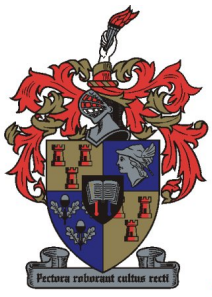
# Computer Vision and Machine Learning in Radiographic Diagnosis of lung disease

Supervised by Prof. R.T. Newman

**Tarryn Bailey 17843162**

Literature Survey




UNIVERSITEIT • STELLENBOSCH • UNIVERSITY

Physics Department
Stellenbosch University
South Africa
September 2018

# Contents

# Chapter 1

# Introduction

No matter the method of diagnosis, an early, accurate diagnosis can be the difference between life and death.

Hundreds of years ago, doctors could only diagnose patients based on symptoms or by invasive surgery, but today, over fifty percent of medical diagnoses are performed using radiological imaging modalities [1]. With the growing quantity and quality of radiological methods, radiologists have had to deal with an ever-increasing workload. Radiologists often analyze hundreds of images per week. Errors are both possible and dangerous.

In the 1950's, the first Artificial Intelligence based medical systems modelled the relationship between symptoms and diseases [2]. They used heuristic reasoning to apply knowledge to the available data and rank alternatives continuously, making the most efficient decision at each branching step to approximate the exact solution.

The initial difficulties were complex as scientists and doctors realized machines would have to master intuitive thinking. The art of diagnosis was not monochrome, the machines would have to diagnose multiple concurrent disorders, requiring deep causal reasoning. Additionally, the uncertainties had to be traced and presented clearly. Furthermore, communications between practised radiologists and computer scientists were sometimes ambiguous. It did not help that experts struggled to identify their logical systems [2].

Today, computer aided diagnosis (CAD) is a concept that requires the collaboration of computer vision and machine learning to teach computers to analyze radiological images the same way a radiologist would. The computer is then able to provide a second opinion for the radiologist or classify most images where a full time radiologist is not affordable [3].

# Chapter 2

# Modern Day Ventures

Traditional image processing systems involved a multi-step approach. Modern CAD systems use Convolutional Neural Networks (CNN's). These are end-to-end deep learning systems consisting of a single neural network. These deep learning systems require a large amount of training data to outperform the traditional systems[4]. They are highly effective for computer vision and medical image analysis. CAD systems therefore require a large data set of training images to perform a reliable classification of an input image.

Medical Imaging systems are able to control image quality. For classification systems, this is particularly useful since uniform image quality is necessary for uniform performance throughout the data set. Suspicious regions in the image can be located and selected for classification purposes using shape and texture analysis.

Building a system that can classify images and master computer vision has been a worldwide project with countless applications from facial recognition to self-driving cars. The use of CNN's in this endeavour required a large open source database for all students, scientists and engineers. This database is called ImageNet.

ImageNet consists of over 14 million individually labelled images of objects. This is useful in medical imaging research as it is often much easier to train a model to see complex features if it has been trained on previous data. These systems are therefore already capable of interpreting edges, shapes and objects in image data.

ImageNet hosts an annual competition where scientists and engineers around the world can enter their unique classifying systems. GoogLeNet, InceptioNet and ResNet are classifiers that were pre-trained on ImageNet and made commercially available because of their involvement in the ImageNet competition. For work solely focused on medical image classification, there is an open source database of thousands of frontal view chest x-rays called ChestX-ray14.

CAD4TB is a commercially available software used to detect tuberculosis [3]. It was made by Delft Imaging Systems, based in the Netherlands. CAD4TB uses textural abnormality, shape detection and feature extraction. It can be used with support vector machines to improve classification. In Africa, it is used to combat TB in impoverished countries. In South Africa it is also used in prisons [5]. The combination of low cost, quick digital x-rays, machine learning and remote medical expertise allows for a quick diagnosis and immediate treatment.

Thanks to CAD4TB, South African patient information was used in [6] to improve diagnoses using machine learning to generate a CAD score based on a combination of chest x-rays and clinical features. The model built significantly outperformed the individual approaches. This was a positive outcome for future TB screening worldwide.

Diagnostic systems created for classification of lung pathologies may use different models and analyze images for various abnormalities, but the goals are common. The goals are to detect abnormalities early [7], correctly identify the disease, highlight patients that need immediate attention,

inform practitioners when more scans are needed [8] and inform radiologists of uncertainties in the system's diagnosis[2].

The central challenge is to build a fully automated system that can analyze large quantities of images. The system must be capable of detecting and locating abnormalities and classifying chest x-ray images accurately even when biological variations are present.

# Chapter 3

# Computer Vision

It is common knowledge that images are stored on computers as matrices of pixels. Each pixel has its own RGB (red, green, blue) value which indicates the colours present in the pixel. Computers are able to traverse pixels in an image and locate specific pixels, based on their RGB values. For x-ray images we consider greyscale images, where each pixel has a greyscale value [9].

The algorithm starts at the top left pixel and moves across the row and down the columns analysing each pixel using these greyscale values. If it is looking for a pixel with a particular greyscale value it will calculate the difference between the input greyscale value and the one it is looking for. The algorithm then returns the pixels, highlighting the ones that were closest to the requested greyscale value.

If the algorithm is looking for something bigger than a pixel, it has to search for and identify groups of pixels called patches. For any computer vision system, it is important to identify edges. The algorithm is able to detect edges by detecting a change in greyscale values that persists in a particular direction. Consider a pixel on an edge in an image. The algorithm will be able to detect a significant colour difference on either side of the pixel. For vertical edges, it will find a large difference in the greyscale values of the pixels on its left and on its right [9].

Mathematically, a matrix called a kernel or filter is used. The kernel for detecting vertical edges can be as simple as a $3 \times 3$ matrix, consisting of a column of negative ones, a column of zeros and a column of positive ones [9]. Essentially, it performs multiplication on the pixel and the surrounding pixels. It then sums all nine values, calculating the difference between pixels to the left and to the right of the pixel.

Finally, it replaces the pixel's greyscale value with a new value indicating the contrast or difference between the pixels to its left and to its right. After this replacement of pixels (or filtering) the highest pixel values indicate the location of an edge in the image [9].

This process of applying a kernel to each pixel in a group of pixels is called a convolution [9] For the detection of horizontal edges, a different kernel is used. Kernels have many uses in computer vision. They are not only able to detect edges in different directions, but they are also used to sharpen images, to blur them, to detect islands of contrast[9], to recognise patterns and assign them to objects, such as noses or eyes.

Convolutional neural networks (CNN's) can learn their own kernels and recognise feature in images. CNN's are able to process image data by using groups of neurons. A single input image can be passed through multiple neurons which each detect a unique patch that corresponds to a pattern or texture found in the image. The neurons then output stacks of images that are called filters. This is called a convolutional layer [9]. The outputs can then be processed by another set of neurons or another layer.

Convolutional neural networks can consist of many layers. For example, the first layer could detect edges, while the second layer detects shapes and the third detects objects and so on. The

final layer is called the fully connected layer and it is usually the one to classify images. CNNs may be many layers deep to classify complex images[9]. Deep learning is the term used when deep convolutional neural networks are used in machine learning. This is useful for analysing images for patterns and for considering the ways in which these patterns occur with respect to each other [9]. When used on images of objects on and inside the human body, deep convolutional neural networks are able to recognise biometric data. This is used in facial recognition.

Classification of an image requires four layers. These layers are the convolutional layer, the ReLU layer, the pooling layer and finally the fully connected layer [10]. For example, consider a convolutional neural network that classifies images as X's or O's. The classification images would be images of a perfect X and a perfect O. These are the image that the network initially learns on. During training, the network will compare any input image to this classification image. The network will first apply each layer to the classification images and output classification image results. For any input image it will also apply each layer to the input image and compare its output to the output of the classification images.

The convolutional layer compares patterns mathematically. In this layer one image becomes a stack of filtered images. The pixels are assigned new values indicating to what extent the pixel and its surrounding pixels is similar to the pattern that the algorithm is searching for. Each filtered image is simply an array of values, specifying whether and where the patterns occur. This process is done for the classification images.

The Rectified Liner Units (ReLU) layer takes the filtered images as input and changes any negative values in the pixels to zeros but leaves the positive values unchanged [9]. This is to correct for any mathematical errors that could occur later due to the summing of negative values.

The pooling layer shrinks the filtered image by replacing windows of pixels by a single pixel with the highest value that occurred in the window. It is essentially a method of compressing the filtered image without losing too much information. The window size and slide size should be chosen according to the extent of compression desired.

The convolutional, the ReLU and the pooling layers can be stacked up to filter and compress images until they have been reduced to a size that is small enough to be analyzed easily [11].

The final classification takes place at the fully connected layer. This layer organises all the values into a single list. Each classification image has a unique list. Each list has high values at particular positions which is unique to the classification [10]. The list created for any input image is compared to the list of each classification image. The similarity between the list of the input image and the lists of the classification images is calculated based on the positions of the highest values in the lists. The input image list is then assigned values indicating the similarity to each classification image's list. The final classification is made based on the highest similarity.

Analysts are able to decide whether the neural network performed the correct classification. If they find that it was incorrect, they calculate the error in the prediction. Subsequently, they will carry out a weight adjustment to decrease errors in the future [11].

There are other techniques for analysts to enhance the performance of a neural network. These are machine learning techniques which will be discussed later.

# Chapter 4

# Machine Learning and Artificial Intelligence

Artificial Intelligence is a field of study and applied research where the goal is to create a machine that exhibits human-like intelligence [12]. Artificial Intelligence does not simply involve the absorption and regurgitation of large quantities of information. The machine is expected to demonstrate the ability to learn from past experience. It should master knowledge representation and apply reasoning. Finally the machine must display both intuitive and abstract thinking.

Machine learning is only a subsection of Artificial Intelligence [13]. It is the use of algorithms that allow computers to learn by analyzing data and making statistical predictions and decisions based on data. Computers are able to learn by error correcting.

For our purposes we consider machine learning algorithms called classifiers. Classifiers can be trained using large quantities of real images or by learning to search for features in an image.

The large quantity of data used to train a classifier is called training data. The data usually consists of images, and information about what is in the images as well as a classification. The algorithm learns to classify data based on features in the images [13]. However, these features can be ambiguous at times, multiple diseases could have similar symptoms or the images could simply be unclear.

In computer science, it is said that the computer tries to make a decision boundary, which is based on features and may not necessarily be a straight line, or exist in only two dimensions. Ultimately, the algorithm must make as many correct classifications as possible and minimise its error [13]. There are seven steps in the machine learning process. They are gathering data, data preparation, choosing a model, training, evaluation, hyper-parameter tuning and prediction [14].

At the first step, gathering data, the quality and quantity of data will determine the performance of the classifier [14]. It is important to have a variety of images for the machine to learn on so that it will be prepared to classify correctly regardless of variations.

For data preparation, the data is accumulated and it undergoes certain processes such as randomization, augmentation, normalization and error correction. This is where data imbalances are managed. A collection of data with significantly more healthy than diseased images will train a classifier to be biased toward classifying an image as healthy. The data must be split into training data and evaluation data. It is important for the classifier to be evaluated on data that it was not presented with during training. The ratio of training data to test data depends on the amount of data. The larger the amount of data, the smaller the required ratio of training to test data [14].

The model chosen depends on what the classifier will be classifying such as images, sounds or numbers. It also depends on the number of features that the classifier has to be able to take into account [13]. The classifier must be able to handle the number of parameters that must be considered in order to determine whether the image should be classified as, for example healthy,

pneumonia or tuberculosis.

In the course of training the classifier improves, learning one iteration at a time. Like a radiologist learning on the job, its initial classifications may not be very accurate, but as it undergoes more iterations it learns to classify correctly, improving its performance with experience. Each attempt to improve performance is called a training step [14]. A training step can be done thousands of times to reach the desired accuracy.

Evaluation data is used during evaluation to test the classifier on data it has never seen before. Once it has classified this data, an analyst would have to evaluate the results to assess the performance of the classifier. If the classifier's performance was not up to standard, the parameters must be tuned to improve the training [14]. For example, if too many diseased lungs were classified as healthy, perhaps a larger variety of, or more, images of diseased lungs is needed during training. Other improvements include an increased number of times that the model is trained on the same data, an increased learning rate (how much the model is adjusted at each training step) or a change in the initial conditions [14]. The hyper-parameter tuning is more of an art than a science, it depends on the model's function and where it is lacking.

Finally, the model is able to predict and classify the chest x-ray images, diagnosing patients. Hereafter the accuracy, specificity, stability, F1 score and more can be calculated, reported and compared to other state of the art models or human radiologists.

The goal is to build a classifier that is as good as or better than the average human radiologist so that it can assist radiologists or perform the job of a standard radiologist in areas where a radiologist is needed but not financially attainable.

# Chapter 5

# Case Studies

## 5.1 Deep Learning for Abnormality Detection in Chest X-Ray images [8]

This study was published in 2017 and performed by students at Stanford University. The goal was to focus radiologists on high risk tuberculosis cases and correct their misdiagnoses. A neural network was developed that classified images as normal, non-normal, high risk and/or more scans needed. 50 000 labelled chest x-ray images, .tiff format, were chosen from Stanford's database.

The study compared the performance of GoogLeNet, InceptioNet and ResNet. Neural Network visualisation was used to investigate the algorithm's classification method.

Images were labelled 0, 1 or 2 meaning normal, abnormal or emergent respectively. Each image was originally $3000 \times 3000$ pixels but they were downsized to $512 \times 512$ and $224 \times 224$ by the random selection of pixels. Images were pre-processed using histogram equalization to increase contrast [15] and enhance the difference between bone, empty space and tissue. This was done using Python's scikit image library.

For data augmentation, the images were flipped 0, 90, 180 or 270 degrees and flipped left to right or not and a random small amount of Gaussian noise added to each pixel. It is of utmost importance that the noise is random, not structured, and that it is added to all the training data and not solely to one classification since the neural network could learn to look for structured noise as a symptom in its classification [16].

The CNN models ran on a server with four Nvidia Tesla P40s. These are Graphics Processing Unit accelerators used for visual computing. The training data to test data had a 90:10 ratio. The training data normal to abnormal ratio was about 65:35.

Overfitting occurs when the classifier learns the training data too well, performing well during training, but poorly during testing [17]. Overfitting was evaluated by the comparison of cross entropy loss (measures performance [18]) and accuracy on training data vs. test data.

Default parameters were used, with the learning rate set to 0.001, the regularization set to L2, with drop out set to 0,5 and the batch size set to 256. Batch size refers to the number of images the network trains on at a time. Regularization prevents overfitting by decreasing the weights, improving the model's ability to generalize [19]. Drop out is a regularization technique that forces the model to randomly skip neurons, forcing other neurons to learn what the skipped neuron learned [20]. Nodes were randomly skipped with a probability of 0,5. The parameters were previously optimized on ImageNet.

The models being compared included GoogLeNet, Inception V3 and ResNet. Both GoogLeNet and Inception V3 rely on inception modules, but GoogLeNet consists of only 22 layers while Inception V3 consists of 48 layers. Inception modules take an input of an image. Instead of the programmer choosing a filter size or whether they want to add an extra pooling layer, the in-

ception module chooses various sizes simultaneously and can add a pooling layer too. Finally it concatenates the results. It learns to search for small features and large features simultaneously. This results in higher accuracy. [21]

GoogLeNet implements a dimensionality reduction step which effectively saves computational power. Inception V3 uses factorized convolutions and aggressive regularization. ResNet relies on residual networks which allows it to contain 152 layers without encountering problems. Essentially, they skip steps in the layers, taking shortcuts to decrease error rate and maintain accuracy [20]. They are therefore easier to optimise than traditional networks. Factorized convolutions reduce 3D convolutions to 2D convolutions in convolutional layers. Essentially, it preserves spatial information and maintains accuracy with less computation.

For GoogLeNet, the results showed that the complexity and depth of the network didn't improve performance. As anticipated, using a larger dataset increased training accuracy, however it hardly affected validation accuracy, which was 0,8. The larger dataset trained the model after fewer iterations and resulted in a relatively constant validation accuracy which can be attributed to the larger batch sizes. Finally, the image resolution did not play a role in prediction accuracy.

It was found that GoogLeNet performed significantly better on 50 000 images with dimensions of $224 \times 224$. It was marginally biased to predicting normal, this was probably due to imbalances in the training data. When the networks methods were investigated, it was found that the system was basing its classification on symmetry. Therefore the conclusion was drawn that the macroscopic features were learned. To this end, it was proposed that segmentation could be used to recognize small features.

It is notable that training a deep learning network using thousands of training images can be risky since it is not always predictable what the classifier will base its classifications on. This could lead to classifications that seem to exhibit relatively good diagnoses even though the classifier is not looking for symptomatic signs. In this case, the classifier's accuracy during testing will not reflect its accuracy in real life.

## 5.2 Deep learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by using Convolutional Neural Networks [3]

The second study looks at the use of ensembles to improve performance. This was done at Thomas Jefferson University in 2017. 1007 postero-anterior chest x-rays were used and the deep convolutional neural networks compared and combined were AlexNet and GoogLeNet, pre-trained on 1.2 million colour images in 1000 categories on ImageNet. The performance of these networks when they were untrained was also evaluated in this study.

This study involved four datasets, two from the National Institute of Health from Maryland and China, one from Thomas Jefferson University in Philadelphia, consisting of only healthy cases, and one from Belarus Tuberculosis Portal, consisting of only TB cases. Positive cases were confirmed using sputum tests and radiology reports and an independent radiologist.

Images were resized to $256 \times 256$ pixels and converted to .png format. The computer used was installed with Ubuntu 14.04. The Caffe Model Zoo is an open source repository for pre-trained networks, where the Caffe deep learning framework is implemented to train the network. This is where pre-trained AlexNet and GoogLeNet was found.

The computer had CUDA 7.5/cuDNN 5.0 dependencies for GPU acceleration. It also included an Intel i5 3570k 3.4 gHz processor with 4 TB of hard disk space, 32 GB of RAM and a CUDA-enabled Nvidia Titan $\times$12 GB GPU.

The solver parameters set during training on the postero-anterior chest radiographs included 120 epochs, stochastic gradient descent, step down set to 33% and $\gamma$ set to 0,1. An epoch is one complete run through of the dataset to be learned. Neural networks that use iterative algorithms need many epochs during training [22]. Stochastic gradient descent is a method used to save computational cost due to back propagation through the entire training set [19].

The base learning rate for pre-trained models was set to 0,001 and for untrained models it was set to 0,01.

For data preparation, the augmentation included cropping the images randomly to $227 \times 227$ pixels, mean subtraction of each pixel and mirroring of images. As an additional test, the performance of AlexNet and GoogLeNet was also tested when the images were augmented further by Contrast Limited Adaptive Histogram Equalization and rotated by 90, 180 and 270 degrees. The Histogram Equalization was done using ImageJ v. 1.50i.

75 healthy and 75 TB images were randomly selected for testing from 1007 images using pseudorandom numbers from Python 2.7.13. The 75 TB images were assessed by a cardiothoracic radiologist for degree of pulmonary parenchymal involvement and they were then classified as subtle, intermediate and readily apparent.

Once these testing images were excluded, the remaining 857 images were randomly split according to a 80:20 ratio so that there were 685 training images and 172 validation images.

The results were analysed using statistical analysis performed by MedCalc v.16.8. ROC and AUC curves were determined on the results from the test dataset. An ROC curve is a Receiver Operating Characteristic curve. It is a plot of the true positive rate against the false positive rate. It shows the relationship between sensitivity and specificity [24].An AUC is the area under a ROC curve, measuring accuracy [25].

Contingency tables, accuracy, sensitivity and specificity were determined from the Youden Index. The Youden Index gives the maximum potential effectiveness for a particular biomarker to indicate the presence of a disease. It provides a summary of the information contained in the ROC curve [26]. The adjusted Wald method was used to determine 95% confidence intervals on the

accuracy, sensitivity and specificity from the contingency tables.

Ensembles were built by taking varying weighted averages of the probability (of tuberculosis) scores generated by the AlexNet and GoogLeNet. The weighting varied from equal weighting to ten-fold weighting in either direction. ROC curves, AUC, sensitivity and specificity values were determined for these ensemble approaches. Finally, if AlexNet and GoogLeNet disagreed, an independent cardiothoracic radiologist with over 18 years of experience blindly classified the images as healthy or as having TB.

For both AlexNet and GoogLeNet, the AUCs of the pre-trained models surpassed that of the untrained models. Further augmentation using Histogram equalization and rotations increased accuracy for both models over their untrained versions.

The best performing ensemble had an AUC of 0,99 which significantly surpassed the AUC of the untrained models individually, with AlexNet's 0,90 and GoogLeNet's 0,88. The Classifiers disagreed on 13 of the 150 test cases which were then analysed by the radiologist who correctly classified all 13. This is the radiologist augmented approach. There were, however, 2 false negatives which were dismissed by both systems and therefore never assessed by the radiologist.

Ensemble methods in this study significantly improved performance because they were essentially a blend of multiple algorithms that removed uncorrelated errors of individual classifiers using averaging. Weighted averages of probability scores for AlexNet and GoogLeNet were used and a 10-fold weighting toward GoogLeNet provided the best accuracy. The accuracy was then improved by the radiologist augmented approach. To make up for the false positives, a larger training set could be used as well as more augmentation methods with additional machine learning approaches.

## 5.3 CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning [27]

The third case study is quite interesting and has been a polarising topic; the use of ChestX-ray14, an open source data base with over 100 000 frontal view chest x-rays labelled with up to 14 diseases.

The CheXNet model was designed at Stanford University in 2017 for the diagnosis of pneumonia. It was trained on ChestX-ray14. It was then extended to diagnose all 14 diseases. The model's performance was compared to the performance of 4 practising academic radiologists.

CheXNet takes an X-ray image as input and outputs the probability of pneumonia with a heat map of indicative areas. Pneumonia's appearance in chest x-rays can be ambiguous, overlap with other diagnoses and resemble benign abnormalities.

CheXNet is a 121 layer CNN called a DenseNet. DenseNets improve information flow through the network. The standard parameters were set. Stochastic gradient descent was used with $\beta 1$ set to 0,9 and $\beta 2$ set to 0,999. The model was trained using mini-batches of 16 and an initial learning rate of 0,001.

All 112120 frontal view images from ChestX-ray14 were used. Each image in Chest X-ray 14 is labelled with up to 14 pathology labels. Images that were labelled with pneumonia were marked as positive and without pneumonia as negative. In total, 98637 images were used for training (93.6% of the total), 6351 images were used for validation (6%) and 420 images were used for testing (0.4%).

The images were downsized to $224 \times 224$ pixels and normalized, based on the mean and standard deviation of images in ImageNet. The training data was augmented using random horizontal flipping of images. The performance of 4 radiologists at Stanford University was also measured, by their annotations on each image. The radiologists had 4, 7, 25 and 28 years of experience, with one of them being a specialized cardiothoracic radiologist.

For the detection of pneumonia, each image in the test dataset was subsequently assigned 5 labels, 4 from the radiologists and 1 from CheXNet. The F1 score was calculated to indicate the precision for each radiologist and finally for CheXNet. It was found that CheXNet obtained a significantly higher F1 score than 3 of the radiologists. The confidence intervals were determined using a statistical technique called bootstrapping.

The average radiologist score was 0.387 (95% confidence interval of (0.330, 0.442)) the highest F1 score obtained was by the fourth radiologist with 0.442 (CI: 0.390, 0.492) and the F1 score for CheXNet was 0.435 (CI: 0.387, 0.481) which was higher than the average radiologist F1 score.

Certain limitations clearly exist, the model and the radiologists were not permitted to view patient history, which hinders performance. Additionally, the unavailability of lateral view images makes it more difficult to diagnose.

For the extension of CheXNet for detection of all 14 diseases, the algorithm was altered to include 3 changes. Firstly, instead of a binary label, CheXNet was altered to output a vector indicating the absence or presence of all 14 diseases. Secondly, the fully connected layer was substituted with a fully connected layer with a 14 dimensional output, where the final output is a probability of the presence of each of the 14 diseases. Finally the loss function was modified to optimize cross entropy losses.

This time the dataset was indiscriminately split into training (70%), validation (10%) and test (20%). It was found that CheXNet obtained state of the art results on the 14 classes. It was important to understand exactly what the model was basing its classifications on, therefore feature maps were extracted from the final convolutional layer to clearly visualize the areas that the classifier considered to be indicative of the assigned pathologies.

# Chapter 6

# ChestX-ray14

As mentioned earlier, the use of ChestX-ray14 has been a polarising topic. According to [16] ChestX-ray14 is not a suitable dataset for training.

The radiologist who made this statement elaborated, saying various images in the dataset did not clearly show signs or symptoms of the disease/s with which they were labelled. In [27] it is stated that the patients were diagnosed by alternative methods, specifically, these labels were automatically extracted from radiology reports. It is possible that a radiologist did not go through each image to ensure that the pathology is visible. That would explain how these images remained in the system even though the symptoms are not visible.

Additionally some of the images for curable diseases show cured lungs even though the images are still in the dataset for the particular pathology. In the dataset for a pneumothorax image, certain images contained a clear indication that the patient had received a chest drain. This could teach the system to consider a chest drain a main symptom of a pneumothorax and to only diagnose a pneumothorax if a chest drain is present. This is pointless since that patient has already been diagnosed, and it is the more subtle symptoms that need to be focused on.

Furthermore some of the diseases such as pneumonia and emphysema are mostly diagnosed clinically, not solely by imaging since it is not always visually clear using x-rays.

Despite these issues, the case study reviewed regarding CheXNet reported impressive results. The argument made by [16] would explain why three of the radiologists' F1 scores were significantly lower than CheXNet's. The symptoms in the images may not have been clear or even present, had the images been of patients who had already received treatment.

It is speculated that CheXNet performed well during testing, because the problems in the training set were likely still present in the test set. CheXNet could have occasionally learned to look for irrelevant features that were common in the training and the test data. These additional features may not have been symptomatic, thus making them irrelevant to the radiologists. This could lead to misdiagnoses even though the AUC values indicate that the model classified images according to their labels.

Once again, these labels were automatically extracted from radiology reports. Using automatic methods to extract text could lead to structured noise in images [16]. This could teach a classifier to see patterns that aren't symptomatic, they just predict according to unrelated coincidences in the dataset.

These problems could be minimised with the help of an experienced radiologist willing to analyse a few thousand images from ChestX-ray14. They could select images with visually present pathologies and an output could be created for images that don't have any clear pathologies. This way images from ChestX-ray14 could still be used to train a model, without the issue of the irrelevant features.

# Bibliography


[1] Fullstack Academy. Introduction to Computer-Aided Diagnosis in Medical Imaging (Radiology) [Online] 2016 [access 2018, April 30]; Avaliable: https://www.youtube.com/watch?v=46Oz5JNFR24

[2] Chabat F, Hansell DM, Yang GZ. Computerized decision support in medical imaging. IEEE Engineering in Medicine and Biology Magazine 2000;19(5):89-96.

[3] Lakhani P, Sundaram B. Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks. Radiology. 2017;284(2):574-582.

[4] Deep learning.ai. What is end-to-end deep learning? (C3W2L09). [online video] 2017 [access 2018, April 12]; Available: https://www.youtube.com/watch?v=ImUoubi_t7s

[5] Computer-Aided Detection for Tuberculosis (CAD4TB) [Online][s.a][access 2018, September 1]; Available: https://www.delft.care/cad4tb/

[6] Melendez J, Sánchez CI, Philipsen RH, Maduskar P, Dawson R, Theron G, Dheda K, Van Ginneken B. An automated tuberculosis screening strategy combining X-ray-based computer-aided detection and clinical information. Scientific reports 2016;6:25265

[7] Wozniak M, Pola D, Kosmider L, Napoli C, Tramontana E. A novel approach toward X-ray images classifier in Computational Intelligence. IEEE Symposium Series 2015;1635-1641.

[8] Tataru C, Yi D, Shenoyas A, Ma A. Deep Learning for abnormality detection in Chest X-Ray images [Online] 2017 [access 2018, April 25]; Available: http://cs231n.stanford.edu/reports/2017/pdfs/527.pdf

[9] CrashCourse. Computer Vision: Crash Course Computer Science #35. [Online video] 2017 [access 2018, July 29] Available: https://www.youtube.com/watch?v=-4E2-0sxVUM&t=21s

[10] Edureka! Convolutional Neural Network (CNN) — Convolutional Neural Networks with TensorFlow —Edureka. [Online video] 2017 [access 2018, April 12]; Available: https://www.youtube.com/watch?v=umGJ30-15_A

[11] Rohrer B. How Convolutional Neural Networks work. [Online video] 2016 [access 2018, April 13]; Available: https://www.youtube.com/watch?v=FmpDIaiMIeA

[12] Android Authority. What is machine learning? [online video] 2015 [access 2018, July 30]; Available: https://www.youtube.com/watch?v=WXHM_i-fgGo

[13] CrashCourse. Machine Learning and Artificial Intelligence: Crash Course Computer Science #34. [online video] 2017 [access 2018, July 30]; Available: https://www.youtube.com/watch?v=z-EtmaFJieY&t=1s

[14] Google Cloud Platform. The 7 Steps of Machine Learning. [online video] 2017 [2018, July 30]; Available: https://www.youtube.com/watch?v=nKW8Ndu7Mjw&t=14s

[15] Histogram Equalization [Online][s.a][access 2018, August 1]; Available: https://www.tutorialspoint.com/dip/histogram_equalization.htm

[16] Oakden-Rayner L. Exploring the ChestX-ray14 dataset: problems [Online] 2017 [access 2018, August 25]; Avaliable: https://lukeoakdenrayner.wordpress.com/2017/12/18/the-chestxray14-dataset-problems/

[17] Overfitting in Machine Learning: What It Is and How to Prevent It [Online][s.a.][access 2018, August 1]; Available:https://elitedatascience.com/overfitting-in-machine-learning

[18] Loss Functions [Online][s.a.][access 2018, August 1]; Available:http://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html

[19] Understanding regularization for image classification and machine learning [Online][s.a.][access 2018, August 1]; Available: https://www.pyimagesearch.com/2016/09/19/understanding-regularization-for-image-classification-and-machine-learning/

[20] Brownlee J. Dropout Regularization in Deep Learning Models with Keras [Online] 2016 [access 2018, August 1]; Available: https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/

[21] Mathew J. Inception of the Deep Learning World! [Online] 2017 [access 2018 August 1]; Available: https://becominghuman.ai/inception-of-the-deep-learning-world-b1b6889cbdc1

[22] Epoch [Online][s.a.][access 2018, August 12]; Available: http://www.fon.hum.uva.nl/praat/manual/epoch.html

[23] Optimization: Stochastic Gradient Descent [Online][s.a.][acccess 2018, August 12]; Available: http://ufldl.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent/

[24] Receiver Operating Characteristic (ROC) Curve: Definition, Example, statisticshowto [Online][s.a.][access 2018, August 12]; Available: http://www.statisticshowto.com/receiver-operating-characteristic-roc-curve/

[25] Area under curve (AUC) [Online][s.a.][access 2018, August 12]; Available: https://analyse-it.com/docs/user-guide/diagnosticperformance/auc

[26] Ruopp MD, Perkins NJ, Whitcomb BW, Schisterman ER. Youden Index and Optimal Cut-Point Estimated from Observations Affected by Lower Limit of Detection. Wiley Online Library 2008;50(3):327-452

[27] Rajpurkar P, Irvin J, Zhu K, Yang B, Mehta H, Duan T, Ding D, Bagul A, Langlotz C, Shpanskaya K, Lungren MP. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv 2017:1711.05225.