image until all histograms are normalized. Finally, the histograms are concatenated together to provide the final HOG feature vector. In the PHOG algorithm, HOG features are computed over an image pyramid, by filtering and resampling the image at different resolutions, computing HOG features at each resolution, and concatenating the results into a final feature vector. Figure 2 shows an example of an image pyramid resampled at 4 levels.

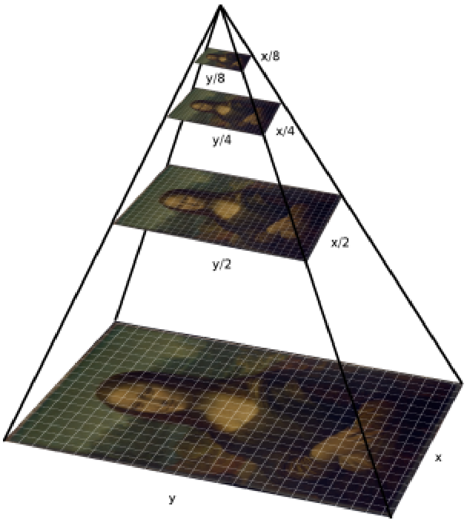


FIGURE 2: An image pyramid [24].

Local Binary Patterns

The Local Binary Patterns (LBP) feature descriptor is a local texture descriptor that encodes each pixel in an image by thresholding its intensity based on its eight neighbors. For each pixel, a new value is computed by creating an vector of 8 bits, assigning a value to each bit by comparing the pixel to each of its eight neighbors, starting with the upper left pixel and moving clockwise. For each neighbor, a value of 1 is assigned if the center pixel has an intensity higher than or equal to the neighbor, and a value of 0 is assigned if the pixel has an intensity less than the neighbor. This corresponds to an 8-bit value, which is assigned as the new value for the pixel in the output mask. Figure 3 shows the mask created by the LBP algorithm.

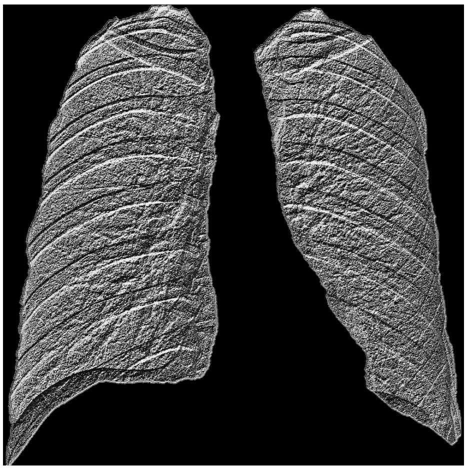


FIGURE 3: Local Binary Patterns mask computed for a segmented image in the Shenzhen dataset.

Because the feature descriptor does not reduce the dimensionality of the image, we compute a histogram of pixel values, resulting in a feature vector of length 256. As the segmented CXR images have large regions of black pixels, we remove this value from the histogram, and normalize the final length 255 histogram.

As the final feature vector of LBP is a histogram, any location-based data in the image is lost. Instead of computing one histogram over the entire image, the LBP mask can be divided into segments, a histogram can be computed for each segment, and then each histogram can be concatenated together to create a single output vector. This method is similar to the grid of histograms used in the HOG descriptor. To reduce the length of the output vector, a smaller number of bins for each histogram can be used. As the location of texture features can be an indicator of TB infection, incorporating location data into the LBP feature descriptor should improve classifier performance.

Autoencoder Networks

An Autoencoder network is type of artificial neural network that uses unsupervised learning to re-create its input as its output, while passing the data through one or more smaller hidden layers. Functionally, an autoencoder is similar to a multi-layer perceptron model. The first half of the network acts as an encoder, mapping the input to a smaller feature space, while the second half acts as a decoder, attempting to re-create the input data based on the encoded data. For this reason, autoencoders are typically symmetrical, with the decoder consisting of the same steps as the encoder, but in reverse. If the central hidden layer (acting as the output of the encoder, and the input to the decoder) has significantly smaller dimensionality than the original data, the encoded data should represent the data most important for the reconstruction of the original data. As such, autoencoder networks are an effective method for both image feature extraction and data compression. Figure 4 shows an example autoencoder model with three fully connected hidden layers, with the original data consisting of 5 variables and the encoded data consisting of two variables. The first autoencoder network was proposed by D.E. Rumelhart et al. in 1985, and has been used successfully for both dimensionality reduction and compression [25]. As a method for dimensionality reduction, an optimally trained autoencoder produces an encoding similar to principal component analysis (PCA) [26].

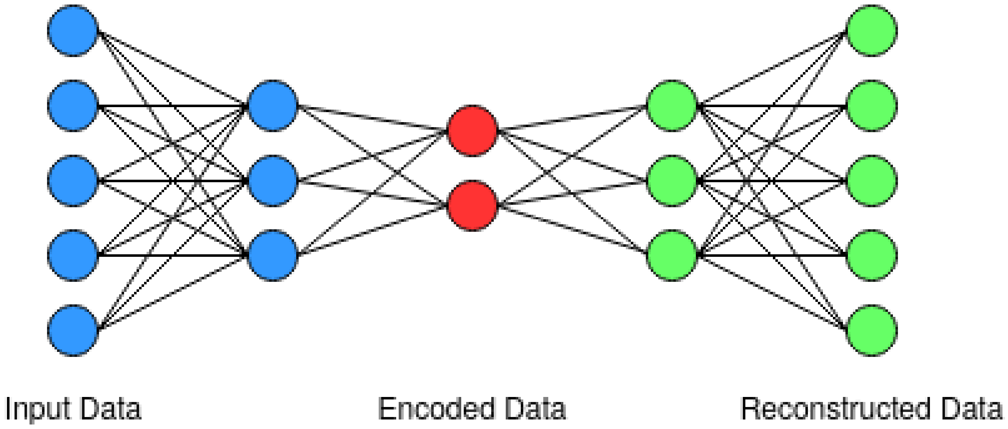


FIGURE 4: A schematic of an autoencoder network with three fully connected hidden layers.

Speeded-Up Robust Features

Speeded up robust features (SURF) are a faster variation of the Scale Invariant Feature Transform (SIFT) algorithm, which detects local features in an image. Like SIFT features, SURF is scale-invariant, meaning the same interest points can be found at different image sizes. In SURF, interest points are detected using a Hessian matrix approximation, and for each interest point, a local feature descriptor containing 64 features is computed. An full description of the SURF algorithm is given in [27]. SURF features, and other similar blob-detection algorithms, are commonly used in image-retrieval systems, where a compact description of the image is required. However, because the number of local features extracted using SURF varies for each image, it cannot be directly used in a classifier model that takes a fixed-length vector as an input. For this reason, SURF features are typically used for classification by quantizing detected features into a fixed set of clusters, using K-Means or another similar clustering algorithm. This method is commonly used in content-based image retrieval applications [28] and has been successfully applied to medical image classification [29].

Principal Component Analysis

Generally, the main purpose of feature extraction is to reduce the dimensionality of the image, while preserving the elements that contribute the most to successful classification. As such, we can use well-known dimensionality reduction methods such as Principal Component Analysis (PCA). The PCA method was first proposed in the early 20th century, but was not commonly used until advances in computing power made working with large dimension datasets possible [30]. PCA creates an orthogonal linear transformation of the data, by deriving a set of principal components which maximize the variance in the data.

The operation of the PCA algorithm as as follows: given a set of n samples, each with m dimensions which we want to reduce to k dimensions, we take a n by m matrix X such that each row represents a single sample and each column represents a single variable. The covariance matrix C_x can be calculated as $C_x = \frac{1}{n-1}(X - \bar{X})(X - \bar{X})^T$. Given that C_x represents a linear transformation of X, we can calculate the eigenvalues and corresponding eigenvectors of the transformation. Eigenvalues and eigenvectors represent properties of linear transformations in matrices. Specifically, an eigenvector is a vector measurement of the direction of a transformation, and eigenvalues represent a scalar measurement of the factor by which an eigenvector is scaled. Formally, eigenvalues and eigenvectors represent a property of a matrix such that $Ax = \lambda x$, where A represents a matrix, x represents the eigenvector, and λ represents the corresponding eigenvalues. By ranking the eigenvalues from largest to smallest, we can take the first k eigenvectors, which represent the k most significant components, giving us a n by k matrix E. Finally, we transform our original dataset X by taking the transpose of the eigenvector matrix multiplied by our original matrix X. Therefore, given a set of original values, we can substitute a set of optimal derived principal components with lower dimensionality than the original dataset. PCA is often used by reducing the dataset to two or three principal components, which allows a higher dimension dataset to be visualized in two or three dimensions.

Other Feature Descriptors

In addition to the feature descriptors mentioned above, I examined Zernick Moments, Gabor Filters, Gray-Level Co-Occurence Matrix, Haralick texture features, Determinant of Hessian, and Oriented Fast and Rotated BRIEF features. However, the performance of my classification experiments with these descriptors was poor, as such, I will not describe their operation in this paper.

CHAPTER IV

IMAGE CLASSIFICATION

In this chapter, I describe the various machine learning models used in this thesis. While I will attempt to defend my choice to use each model, when selecting a classifier, there is often little indication of the potential for success of one model compared to another, as such, I will use the models that have proven to be successful in both CXR classification and other computer vision applications.

Classification Metrics

To evaluate the performance of the classifier models, I used accuracy (ACC) and area under the curve (AUC). In most image classification systems, ACC is the primary metric of classifier performance [3]. However, the simple ratio of correctly classified samples to incorrectly classified samples in the dataset is not sufficient for working with medical data. Specifically, we cannot have an effective classifier that has a chance of classifying a patient infected with TB as healthy, and so it is significantly more important to minimize the number of false negatives than it is to minimize false positives. Therefore, we calculate the receiver operating characteristic (ROC) curve, which plots the ratio of the true positive rate of classified samples to the false positive rate:

$$TPR = \frac{TP}{TP + FN} \tag{4.1}$$

$$FPR = \frac{FP}{FP + TN} \tag{4.2}$$

From the ROC curve, we can calculate the area under the ROC curve (AUC) across the unit square, which provides a useful metric of classifier performance between 0 and 1:

$$AUC = \int_{-\infty}^{\infty} TPR(T)FPR'(T)dT \tag{4.3}$$

AUC better encapsulates the performance of the classifier for medical purposes than by ACC alone [31]. Additionally, ACC and AUC are commonly used in image classification literature, so this allows me to preserve compatibility with other authors work. Therefore, for all classification experiments, I consider both ACC and AUC.

Multi-Layer Perceptron

The Multi-Layer perceptron (MLP) classifier is the most common type of artificial neural network classifier, consisting of fully connected layers of artificial neurons (nodes). MLP classifiers have been used extensively for image recognition tasks, including classification of TB in CXR images by Vajda et al. [8]. Therefore, the use of this classifier will allow me to easily compare the performance of my set of image features to other current research. A MLP always contains an input layer with a number of nodes equal to the dimensionality of the data, an output layer, and one or more hidden layers, containing a variable number of nodes. There is evidence that any mathematical model can be represented with a single hidden layer, and my empirical testing showed no increase in classifier performance from using multiple hidden layers. With the exception of the nodes in the input layer, each node in a MLP uses an activation function to map the sum of its inputs to its output. In my experiments, I use two common activation functions, The rectifier linear unit (ReLu), given in Equation 4.4 and SoftMax, given in Equation 4.5.

$$f(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases} \tag{4.4}$$

$$s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{4.5}$$

Random Forest

The Random Forest algorithm is a supervised machine learning algorithm that uses an ensemble of randomized decision trees. Decision trees were one of the first classification algorithms, and have been successfully used in a wide variety of classification problems. Individual decision trees can be very fast, both for training and prediction, but can be highly sensitive to overfitting. The Random Forest classifier attempts to mitigate this problem, by taking an ensemble of randomized decision trees, the output can be averaged, reducing the variance.

In general, a higher number of trees in the forest increases the performance of the classifier, but at the cost of slower prediction time. A Random Forest classifier is trained by bagging, where each tree is trained on a random sample with replacement of the original dataset. Figure 5 shows an visualization of a Random Forest classifier consisting of three decision trees.

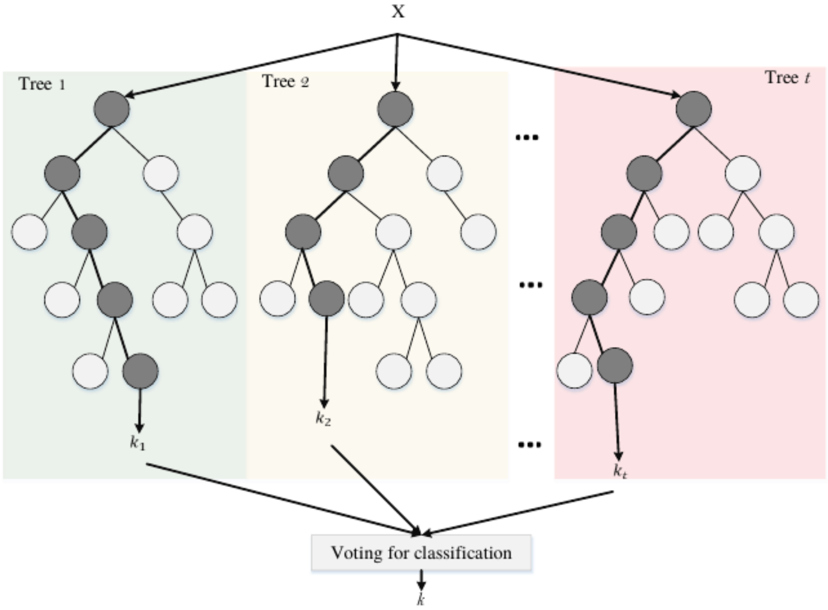


FIGURE 5: The Random Forest algorithm [32].

Ensemble Classifiers

In some cases, a combination of individual classifiers can be more effective than an individual classifier alone. An ensemble classifier, or multiclassifier system, combines multiple classifiers and aggregates their output to produce a single prediction. An typical analogy to a multiclassifier system is a panel of experts making a decision by majority vote. Multiclassifier systems allow different classifiers to take different approaches to classification in order to better classify data. In a multiclassifier system, the output of each classifier is aggregated to produce a single prediction, either by a majority vote or other aggregation method [33]. In Figure 6, an example ensemble classifier with 5 individual classifiers is shown.

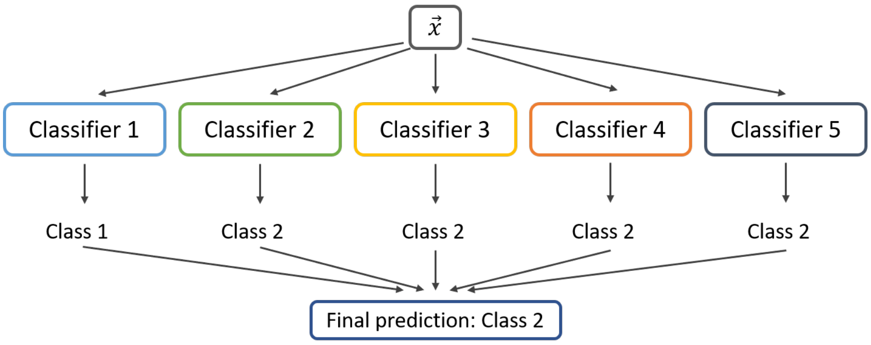


FIGURE 6: An example ensemble classifier system with 5 individual classifiers [34].

An ideal ensemble classifier does not necessarily require that the individual classifiers are completely error free, provided that each individual classifier makes different kinds of errors. That is, the samples that are incorrectly classified by one classifier have little overlap with the types of errors made by another classifier, and therefore the errors produced by any one classifier can potentially be canceled out by the correct classifications performed by other classifiers. A useful measure of the performance of a multiclassifier system is the Jaccard Index, given in Equation 4.6

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.6}$$

Given each set of samples that was classified incorrectly by each individual classifier, we can measure the complimentarity of the classifiers relative to each other by taking the intersection of the sets divided by the union of the sets. This produces a value between 0 and 1, where 0 represents all classifiers making exactly the same mistakes and 1 represents no two classifiers making any of the same mistakes. Equation 4.6 is shown for a multiclassifier system consisting of two classifiers; however, the Jaccard Index can be generalized for any number of classifiers.

Convolutional Neural Networks

In recent years, Convolutional Neural Networks (CNNs) have been introduced as one of the most promising methods for image classification. Breakthroughs in computing power, as well as the availability of large training datasets, have made possible more complex neural networks which have in some cases reached levels of performance comparable to humans. Additionally, CNNs act as both a feature extractor and a classifier, eliminating the need for a separate feature extraction step. Typically, CNNs are comprised of an input layer, followed by one or more convolutional and pooling layers, which are then fed into a fully connected neural network. A convolutional layer operates by passing a filter kernel over the image, multiplying the kernel matrix by the underlying pixels at each step. Pooling layers reduce the dimensionality of the image by sub-sampling. Typically, a pooling layer passes a filter across the image, with each step taking a statistical measure such as the sum, average, or maximum. An example CNN architecture with convolutional and pooling layers is shown in Figure 7. In general, CNNs with multiple convolutional and neural networks are the most effective at image recognition tasks.

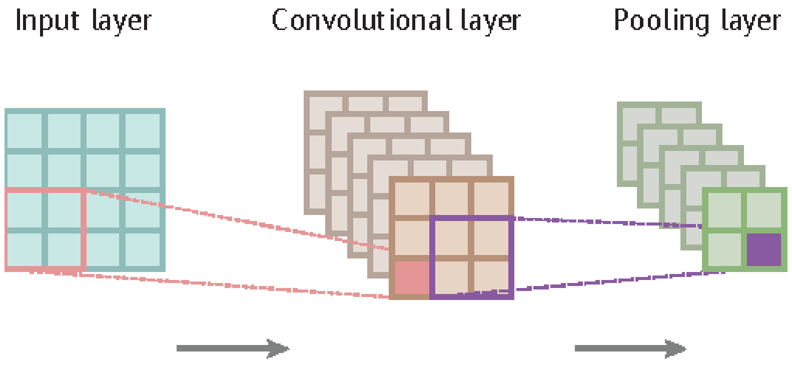


FIGURE 7: An example CNN architecture showing convolutional and pooling layers [35].

Convolutional Neural Networks (CNNs) have shown promising results in medical image classification, and promising results have been achieved using CNNs to detect TB in CXR images [36]. However, CNNs typically require a very large amount of training data, and so are limited in their application on small datasets. Additionally, because CNNs combine feature extraction and classification, the training time and computational requirements of CNNs are frequently much higher [17]. Some success with smaller datasets have been reported using transfer learning; however, this method has not so far surpassed the accuracy of explicit feature-extraction based methods for TB detection [15].

Other Classifier Models

In addition to the classifier models described in this chapter, I performed experiments with Support Vector Machines, Stochastic Gradient Descent classifiers, Naive Bayes classifiers, and K-Nearest Neighbor classifiers. However, as the performance of these classifiers was poor, I will not describe their operation here.

CHAPTER V

EXPERIMENTAL RESULTS

In Chapter III, I outlined many feature extraction and dimensionality reduction methods, and in Chapter IV I discussed multiple classifier models. In this chapter, I discuss my experiments to develop a set of CXR image features effective for use in a CAD system for Tuberculosis detection, as well as my experiments in lung region segmentation. For my classification experiments, I used Multi-Layer Perceptron, Random Forest, Gaussian Process, Support Vector Machine, and K-Nearest Neighbor classifiers. Of these classifiers, SVM and KNN did not achieve meaningful classification accuracy with any feature set during my initial experimentation, and as such were excluded from later experiments.

Datasets

To test my feature descriptors and classifier models, I use two publicly available CXR datasets, the Shenzhen dataset containing 662 samples, and the Montgomery dataset containing 138 samples. For both datasets, lung region segmentation was provided by Jaeger et al. [5]. In order to train the classifiers, each dataset was split into 80% of samples for training, and reserved 20% of samples for testing, with the samples in each category randomized for each experiment. The size of this split allows the performance of my classifier to be easily assessed given the small size of the datasets. Additionally, for my experiments with segmentation, I examine the CheXpert dataset, a large multi-class CXR dataset containing 223,648 samples. More information on these datasets is available in Appendix A.

Image Processing

Typically, CAD systems apply various forms of pre-processing to the CXR images in order to enhance the image quality, or to reduce variation across the dataset [2]. For my experiments with the Shenzhen and Montgomery datasets, it was desirable to scale the images to be a consistent size across all samples. However, the aspect ratio varied between images, and it was necessary to preserve the original aspect ratio of the image. Therefore, in the pre-processing step, I isolated the region of interest, removing unnecessary black pixels on the sides of the image. Next, I downscaled the image so that the large of the two dimensions was 1000 pixels. Finally, I padded the smaller of the two dimensions equally on each side, so that the final dimension of all images was 1000 by 1000 pixels, with the original aspect ratio of the images preserved and the region of interest centered in the image.

Segmentation

In order to segment the images in the CheXpert [37] dataset to improve classifier performance, I experimented with two segmentation methods: a rule-based method using Canny edge detection, and pixel-classification based methods using K-Means clustering and Self-Organizing Maps (SOM). Similar methods have been applied to image segmentation with varying degrees of success, but is typically suceptable to noise [38]. The rule-based segmentation used a Canny edge detector followed by pixel gradient detection. For each CXR image, I first extracted the edge pixels in the image using a canny edge detector, then split the image into left and right lungs by measuring the vertical column of pixels with the lowest number of white pixels in the middle one-third of the image. This method proved successful for images that are aligned so that a clear vertical line can be drawn that separates the left and right lung. However, if the image is rotated so that a vertical line cannot be drawn without intersecting at least one lung, this method is ineffective. In general, this method was suitable for the majority of images in the CheXpert dataset. For the left and right lung regions, I found the pixels with the greatest gradient (light to dark) in all four directions (top to bottom, left to right, etc).

I used this set of interest points for each direction to create a mask between the edge of the image and the set of points, and subtracted those masks from the image. Figure 8 shows the interest points extracted from the left lung as well as the corresponding segmented image. While this segmentation method was effective for some images, it was computationally expensive, and was susceptible to noise in the image, particularly around the rib bones that overlap the lung region. Additionally, there were many edge-cases for which this segmentation was ineffective, particularly if the left and right lungs were unable to be separated due rotation or noise in the image.

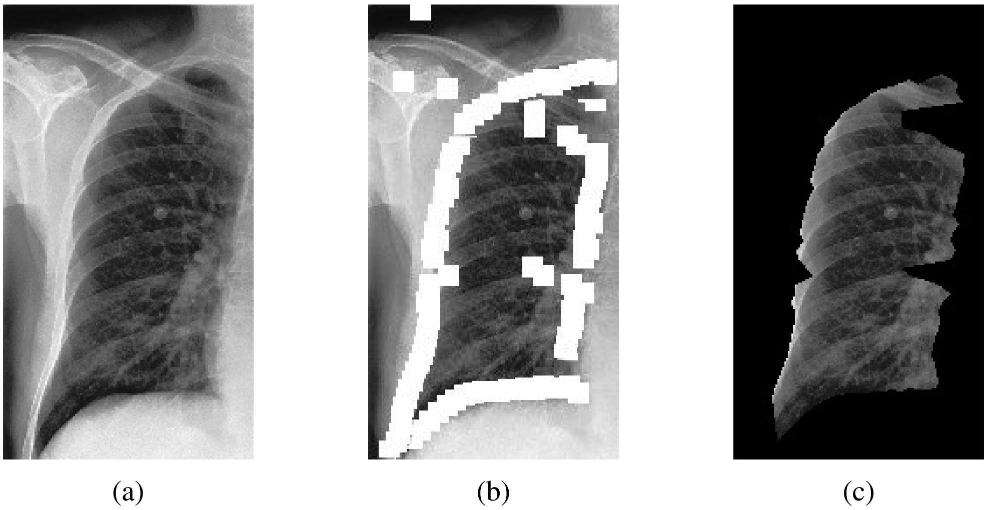

(a) (b) (c)

FIGURE 8: Interest points extracted from the left lung using the rule-based method.

For pixel-classification based segmentation I experimented with two methods: K-Means Clustering and Self-Organizing Map (SOM). Figure 9 shows the pixel regions classified using K-means with 2, 4, and 16 clusters. Typically, the higher number of clusters used, the more likely the separation between the lung region and the surrounding tissue could be identified. However, there was no single number of clusters that was effective for all images in the dataset. My experiments with Self-Organizing Map (SOM) pixel classification had similar results to my K-Means method. Figure 10 shows two images after SOM classification. SOM was more effective than K-means for images with high contrast between the lung region and the surrounding tissue; however, SOM was ineffective at classifying images with low contrast and was unable to classify regions with overlapping tissue structures. Because of the poor performance of my segmentation methods on the highly varied images in the CheXpert dataset, I chose to focus my research on the Shenzhen and Montgomery datasets using the high-quality segmentation provided by Jaeger et al. [5].

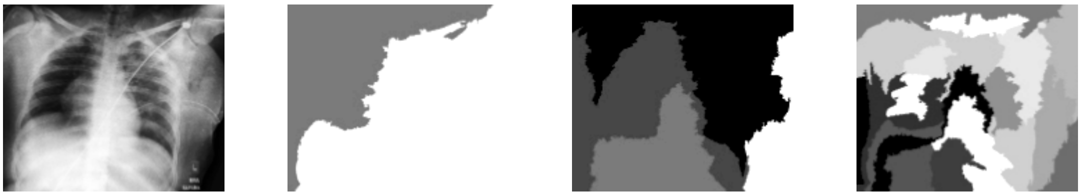


FIGURE 9: K-Means segmentation with 2, 4, and 16 clusters.

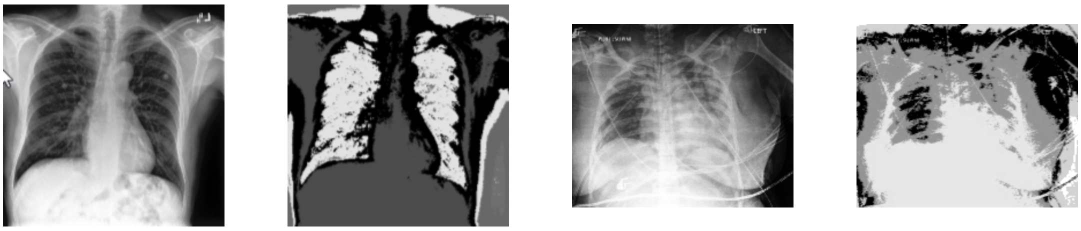


FIGURE 10: Pixel regions classified using self-organizing maps.

Classification

To compare and evaluate the performance of my feature descriptors, I experimented with various supervised machine learning models, including Multi-Layer Perceptron, Random Forest, Gaussian Process, Support Vector Machine, and K-Nearest Neighbor. Details on the operation of these models is given in Chapter IV. Here, I describe the parameters and inputs to each classifier model.

Classifier Models

The Multi-Layer Perceptron (MLP) classifier consisted of an input layer containing a number of nodes equal to the size of each feature vector, a single fully connected hidden layer, and an output layer. Each node used ReLu as the activation function. I experimented with two methods for the number of nodes in the output layer: a single output node, where an activation less than or equal to 0.5 corresponded to the negative class, and an activation greater than 0.5 corresponded to the positive class, and two output nodes, with one node corresponding to the negative class and the other node corresponding to the positive class. After some experimentation, I determined that using two nodes in the output layer provided slightly better classifier performance. For the hidden layer, I found that modifying the number of nodes in the hidden layer had little effect on the performance of the classifier, as such, I settled on 32 nodes. Finally, I found that 40 training epochs provided the maximum classifier performance without overfitting. All MLP experiments were performed using the Keras machine learning library in Python version 3.6 on Ubuntu Linux 19.04.

For the Random Forest classifier, I performed hyperparameter optimization using grid search with cross validation, with optimal performance achieved using 100 estimators with a maximum depth of 7. For both my experiments with the Gaussian Process classifier and Support Vector Machine, I used the Radial-Basis Function kernel. Finally, for K-Nearest Neighbors, I used the KDTree algorithm using the Euclidian distance metric. All RF, GP, SVM, and KNN experiments were performed using the Scikit-Learn library in Python version 3.6.

Feature Descriptors

Pyramid Histogram of Oriented Gradients

Two sets of Pyramid Histogram of Oriented Gradients (PHOG) features were extracted from the CheXpert, Shenzhen, and Montgomery datasets: 3 pyramids, with 20 bins per histogram, and 2 pyramids with 10 bins per histogram, resulting in feature vectors of length 1700 and 210, respectively.

Local Binary Patterns

I extracted three primary Local Binary Patterns (LBP) feature descriptors.

First, by simply taking a histogram of pixel values across the entire image, resulting in a feature vector of length 255 with black pixels removed, as I ignore the empty space outside the lung contour. Before classification, the histogram values are normalized between 0 and 1.

Second, in order to attempt to encode location-based data in the LBP histogram, I compute LBP features across the entire image, divide the image into a uniform grid of cells, and compute a histogram of each cell, concatenating the histograms together to produce the final feature vector. In order to determine optimal parameters, I experimented with taking 16, 32, 128, and 256 cells across the image, as well as computing 5, 10, and 20 bins per histogram. While I expected encoding the additional location data would increase the performance of the classifier, this method was less effective than taking a single histogram across the entire image.

Speeded-Up Robust Features

Because the SURF algorithm outputs a variable number of interest points, I used K-Means clustering to quantize the SURF descriptors into a fixed-length feature vector. This method is commonly used when using SURF features for classification, and is similar to the bag-of-words approach common in natural language processing applications [32]. In order to avoid cross-contaminating the test data, I train the K-Means model on the training set only, and apply the same transformation to the set of samples reserved for testing. For my experiments, I computed SURF features with 50 and 100 cluster centers. In the set of SURF interest points detected on a CXR image shown in Figure 11, we can see that the vast majority of interest points are located on the contour of the lung, with no interest points in the center of the lungs. However, the interest points alone represent a rudimentary outline of the lung shape, with low granularity for detecting small features along the contour. This pattern was similar among all samples in the Shenzhen dataset. As such, it is likely that SURF features alone are insufficient for the detection of TB in CXR images.

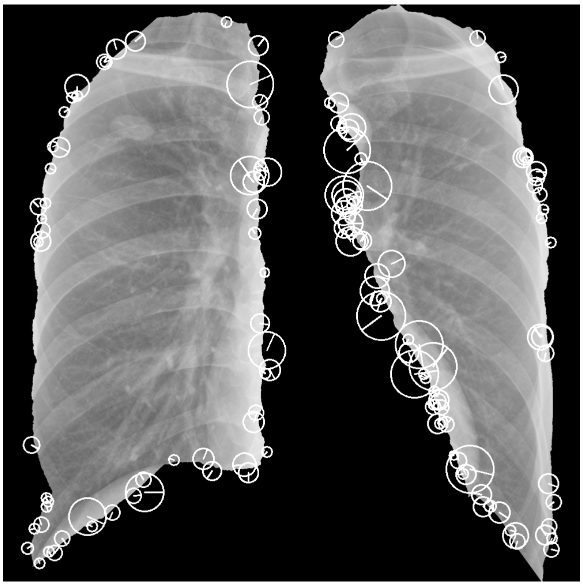
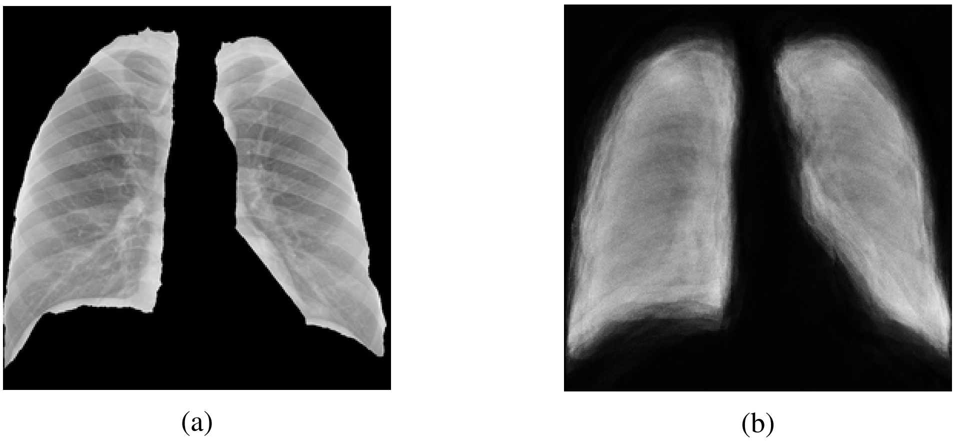


FIGURE 11: A set of SURF interest points extracted from a CXR image in the Montgomery dataset.

Autoencoder Networks

For my experiments with autoencoder networks, I scaled the images in the Shenzhen dataset to 250 by 250 pixels, giving us a flattened feature vector of length 62,500. The encoder network consisted of an input and output layer with 62,500 nodes each, and a single hidden layer with 64 nodes. The pixel values in each image were normalized between 0 and 1 before passing them to the network. Figure 12 shows an input CXR image on the left, and the output from passing the image through the trained autoencoder on the right.



(a) (b)

FIGURE 12: A CXR image from the Shenzhen dataset before and after passing through the trained autoencoder network.

In order to use the encoded images in a classifier, I trained the autoencoder on the set of training data, and then passed each image in the whole dataset through the network and captured the activations in the hidden layer, giving us a length 64 feature vector. These features were then used to train a multi layer perceptron model using the same train-test split as the autoencoder. Unfortunately, the classification accuracy for this method was poor, likely due to the relatively small size of the Shenzhen dataset. Similar results were found from using a hidden layer with 128, 256, and 512 nodes.

To factors likely contributed to the poor performance of the autoencoder-classifier system. First, by downsampling the images to 250 by 250 pixels, some granularity in the image is lost. There may be some manifestation of TB, such as pulmonary nodules, that are lost in the smaller resolution image. Second, even with the downsampled image, the feature vector is still very large, consisting of 62,500 data points, which increases the number of samples required to train the classifier. In this case, as the Shenzhen dataset consists of only 662 samples, it is likely that the size of the dataset was insufficient in order to train the autoencoder. As such, an an autoencoder may be effective as a feature extractor for a larger TB dataset. Similarly to Convolutional Neural Networks, transfer learning may be used to improve the performance of an autoencoder; however, I did not perform experiments using transfer learning.

Principal Component Analysis

In addition to feature extraction methods, I evaluated using Principal Component Analysis for dimensionality reduction, both on a flattened representation of the original CXR image, and on PHOG and LBP feature descriptors. In order to avoid introducing any test data into the training dataset, I first split the data into train and test sets, then fit the PCA model to the training set only. Then, the same transformation is applied to both the train and test datasets. While I experimented with various levels of reduction, I found that classifier performance decreased significantly with less than 500 dimensions.

Combined Feature Descriptors

Based on previous research, I expect that by combining feature sets, I would achieve higher classifier performance than with a single feature vector. Particularly because TB can manifest in multiple ways on a CXR image, I expect that a combination of feature descriptors would be more invariant to different manifestations of TB. I experimented with two methods for combining PHOG and LBP features: concatenating the two feature vectors into a single feature vector with 1955 dimensions, which I will refer to as PHOG + LBP I, and by first computing the LBP mask across the image, and then computing PHOG features, resulting in 1700 dimensions, which I will refer to as PHOG + LBP II. For each method, I also evaluated the performance of PCA reduction on the final feature vector by reducing the feature vector to 500 dimensions.

Results by Dataset

As the CheXpert dataset does not have segmentation, I experimented with PHOG features only on the CheXpert dataset. In order to use binary classification on the CheXpert dataset, the data was re-labeled to 2 classes, with ”No finding” as the negative class and any other observations as positive. Additionally, to preserve compatibility with other datasets, only frontal images were considered. This resulted in a dataset of 17,075 positive samples and 174,155 negative samples. I extracted PHOG features using 3 pyramids, resulting in a feature vector of length 1700, and the dataset was split into 80% training data and 20% testing data. Additionally, because of the unbalanced nature of the dataset, I experimented with oversampling the negative samples so that the dataset contained an equal number of positive and negative samples. However, this did not provide a significant change in the classification accuracy.

For all experiments with the Shenzhen dataset, I split the dataset into 80% of samples for training, and 20% for testing. Both ACC and AUC are taken as an average over 5 iterations, with the train-test split randomized at each iteration. A summary of classifier performance for different feature sets with the Shenzhen dataset is given in Table 1. Overall, the MLP model achieved significantly higher classification accuracy than either the RF or GP models, with the PHOG + LBP I + PCA feature set achieving the highest ACC of 92% and AUC of 0.96.

TABLE 1: Classification results of different feature descriptors and classifier models on the Shenzhen dataset. The feature set with the highest accuracy is given in bold.

Feature Descriptor	Classifier Model	Vector length	ACC	AUC
PHOG	MLP	1700	90%	0.94
PHOG	RF	1700	81%	0.81
PHOG	GP	1700	82%	0.82
PHOG	MLP	210	87%	0.88
LBP	MLP	255	79%	0.85
LBP	RF	255	86%	0.87
LBP	GP	255	86%	0.85
PHOG + LBP I	MLP	1955	86%	0.90
PHOG + LBP II	MLP	1700	76%	0.86
Flattened image + PCA	MLP	500	72%	0.76
PHOG + PCA	MLP	500	89%	0.92
PHOG + LBP I + PCA	**MLP**	**500**	**92%**	**0.96**
PHOG + LBP II + PCA	MLP	500	84%	0.92
Autoencoder Network	MLP	64	54%	0.58
Autoencoder Network	MLP	128	68%	0.61

On the Montgomery dataset, the data was split into 80% training and 20% testing samples, resulting in 110 samples reserved for training and 28 samples reserved for testing. As with the Shenzhen dataset, the segmented images were pre-processed before feature extraction was performed, and the train-test split was randomized at each iteration. As I did not achieve significant classifier performance with Autoencoder networks or SURF features, I did not perform experiments on the Montgomery dataset with these features. Similarly, I performed all experiments on the Montgomery dataset using the same MLP model as I used for my experiments with the CheXpert and Shenzhen datasets, as the performance of this classifier model was significantly higher than my experiments with RF or GP classifiers. Finally, as the Montgomery dataset is both similar in quality to the Shenzhen dataset, and very small, approximately one-fifth the size of the Shenzhen dataset, it is expected to achieve a lower ACC and AUC using the same feature descriptors. An overview of my classification experiments with the Montgomery dataset is given in Table 2. In order to apply the same PCA transformation to the Montgomery dataset as the Shenzhen dataset, I used oversampling to augment the number of samples in the Montgomery dataset.

TABLE 2: Classification results of different feature descriptors using an MLP classifier on the Montgomery dataset. The feature set with the highest accuracy is given in bold.

Feature Descriptor	Vector length	ACC	AUC
PHOG	1700	72%	0.80
PHOG	210	62%	0.67
LBP	255	67%	0.64
PHOG + LBP I	**1955**	**78%**	**0.82**
PHOG + LBP II	1700	65%	0.76
PHOG + LBP I + PCA	500	67%	0.68
PHOG + LBP II + PCA	500	67%	0.70
PHOG + PCA	500	75%	0.84

Ensemble Classifiers

To evaluate the performance of ensemble classifiers for TB detection, I performed experiments using MLP, RF, and GP classifier models, as these models achieved the highest performance with PHOG features on the Shenzhen dataset. Table 3 shows the Jaccard index calculated for ensembles of two and three classifiers. As I did not achieve a high Jaccard Index on any combination of classifiers, I concluded that the use of ensemble classifiers was not a more effective solution for TB classification than using a single classifier. As such, I did not continue my experiments with ensemble classifier using other datasets or feature descriptors.

TABLE 3: Jaccard index for an ensemble of three classifiers.

Classifiers	Jaccard Index
MLP + RF	0.5161
MLP + GP	0.2692
RF + GP	0.5152
MLP + RF + GP	0.3637

Convolutional Neural Networks

I experimented with Convolutional Neural Networks (CNNs) using both the CheXpert and Shenzhen datasets. As CNNs require very large training datasets, I did not perform experiments with the Montgomery dataset. The CNN architecture was a modified version of the AlexNet network, which is a commonly used architecture and has been used successfully for a wide variety of image recognition tasks. In 2012, AlexNet was used to win the ImageNet Large Scale Visual Recognition Challenge, and contributed to a significant increase in interest in deep neural networks [39]. The AlexNet architecture contains 5 convolutional layers, 3 max pooling layers, and three fully connected connected layers with dropout. The final dense layer uses the Softmax activation function, with the ReLu activation function used for all other hidden layers. An overview of the AlexNet architecture is shown in Figure 13. For performance reasons, I downsampled the CXR images to the standard AlexNet input size of 224 x 224 pixels. My experiments with CNNs were performed using an Intel i7-3770 processor with 16 gigabytes of memory and an NVidia Titan GPU, using an implementation of the Keras machine learning library for Python 3.6 optimized for GPUs on Windows 10.

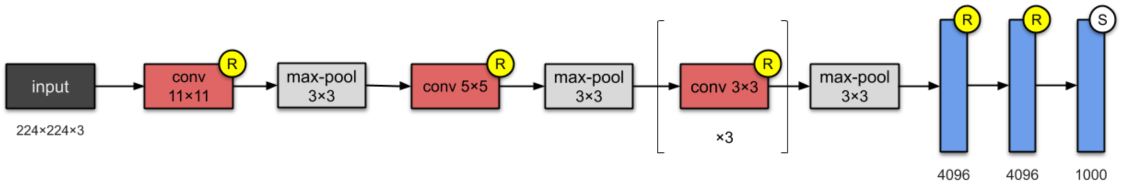


FIGURE 13: The AlexNet Convolutional Neural Network Architecture [40].

For both the CheXpert and Shenzhen datasets, the CNN classifier performance was very poor. In the case of the Shenzhen dataset, it is likely that the relative small size of the dataset was insufficient to fully train the network. While the CheXpert dataset is significantly larger, the lack of segmentation increases the amount of noise in the dataset, and does not encode lung contour information, which can be an indicator of TB. Variations in the number of convolutional layers, kernel size, dropout, and nodes in the hidden layer did not have an effect on classifier performance. As my experiments with traditional neural-network based classifiers showed significantly higher classification performance, I chose to not move forward with CNNs for my research. However, other researchers, including Hwang et al. [15] and Pasa et al. [16], have reported successful results using similar CNN architectures.

Analysis of Results

Overall, I obtained the highest classifier performance using an MLP classifier with a combination of PHOG, and LBP, and PCA features, obtaining an ACC of 92% on the Shenzhen dataset, and an AUC of 0.96. However, the size of the feature vector is still relatively large at 500 features. On the Montgomery dataset, the combination of PHOG and LBP features without any dimensionality reduction achieved the highest ACC of 78% and AUC of 0.82, with applying PCA resulting in a reduction of AUC by 0.20. This decrease is somewhat expected, as the amount by which I am reducing the dimensionality of the dataset, from 1700 to 500 features, is significantly larger than the size of the training dataset at 110 samples. While I used oversampling to attempt to augment the training data, the variance in the training data remained very low, which limited the performance of the classifier.

TABLE 4: Comparison of results with other studies on the Shenzhen and Montgomery datasets.

			Shenzhen		Montgomery	
Study	Descriptor	Classifier Model	ACC	AUC	ACC	AUC
Lopes et al. [18]	CNN	SVM	85%	0.93	83%	0.92
Pasa et al. [16]	None	CNN		0.90		0.81
Hwang et al. [15]	None	CNN	83%		67%	
Vajda et al. [8]	Shape, edge, and texture	MLP	96%	0.99	78%	0.87
My Model	PHOG + LBP	MLP	92%	0.96	78%	0.82

Table 4 shows a comparison of my model against other recent studies on the Shenzhen and Montgomery datasets. We see that my model obtained higher classifier performance than each model using convolutional networks for either feature extraction or classification. Likely due to the small size of the training datasets, the performance of single classifier models with image feature descriptors far out-performed my experiments with both Convolutional Neural Networks and Ensemble Classifiers. Similarly, texture-based descriptors such as PHOG and LBP were significantly more effective than transformation-based descriptors using PCA and autoencoder networks.

CHAPTER VI

CONCLUSION

In conclusion, the difficulty of Tuberculosis detection in chest radiographs is the different manifestations of the infection, combined with variance in image quality, clarity, and orientation. As accurate interpretation of CXR images requires skilled radiologists, the development of Computer-Aided Detection systems for automatic diagnosis of TB in CXR images is promising for reducing the spread of the disease, particularly in developing countries, which are disproportionately affected by TB.

In this thesis, I have examined methods for building a CAD system for the automatic diagnosis of TB from CXR images, including methods for lung region segmentation, image feature extraction, and classification. I have evaluated my results on three publicly available CXR datasets. My experimental results showed a combination of shape and texture features using Pyramid Histogram of Oriented Gradients and Local Binary Patterns, and dimensionality reduction through Principal Component Analysis provided the highest classifier performance. Using a Multi-Layer Perceptron classifier model, I achieved a highest ACC of 92% and an AUC of 0.96 on the Shenzhen dataset, and an ACC of 78% and an AUC of 0.82 on the Montgomery dataset.

Recently, the use of convolutional neural networks for CXR classification has achieved promising results for CXR classification; however, these models require large amounts of training data, and the lack of publicly available large segmented CXR datasets limits the effectiveness of CNN models for TB diagnosis. For smaller datasets, the use of texture and shape descriptors provides higher classifier performance while minimizing the dimensionality the training data. My proposed CAD system outperforms recent CNN-based methods, including models by Hwang et al. [15] and Pasa et al. [16], and achieves comparable performance to current state-of-the-art feature-based models, including work by Vajda et al. [8] and Karargyris et al. [41].

REFERENCES CITED

[1] W. H. Organization, “Global tuberculosis report.” `https://apps.who.int/iris/bitstream/handle/10665/329368/9789241565714-eng.pdf`. Accessed: 2020-02-28.

[2] 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.

[3] C. Qin, D. Yao, Y. Shi, and Z. Song, “Computer-aided detection in chest radiography based on artificial intelligence: a survey,” *Biomedical Engineering Online*, vol. 17, no. 1, p. 113, 2018.

[4] B. van Ginneken, L. Hogeweg, and M. Prokop, “Computer-aided diagnosis in chest radiography: Beyond nodules,” *European Journal of Radiology*, vol. 72, no. 2, pp. 226–230, 2009.

[5] S. Jaeger, S. Candemir, S. Antani, Y.-X. Wáng, 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, pp. 475–7, 12 2014.

[6] L. G. Quekel, A. G. Kessels, R. Goei, and J. M. van Engelshoven, “Miss rate of lung cancer on the chest radiograph in clinical practice,” *Chest*, vol. 115, no. 3, pp. 720–724, 1999.

[7] L. Hogeweg, C. I. Sánchez, P. Maduskar, R. H. H. M. Philipsen, A. Story, R. Dawson, G. Theron, K. Dheda, L. Peters-Bax, and B. van Ginneken, “Automatic detection of tuberculosis in chest radiographs using a combination of textural, focal, and shape abnormality analysis,” *IEEE Transactions on Medical Imaging*, vol. 34, no. 12, pp. 2429–2442, 2015.

[8] 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,” *Journal of Medical Systems*, vol. 42, no. 8, pp. 146:1–146:11, 2018.

[9] A. Zakirov, R. Kuleev, A. Timoshenko, and A. Vladimirov, “Advanced approaches to computer-aided detection of thoracic diseases on chest x-rays,” *Applied Mathematical Sciences*, vol. 9, no. 88, pp. 4361–4369, 2015.

[10] K. Murphy, S. S. Habib, S. M. A. Zaidi, S. Khowaja, A. Khan, J. Melendez, E. T. Scholten, F. Amad, S. Schalekamp, M. Verhagen, *et al.*, “Computer aided detection of tuberculosis on chest radiographs: An evaluation of the cad4tb v6 system,” *Scientific Reports*, vol. 10, no. 1, pp. 1–11, 2020.

[11] E. Soleymanpour, H. R. Pourreza, *et al.*, “Fully automatic lung segmentation and rib suppression methods to improve nodule detection in chest radiographs,” *Journal of Medical Signals and Sensors*, vol. 1, no. 3, p. 191, 2011.

[12] D. W. De Boo, M. Prokop, M. Uffmann, B. van Ginneken, and C. M. Schaefer-Prokop, “Computer-aided detection (cad) of lung nodules and small tumours on chest radiographs,” *European Journal of Radiology*, vol. 72, no. 2, pp. 218–225, 2009.

[13] Z. Xue, D. You, S. Candemir, S. Jaeger, S. Antani, L. R. Long, and G. R. Thoma, “Chest x-ray image view classification,” in *2015 IEEE 28th International Symposium on Computer-Based Medical Systems*, pp. 66–71, IEEE, 2015.

[14] J. M. Carrillo de Gea, G. García-Mateos, J. Fernández-Alemán, and J. Hernández, “A computer-aided detection system for digital chest radiographs,” *Journal of Healthcare Engineering*, vol. 2016, pp. 1–9, 05 2016.

[15] S. Hwang, H.-E. Kim, J. Jeong, and H.-J. Kim, “A novel approach for tuberculosis screening based on deep convolutional neural networks,” in *Medical Imaging 2016: Computer-Aided Diagnosis*, vol. 9785, p. 97852W, International Society for Optics and Photonics, 2016.

[16] 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, no. 1, pp. 1–9, 2019.

[17] I. Allaouzi and M. B. Ahmed, “A novel approach for multi-label chest x-ray classification of common thorax diseases,” *IEEE Access*, vol. 7, pp. 64279–64288, 2019.

[18] U. Lopes and J. F. Valiati, “Pre-trained convolutional neural networks as feature extractors for tuberculosis detection,” *Computers in Biology and Medicine*, vol. 89, pp. 135–143, 2017.

[19] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of Machine Learning Research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[20] R. S. Choras, “Image feature extraction techniques and their applications for cbir and biometrics systems,” *International Journal of Biology and Biomedical Engineering*, vol. 1, no. 1, pp. 6–16, 2007.

[21] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, pp. 886–893 vol. 1, June 2005.

[22] A. Bosch, A. Zisserman, and X. Munoz, “Representing shape with a spatial pyramid kernel,” in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR ’07, (New York, NY, USA), p. 401–408, Association for Computing Machinery, 2007.

[23] A. Chauhan, D. Chauhan, and C. Rout, “Role of gist and phog features in computer-aided diagnosis of tuberculosis without segmentation,” *PloS one*, vol. 9, no. 11, 2014.

[24] R. Pillay, “Iipimage documentation.” `https://iipimage.sourceforge.io/documentation/images/`. Accessed: 2020-04-02.

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” tech. rep., California University San Diego La Jolla Institute for Cognitive Science, 1985.

[26] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[27] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *European Conference on Computer Vision*, pp. 404–417, Springer, 2006.

[28] R. Grycuk, “Novel visual object descriptor using surf and clustering algorithms,” *Journal of Applied Mathematics and Computational Mechanics*, vol. 15, no. 3, pp. 37–46, 2016.

[29] F. H. O. Alfadhli, A. A. Mand, M. S. Sayeed, K. S. Sim, and M. Al-Shabi, “Classification of tuberculosis with surf spatial pyramid features,” in *2017 International Conference on Robotics, Automation and Sciences (ICORAS)*, pp. 1–5, IEEE, 2017.

[30] I. T. Jolliffe and J. Cadima, “Principal component analysis: a review and recent developments,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.

[31] T. Fawcett, “An introduction to roc analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[32] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, “Text classification algorithms: A survey,” *Information*, vol. 10, no. 4, p. 150, 2019.

[33] L. Nanni, S. Brahnam, and A. Lumini, “A combination of methods for building ensembles of classifiers,” in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, p. 1, The Steering Committee of The World Congress in Computer Science, 2012.

[34] F. Ye, "Basic ensemble learning." https://www.kaggle.com/fengdanye/machine-learning-6-basic-ensemble-learning. Accessed: 2020-05-22.

[35] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, IEEE, 2015.

[36] P. Lakhani and B. Sundaram, "Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks," *Radiology*, vol. 284, no. 2, pp. 574–582, 2017.

[37] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. L. Ball, K. S. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," *CoRR*, vol. abs/1901.07031, 2019.

[38] S. Candemir, S. Jaeger, K. Palaniappan, J. P. Musco, R. K. Singh, Z. Xue, A. Karargyris, S. Antani, G. Thoma, and C. J. McDonald, "Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration," *IEEE Transactions on Medical Imaging*, vol. 33, no. 2, pp. 577–590, 2013.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[40] R. Karim, "Illustrated: 10 cnn architectures." https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d. Accessed: 2020-04-26.

[41] A. Karargyris, J. Siegelman, D. Tzortzis, S. Jaeger, S. Candemir, Z. Xue, K. Santosh, S. Vajda, S. Antani, L. Folio, *et al.*, "Combination of texture and shape features to detect pulmonary abnormalities in digital chest x-rays," *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, no. 1, pp. 99–106, 2016.

APPENDIX

DATASETS

Shenzhen

The Shenzhen dataset consists of 662 frontal chest images from the Shenzhen No.3 People's Hospital, Guangdong Medical College, Shenzhen, China. Most of the images were taken over the course of September 2012. Of the 662 images, 326 are healthy samples while the remaining 336 are images of patients with tuberculosis. Segmentation was provided for this dataset using the atlas-based method by Jaeger et al., making the dataset suitable for classification because the noise from non-lung regions have been removed [5]. Additionally, the resolution of the dataset is quite large, with the images an average of approximately 2500 by 2500 pixels.

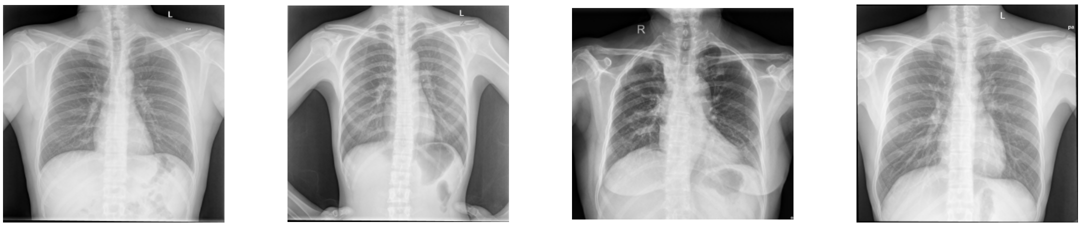


FIGURE 14: A selection of unprocessed images from the Shenzhen dataset.

Montgomery

The Montgomery (MC) dataset consists of 138 frontal chest X-ray images collected from the Montgomery County Tuberculosis screening program in Maryland, USA. The dataset contains 80 normal (healthy) samples and 58 samples from patients with tuberculosis [5]. As with the Shenzhen dataset, the images are relatively high resolution, and segmentation was provided using the atlas-based method. However, because the dataset is extremely small, and because of the relatively unbalanced nature of the dataset, the performance of my experiments with dataset was generally lower than experiments with the Shenzhen dataset.

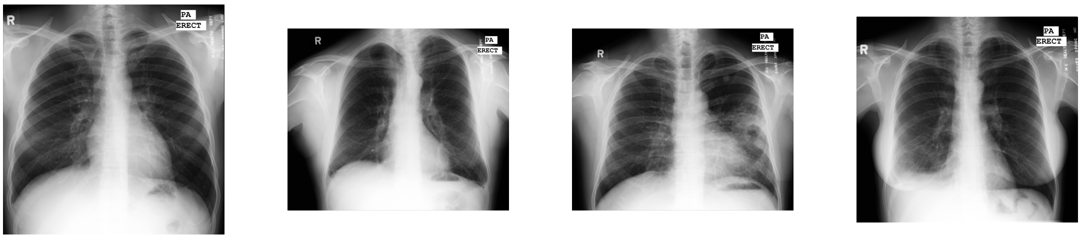


FIGURE 15: A selection of unprocessed images from the Montgomery dataset.

CheXpert

The CheXpert dataset consists of 223,648 chest radiographs taken from 65,240 patients from Stanford hospital between October 2002 and July 2017. Of the 223,648 samples, 191,229 are frontal images and 32,419 are lateral images. The dataset considered 14 observations: No finding (indicating the lack of any other observations), Enlarged Cardiomegaly, Cardiomegaly, Support Devices, Fracture, Lung Opacity, Edema, Consolidation, Pleural Other, Pleural Effusion, Pneumonothorax, Atelectasis, Lung Lesion, and Pneumonia [37]. Labels for the dataset were extracted from radiology reports using a rule-based data mining algorithm developed by the authors of the dataset. For validation, a set of 200 samples was also provided with labels assigned by a consensus of three radiologists. Each label is assigned one of three states, either positive, negative, or uncertain. A positive ”No Finding” label indicates a negative or uncertain label for all other observations. To handle uncertain labels, the authors proposed five methods: *U-Ignore*, where uncertain labels are dropped from the dataset, *U-Zeros*, where uncertain labels are treated as negative, *U-Ones*, where uncertain labels are treated as positive, *U-SelfTrained*, which uses unsupervised learning to re-assign uncertain labels, and *U-Multiclass*, where uncertain labels are treated as their own class.

To analyze the dataset, the authors trained a Convolutional Neural Network classifier using the DenseNet121 architecture, and achieved a best AUC of 0.97 on Pleural Effusion and a worst AUC of 0.85 on Atelectasis. While the large size of the dataset provides a significant advantage for classifier training, particularly for deep learning applications, the lack of segmentation available for this dataset reduces its usefulness for the detection of Tuberculosis, as a tuberculosis infection may alter the apparent shape of the lungs on a CXR image. Additionally, because of the large number of samples in the dataset, the resolution of the images is low, with an average size of 325 by 371 pixels. The dataset also suffers from low variation, with 70% of the images representing only 31% of patients. Additionally, because the observations are automatically extracted from radiology reports, the dataset does not make a distinction between active and latent observations. For example, in the ”Fracture” class, no distinction is made between a patient with a current bone fracture and one with a fully healed fracture. Finally, there is some uncertainty in the ”No Finding” observation, as this simply represents that no observation was found in the radiology report. For these reasons, I consider the CheXpert dataset to be less applicable to the development of a CAD system for TB detection than the Shenzhen or Montgomery datasets.

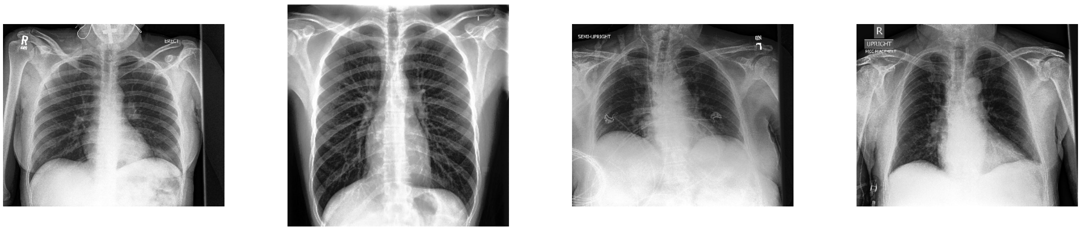


FIGURE 16: A selection of unprocessed images from the CheXpert dataset.