Classification of Lung Diseases Using Deep Learning Models

Matthew Zak

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Computer Science at

Concordia University

Montréal, Québec, Canada

September 2019

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Matthew Zak**

Entitled: **Classification of Lung Diseases Using Deep Learning Models**

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Andrew Delong

_____ External Examiner
Dr. Thomas Fevens

_____ Examiner
Dr. Ching Yee Suen

_____ Supervisor
Dr. Adam Krzyzak

Approved by _____
Lata Narayanan, Chair
Department of Computer Science and Software Engineering

_____ 2019

_____
Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Master of Computer Science

Classification of lung diseases using deep learning models

by Matthew ZAK

Although deep learning-based models show high performance in the medical field, they required large volumes of data which is problematic due to protection of patient privacy and lack of publically available medical databases.

In this thesis, we address the problem of medical data scarcity by considering the task of pulmonary disease detection in chest X-Ray images using small volume datasets ($<10^3$ samples). We implement three deep convolution neural networks pre-trained on the ImageNet dataset (VGG16, ResNet-50, and InveptionV3) and asses them in the lung disease classification tasks transfer learning approach. We created a pipeline that applied segmentation on Chest X-Ray images before classifying them and we compared the performance of our framework with the existing one. We demonstrated that pre-trained models and simple classifiers such as shallow neural networks can compete with the complex systems.

We also implemented activation maps for our system. The analysis of class activation maps shows that not only does the segmentation improve results in terms of accuracy but also focuses models on medically relevant areas of lungs.

We validated our techniques on the publicly available Shenzhen and Montgomery datasets and compared them to the currently available solutions. Our method was able to reach the same level of accuracy as the best performing models trained on the Montgomery dataset however, the advantage of our approach is a smaller number of trainable parameters. What is more, our InceptionV3 based model almost tied with the best performing solution on the Shenzhen dataset but as previously, it is computationally less expensive.

"Everything has been, everything has happened. And everything has already been written about."

Vysogota de Corvo

"There is nothing noble in being superior to your fellow man; true nobility is being superior to your former self."

Ernest Hemingway

Acknowledgements

I want to thank everyone who has had the tiniest input in this thesis. If it had not been for you, I would probably have never made it.

Contents

Declaration of Authorship ii

Abstract iii

Acknowledgements v

1 Introduction 1

1.1 Project objectives 1

1.2 Motivation 2

1.3 Applications 2

1.4 Contributions 2

1.5 Thesis structure 3

2 Related Work 4

2.1 Pixel/Voxel-Based Machine Learning 4

2.1.1 Overview 5

2.1.2 Bone Separation from Soft Tissue in Chest Radiographs (CXRs) by Use of MTANNs 6

2.2 Approaches to lung disease classification 8

2.2.1 Extreme learning machine 8

2.2.2 Automatic CXR screening system 10

2.2.3 Semi-Supervised Learning 12

2.3 Automatic diagnostic using Deep Learning in MODS digital images analysis 15

2.4 Deep Learning Approaches in Chest X-Ray analysis 17

2.5 Dataset . . . 17
2.6 Image Data Augmentation . . . 20
2.7 Evaluation . . . 22
2.8 Convolutional Neural Networks . . . 26
2.9 Summary . . . 29

3 Transfer Learning in Lung Diseases Classification 30
3.1 Transfer learning . . . 30
3.1.1 Pre-trained models approach . . . 31
3.1.2 ImageNet . . . 31
3.2 VGG16 . . . 32
3.2.1 VGG16 Architecture . . . 32
3.3 ResNet-50 . . . 35
3.3.1 Residual Bloc . . . 35
3.3.2 ResNet50 Architecture . . . 36
3.4 Inception . . . 39
3.4.1 Factorizing Convolutions . . . 40
3.4.2 Factorization Into Asymmetric Convolutions . . . 40
3.4.3 Auxiliary Classifiers . . . 42
3.4.4 Effective Grid Size Reduction . . . 43
3.4.5 Architecture . . . 44
3.5 Experiments . . . 46
3.5.1 Dataset . . . 46
3.5.2 Models . . . 46
3.6 Results and analysis . . . 47
3.6.1 VGG results . . . 48
3.6.2 ResNet results . . . 53
3.6.3 Inception results . . . 58
3.6.4 Results comparison . . . 63

3.7 Summary 64

4 Transfer Learning Models Accuracy Improvement in Lung Diseases Classification Using Segmentated X-Ray Images 66

4.1 U-Net - Image Segmentation using Deep Neural Networks . . 66

4.1.1 Architecture 67

4.2 Lungs Segmentation 69

4.2.1 Dataset 69

4.2.2 Training 69

4.2.3 Results 70

4.3 Training Deep Learning Models On Segmented Images 73

4.3.1 Dataset 73

4.4 Results and analysis 73

4.4.1 VGG results 74

4.4.2 ResNet results 79

4.4.3 Inception results 84

4.5 Comparison of results 90

4.6 Comparison with other works 92

5 Conclusions and Future Work 94

5.1 Summary 94

5.2 Contributions 95

5.3 Future Work 95

Bibliography 97

List of Figures

2.1 Ribs separation training samples. Image A is the input and B the corresponding boneless output. 7
2.2 Performance of a trained model in ribs removal task. Image A is an input sample and B is a result with removed ribs 8
2.3 Overview of the screening system developed for tuberculosis detection [1] 11
2.4 Overview of the CST-Voting algorithm [2] 14
2.5 CST-Voting algorithm [2] 14
2.6 The left image shows a tuberculosis cord in a white highlighted box. The right part is a positive tuberculosis culture [3] 16
2.7 "Simplified network architecture. (A) Input to the network is a 224 x 224 grayscale image of a MODSM. tuberculosis culture. The image is passed through the network, and the output of the second fully-connected layer is a probability distribution over the two classes (positive (+): 1 and negative (–): 0). Each block is a stack of feature maps, of dimensions (width x height x number of feature maps). Layer operations take place between each block (see (B)) and are identifiable by the feature map volume produced. Kernels are 3 x 3 and 2 x 2 for convolutional and pooling layers, respectively. The network is trained and evaluated on a dataset of 1008 train/validation and 2502 test images. (B) A schematic representation of the convolution and pooling operations on an input volume." [3] 16

2.8 Sample Chest X-Ray images containing marks of tuberculosis (A) and pneumonia (B). 18
2.9 "Combined distribution of genders and ages among images without (0) and with (1) disease marks." [4] 19
2.10 Example of an original image in SH dataset (left) and the segmented result (right). 20
2.11 Distributions of image (Fig. 2.10, left) and mask (Fig. 2.10, central) areas. [4] 20
2.12 Selected image transformation methods. Image A) shows the original image, B) flipped, C) darkened (brightness change) and finally D) zoomed(cropped and upsized) 21
2.13 T_p vs. F_P using different thresholds of classification.[5] 25
2.14 AUC (Area under the ROC).[5] 26
2.15 Convolutional Neural Network for image processing.[6] . . . 27
2.16 A hypothetical 4x4 image. 28
2.17 Convolution operation. 28
2.18 Max pooling operation. 29

3.1 VGG network with 16 layers. 34
3.2 Residual bloc. 36
3.3 Residual blocs 37
3.4 Resnet-34 architecture 38
3.5 Inception cell introduced with the fist Inception model[7] . . . 39
3.6 $5x5$ convolution replaced by two $3x3$ convolutions. 40
3.7 Inception cell introduced in [8] 41
3.8 Asymmetric convolution [8] 41
3.9 Inception cell using aymmetric convolutions[8] 42
3.10 Wider inception module[8] 43

3.11 "Auxiliary classifier on top of the last 17x17 layer. Batch normalization of the layers in the side head results in a 0.4% absolute gain in top-1 accuracy. The lower axis shows the number of iterations performed, each with batch size 32." [8] 43
3.12 "Two alternative ways of reducing the grid size. The solution on the left violates principle 1 of not introducing a representational bottleneck from Section 2. The version on the right is 3 times more expensive computationally." [8] 44
3.13 "Inception module that reduces the grid-size while expands the filter banks. It is both cheap and avoids the representational bottleneck as is suggested by principle 1. The diagram on the right represents the same solution but from the perspective of grid sizes rather than the operations." [8]. 45
3.14 InceptionV3 architecture. Batch normalization and ReLU nonlinearity are used after every convolution layer.) 45
3.15 VGG16 based model training and validation loss change. . . . 49
3.16 VGG16 based model training and validation accuracy change. 50
3.17 Image A shows the confusion matrix of the obtained results with the model which reached the highest accuracy during the training. Image B show its per class ROC curves. 52
3.18 Two pairs of correctly labelled images containing marks of tuberculosis and pneumonia with their class activation maps. . . 53
3.19 ResNet-50 based model training and validation loss change. . 55
3.20 ResNet-50 based model training and validation accuracy change. 56
3.21 Image A shows the confusion matrix of the obtained results with the model which reached the highest accuracy during the training. Image B show its per class ROC curves. 57
3.22 Two pairs of correctly labelled images containing marks of tuberculosis and pneumonia with their class activation maps. . . 58

3.23 InceptionV3 based model training and validation loss change. 60
3.24 InceptionV3 based model training and validation accuracy change. 61
3.25 Image A shows the confusion matrix of the obtained results with the model which reached the highest accuracy during the training. Image B show its per class ROC curves. 62
3.26 Two pairs of correctly labelled images containing marks of tuberculosis and pneumonia with their class activation maps. . . 63

4.1 "U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations."[9] 67
4.2 An X-Ray image and its corresponding lungs mask. 70
4.3 U-Net training and validation losses change during training. . 71
4.4 Three pairs of CXR images with corresponding, segmented (extracted) lungs. 72
4.5 VGG16 based model training and validation loss change. . . . 75
4.6 VGG16 based model training and validation accuracy change. 76
4.7 Image A shows the confusion matrix of the obtained results with the model which reached the highest accuracy during the training. Image B show its per class ROC curves. 78
4.8 Two pairs of correctly labelled images containing marks of tuberculosis and pneumonia with their class activation maps. . . 79
4.9 ResNet-50 based model training and validation loss change. . 81
4.10 ResNet-50 based model training and validation accuracy change. 82

4.11 Image A shows the confusion matrix of the obtained results with the model which reached the highest accuracy during the training. Image B show its per class ROC curves. 83
4.12 Two pairs of correctly labelled images containing marks of tuberculosis and pneumonia with their class activation maps. . . 84
4.13 InceptionV3 based model training and validation loss change. 86
4.14 InceptionV3 based model training and validation accuracy change. 87
4.15 Image A shows the confusion matrix of the obtained results with the model which reached the highest accuracy during the training. Image B show its per class ROC curves. 88
4.16 Two pairs of correctly labelled images containing marks of tuberculosis and pneumonia with their class activation maps. . . 89
4.17 Two pairs of correctly classified images with their class activation maps. The left columns is non segmented images and the right is the segmented ones. 91

List of Tables

2.1 Different classes of PMLs, their functions and applications [10] 5
2.2 Dataset class distribution. 19
2.3 Hypotetical data distribution 22

3.1 Comparison of transfer-learning based algorithms in terms of accuracy, AUC, F1 score, precision and sensitivity for healthy, pneumonia and tuberculosis classes. 64

4.1 Comparison of all results received on segmented and non-segmented data. 90
4.2 Comparison of different deep learning based solutions trained on the Shenzhen datases. Although our result is not the best, it performs better than any single model (excluding Ensemble). Horizontal line means that those results were not provided in literature. 93
4.3 Comparison of different deep learning based solutions trained on the Montgomery dataset [11]. Our average performance is almost identical to [12]. 93

Chapter 1

Introduction

1.1 Project objectives

Computer vision supported with deep neural networks finds its usefulness in any area of life starting from facial emotion recognition to disease detection. Thanks to the recent technological advances, computer-aided image analysis algorithms compete with professionals in terms of accuracy yet remain unchallenged in speed and volume of reviewed cases. Unlike doctors, computers make quick rational decisions unaffected by emotions and tiredness. The newest WHO (World Health Organisation) report states that just in the United States over 1 million citizens seek care due to pneumonia and there are nearly ten million cases of tuberculosis worldwide. Perhaps some part of it is lethal due to lack of medical staff or human mistake. The manual analysis of x-ray images is a long process requiring radiological expertise and a large volume of time. Deep learning can play a crucial role in exceeding decision making, detecting marks of disease as well as conducting the initial examination and suggesting urgent cases.

1.2 Motivation

Previous approaches required both a large volume of data and strong computing power computers [4]. Instead of following the popular trend of creating new algorithms to solve a problem, we decided to leverage existing tools and show their high accuracy in medical tasks outperforming the suggested solutions [13]. The motivation of this project was to create a pipeline allowing us to detect pulmonary diseases using tiny datasets ($< 10^3$ images per class) (see Chapter 2) and limited computational resources. Also, we want to show the importance of segmentation(see Chapter 2 for details) as a tool focusing our algorithms on information that is crucial in diagnosis. Should models search through pulmonary changes within lungs images omitting redundant data (bones, internal organs)[14], the decision is considered reasonable. To prove that statement, we generate class activation maps marking regions of interest (regions that vastly contributed to the final classification).

1.3 Applications

As mentioned before, the objective of this work is to create a model able to provide relatively good results even if the resources are limited (both data and computational). We believe that hospitals could find this solution useful, especially due to data confidentiality and graphical processing unit (GPU) accessibility.

1.4 Contributions

This work brings a new outlook on pre-trained deep neural networks and their combinations with segmentation models. It makes the following contributions, both theoretical and practical:

1. Segmentation as a mean of focusing transfer learning-based neural networks on regions of interest (containing marks of lung diseases).
2. Transfer learning based classifiers trained on small Chest X-Ray data sets [4] (see chapter 3) and the positive impact of segmentation on labeling accuracy (see chapter 4)
3. Application of different deep learning models in lung diseases problem (see chapter 3 and 4)
4. The usefulness of transfer learning in deep networks training on a small medical dataset (see chapter 3)

1.5 Thesis structure

The first chapter serves as an introduction and motivation to the topic of lung disease classification. We briefly described the contributions and meaningfulness of a given task, yet the real reason is about making the world a better place.

The second chapter describes the essential background and significant related work done in lung disease classification. We present neural networks as a solution to the problem, datasets, and results in evaluation metrics.

Chapter 2

Related Work

As this dissertation focuses on lung diseases (pneumonia and tuberculosis) classification, we first want to discuss selected screening approaches briefly. All techniques use radiography, a medical imaging method that used to be obsolete. However, machine learning and digital advances revived this method [15] and its significance in diagnosis of lungs diseases [16][17][18][19]. Especially, they allow detecting multiple forms of cardiothoracic lessons on x-ray scans. The growing popularity of machine learning models in medical diagnosis correlated with their accuracy is considered as a great success as it leads to better disorder recognition. Recent encouraging results in deep learning applied in the field of lung diagnosis led to the usage of a GPU-based platform which is able to process a large volume of images in high-resolution within seconds and thus exceed the work of radiologists.

2.1 Pixel/Voxel-Based Machine Learning

The availability of computationally powerful machines allowed emerging methods like pixel/voxel-based ML (PML) in medical image analysis/processing. Instead of calculating features from segmented regions, this technique uses voxel/pixel values in input images directly. Therefore, neither segmentation nor feature extraction is required. The performance of PML can possibly

get higher than common classifiers[10] as this method is able to avoid errors caused by inaccurate segmentation and feature extraction.

2.1.1 Overview

Medical image processing/analysis, as well as computer vision, have adopted many PMLs in many tasks. The Table 2.1 provides a brief summary of PML classes, their functions and applications.

PMLs	Functions	Applications
Neural filters (including neural edge enhancers)	Image processing	Enhancement of subjective edges traced by a physician [20]. Edge enhancement from noisy images [21]. Edge-preserving noise reduction [22][23].
Convolution neural networks (including shift-invariant neural networks)	Classification	FP reduction in CAD for detection of masses [24] and microcalcifications [25] in mammography. FP reduction in CAD for lung nodule detection in CXR [26][27]. Character recognition [28]. Face recognition [29].
Massive-training artificial neural networks (MTANNs, including a mixture of expert MTANNs, a LAP-MTANN, an MTSVR)	Classification (image processing + scoring), object detection, suppression and pattern enhancement	FP reduction in CAD for detection of lung nodules in CXR and CT [30]. Distinction between benign and malignant lung nodules in CT. FP reduction in CAD for polyp detection in CT colonography. Bone separation from soft tissue in CXR. Enhancement of lung nodules in CT.
Others	Image processing or classification	Separation of ribs from soft tissue in CXR [31]. Segmenting posterior ribs in CXR [32].

TABLE 2.1: Different classes of PMLs, their functions and applications [10]

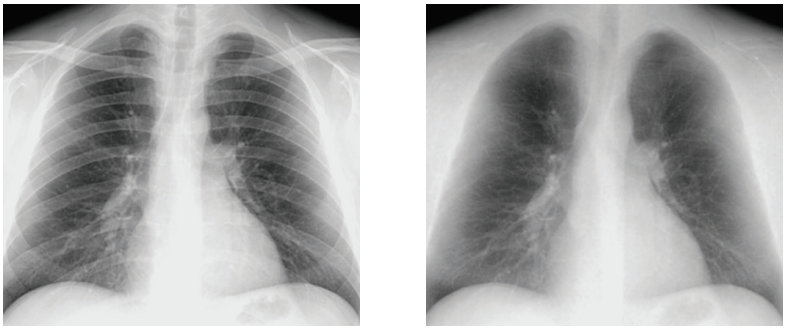
We can distinguish three classes of PMLs [10]: convolution neural networks [27] [33][24] (this includes shift-invariant Neural Networks [34]), neural filters [23] and MTANNs or massive-training artificial neural networks [35] (including multiple variations, i.e., Laplacian eigenfunction MTANN or massive-training support vector regression). Many image analysis tasks used neural filters. Some of them are edge enhancement from noisy images, edge-preserving noise reduction in digital pictures and radiographs [23], and enhancement of subjective edges traced by a physician in cardiac ventriculograms. Convolutional neural networks have been applied to classification in tasks requiring false-positives reduction in CAD schemes for detection of lung nodules [27][26][33] or microcalcifications in chest X-rays, masses in mammography, character, and face recognition. Massive-training artificial neural networks have been used in classification problems such as a false-positive reduction in CAD schemes for detection of lung nodules [30], the distinction between malignant nodules and the benign ones. This class of PMLs has also been applied to suppression, and pattern enhancement of bone tissue from soft tissue in CXR [35] and lung nodules enhancement in X-ray computed tomography.

2.1.2 Bone Separation from Soft Tissue in Chest Radiographs (CXRs) by Use of MTANNs

Chest X-Ray is one of the most frequently used diagnosis when examining medical images for different lung diseases such as pneumonia or tuberculosis. Roughly 1 million of adults require hospitalization because of pneumonia, and about 50,000 dies from this disease annually in the US only [36]. Examination of lung nodules in CSR can lead to overlooking of diseases like lung cancer. However, not all of them are visible in retrospect. Studies show

that 82-95% of lung cancer cases were missed due to at least partially obscured by ribs or clavicle. To address this issue, researchers examined dual-energy imaging, a technique which can produce images of two tissues, which is a "soft-tissue" image and a "bone" one [14]. This technique has many drawbacks, but undoubtedly one of the most important ones is the exposure to radiation.

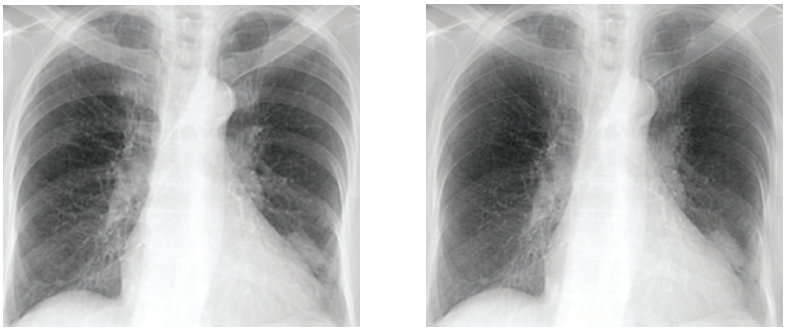
The MTANNs models have been developed to address this problem and serve as a technique for ribs/soft-tissue separation. The idea behind training those algorithms is to provide them with bone and soft-tissue images obtained from a dual-energy radiography system. The MTANN was trained using CXRs as input and corresponding boneless images, as presented in Figure 2.1. Figure 2.2 shows the performance of the model on the unshown image data. The ribs contrast is visibly suppressed in the resulting image, maintaining the soft tissue areas such as lung vessels.



(A) Original input

(B) Boneless image

FIGURE 2.1: Ribs separation training samples. Image A is the input and B the corresponding boneless output.



(A) Test sample

(B) Generated result

FIGURE 2.2: Performance of a trained model in ribs removal task. Image A is an input sample and B is a result with removed ribs

2.2 Approaches to lung disease classification

2.2.1 Extreme learning machine

The grey-level co-occurrence matrix inspired the feature extraction method used in [37], a technique often used in the texture analysis. The proposed method named Spatial Interdependence Matrix or SIM makes use of the co-occurrence statistics to analyze the structural information based on the way human visual system tries interpreting scenes. In this context, the new technique can be implemented to evaluate image structural degradation.

Let us consider an image I and its degraded version J as one set of grey levels in the domain D defined as:

$$D \subset Z^2 \in 0, 1, 2, 3, ..., N \tag{2.1}$$

where N represents the number of all grey levels. The transition of pixel intensities and spacial correspondence between images I and J is arranged into a two dimensional matrix of size NxN, where each element $M_{i,j}$ is defined as:

$$M_{i,j} = \#(i,j) : i = I(p), j = J(p) \, \forall p \in D \tag{2.2}$$

Where the #. operator represents the set cardinality of I(p) and J(p) intensities for all pixels p, which belongs to both I and J.

The three structural attributes extracted from the Spatial Interdependence Matrix matrix relate to the image J (degraded) when compared to I [37]. Those attributes are chi-square (Chi), inverse difference moment (Idm) and correlation (Corr), and represent the degradation level in three different perspectives: structural independence, structural degradation, structural similarity, respectively.

To extract the structural attributes Idm and Corr, the authors in [37] used an asymmetric version of $M_S = \frac{M+M^T}{2}$, where M^T is simply the transposed matrix M. The matrix M is normalized to obtain the pair-transitions weight as a probabilities approximation. The structural attributes in terms of $M_{i,j}$ are described as follows:

$$Corr = \sum_{i,j=0}^{N-1} \frac{(i-\mu_i)(j-\mu_j) M_{i,j}}{\sqrt{\sigma_i^2 \sigma_j^2}} \in [-1,1] \tag{2.3}$$

$$Idm = \sum_{i,j=0}^{N-1} \frac{M_{i,j}}{1+|i-j|} \in [0,1] \tag{2.4}$$

$$Chi = \sum_{i=0}^{N-1} \frac{(O_i - E_i)^2}{E_i} \tag{2.5}$$

$$\mu_i = \sum_{i=0}^{N} \frac{i}{N} \tag{2.6}$$

where μ_j and μ_i correspond to the average values and σ_j and σ_i relate to the standard deviation of the matrix M for each column j and row i. O_i and E_i stand the observed and expected weights respectively, in the diagonal matrix of M (i = j) [37].

The Spatial Interdependence Matrix gives us a visual pattern which is useful to interpret degraded images. If an image is undegraded, the weights are distributed near the diagonal of the matrix. Otherwise, different patterns appear depending on the degradation of the structure. When relating to the lung's diseases, the SIM pattern of fibrosis varies from one image to another image of the healthy lungs. The structures of fibrosis are spread through the lungs area, whereas in healthy lungs they are sparse and small.

A set of attributes extracted from the Spatial Interdependence Matrix (Correlation, chi-square, and inverse difference moment) is used to asses the structural characterization of lung images. Using prior knowledge that the CT images have blurred structures, authors in [37] convolve the inputs with a gaussian kernel. They set the number of gray levels to 64 to compute the Spatial Interdependence Matrix.

The presented lung disease descriptor [37] consists of three attributes in a vector $A = [Corr, Idm, 1 - Chi]$, which are extracted from the SIM matrices of training images. Later on, they trained a multi-layer perceptron (MLP) for 100 epochs. The trained models allowed them to reach the accuracy level above 96%

2.2.2 Automatic CXR screening system

This method is a multistage processing system based on a multistage framework developed (see Figure 2.3) by researchers in [1] which first segment

images and then by combining texture and shape features tries to predict a disease presented on CXRs. The algorithm uses similar intuition as radiologists during lung examination, which is the comparison of right and left lung fiends. The texture features describe the inside lungs fields, and the shape features focus on including the relevant geometrical characteristics

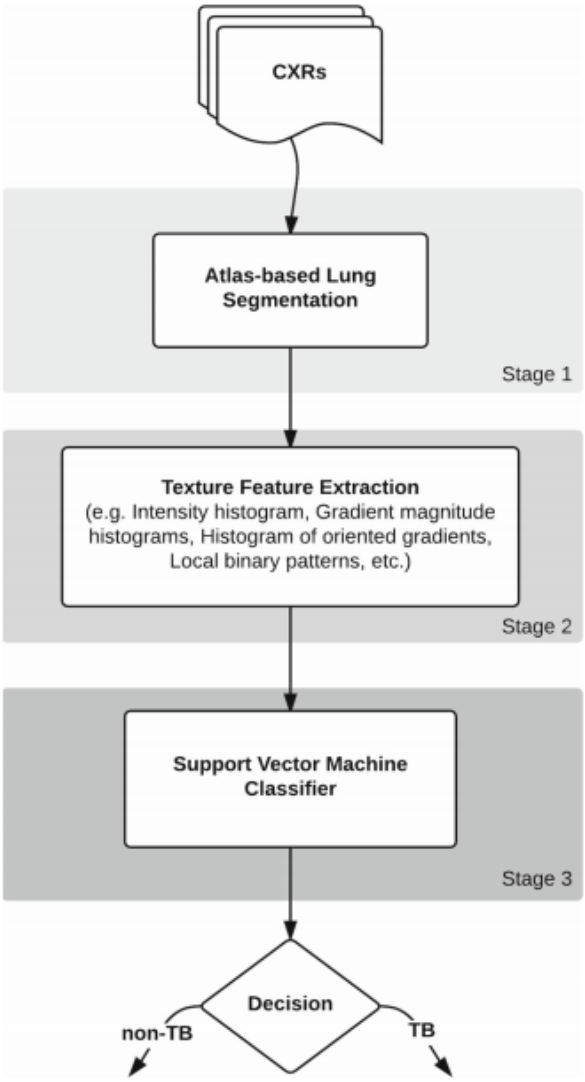


FIGURE 2.3: Overview of the screening system developed for tuberculosis detection [1]

The presented method on 2.3 consists of three consecutive stages, which are segmentation, feature extraction, and classification, respectively. The

first stage is composed of three phases. The first one is content-based image retrieval using Bhattacharyya shape similarity measure [38] and a partial Radon transform [39]. Then, building a patient anatomic model of lung shape based on SIFT-flow [40]. And finally, taking out lungs boundaries with an approach of graph cut optimization. The next stage is texture feature extraction of segmented lungs using features such as intensity histogram, gradient magnitude, a histogram of oriented gradients, etc. The final step is classification using a support vector machine model (SVM) [41].

2.2.3 Semi-Supervised Learning

This section briefs the SSL algorithm proposed in [2] for the classification of the pulmonary disease, which uses on the ensemble method called CST-Voting [42]. The idea behind this algorithm is to generate a classifier by applying multiple Semi-Supervised Learning methods to one dataset. Using this theory, the researchers in [2] created an ensemble model consisting of three algorithms, which are: self-training, co-training, and tri-training.

Self-training algorithm implements a simple arbitrary model that is trained on a small subset of labeled data L and tries to predict U. If the probability of a predicted instance is higher than the defined confidence level, it is added to L. We repeat this procedure until the set U is empty, or we meet some stopping criterion.

Co-training assumes that we have weak algorithms that are trained on the set of labeled instances (L). Later, the two algorithms classify instances of the unlabeled set U' (fixed-size subset of all unlabeled data U) and move to the L those where the prediction was the most confident. After removing samples from U', we refill this set with new instances from U. We repeat this procedure until the set U is empty, or we meet some stopping criterion.

Tri-training is an extension of the co-training method. Similarly to the previous approach, we have multiple weak learners, yet this time, its number is increased to three. Each classifier is trained on the labeled data and predicts the class for instances in the unlabeled set. The majority makes the final decision, and the classified sample is added to the labeled data. One way of looking at this method is that the majority teaches the minority - the majority of votes decides about the final class for all learners.

Those self-labeled methods operating in different manners take full advantage of the encoded information from the unlabelled data set. The crucial feature of making those methods different is the mechanism behind labeling the unlabelled data. Tri-training, as well as for self-training, are single-view algorithms, whereas co-training is a multi-view one. What is more, both tri-training and co-training are ensemble methods themselves. An overview of the described CST-Voting algorithm is presented in Figure 2.4. At first, all the mentioned semi-supervised algorithms (tri-training, co-training, and self-training), which build the ensemble learn using the same unlabelled U and label data set L. Afterwards, the decision on an unlabelled test sample combines all separate predictions of the semi-supervised models. Moreover, here, a simple voting methodology is applied - make a decision based on the majority of votes [2]. The high-level explanation of the described CST-Voting algorithm is shown in Figure 2.5.

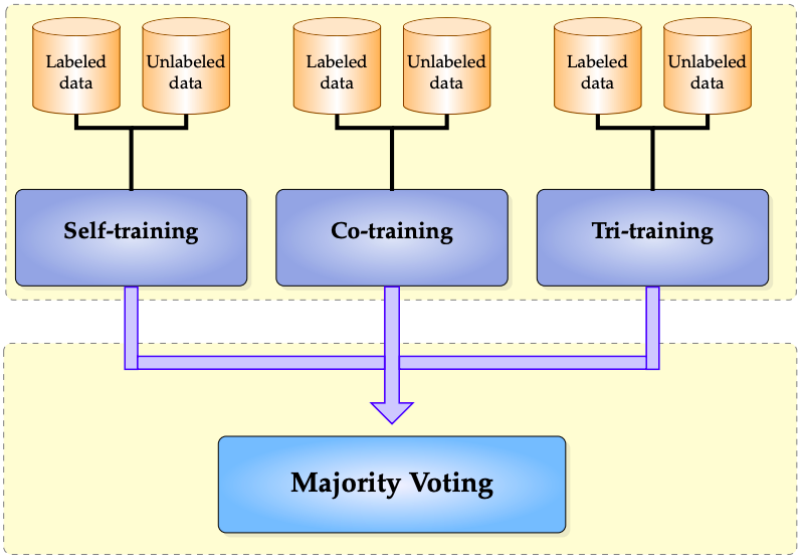


FIGURE 2.4: Overview of the CST-Voting algorithm [2]

Input: L-Set of labeled instances.
U-Set of unlabeled instances.
C-Base learner.

Output: The labels of instances in the testing set.

/* Training phase */
1: Self-training(L, U)
2: Co-training(L, U)
3: Tri-training(L, U)

/* Voting phase */
4: **for each** $x \in T$ **do**
5: Apply self-training, co-training and tri-training on x.
6: Use majority vote to predict the label y^* of x.
7: **end for**

FIGURE 2.5: CST-Voting algorithm [2]

2.3 Automatic diagnostic using Deep Learning in MODS digital images analysis

The Microscopic Observed Drug Susceptibility (MODS) is a test to analyze tuberculosis contamination and drug resistance using a sputum test in about one week period with minimal effort and high effectiveness. In spite of its points of interest, MODS is as yet restricted in remote, low asset settings, since it requires changeless and prepared specialized staff for the picture based diagnostics. Henceforth, it is essential to create elective arrangements, given solid mechanized investigation and elucidation of MODS societies. Researchers in [3] trained and then assessed a deep convolutional neural network 2.8 for MODS digital images interpretation and diagnostics.

The researchers used a dataset which consisted of almost 13 thousand MODS digital images, as shown in Figure 2.6. It contained only two classes od samples - positive, containing marks of tuberculosis and the negative (healthy) ones. Similarly to this thesis, [3] used an adaptation of VGG16 network [43] (also see Section 3.2). A 15 layers deep model comprising fully-connected, max-pooling, convolutional layers organized into five blocks; four convolutional layers separated by pooling layers terminated by one fully-connected layer (classifier). The architecture is given in Figure 2.7

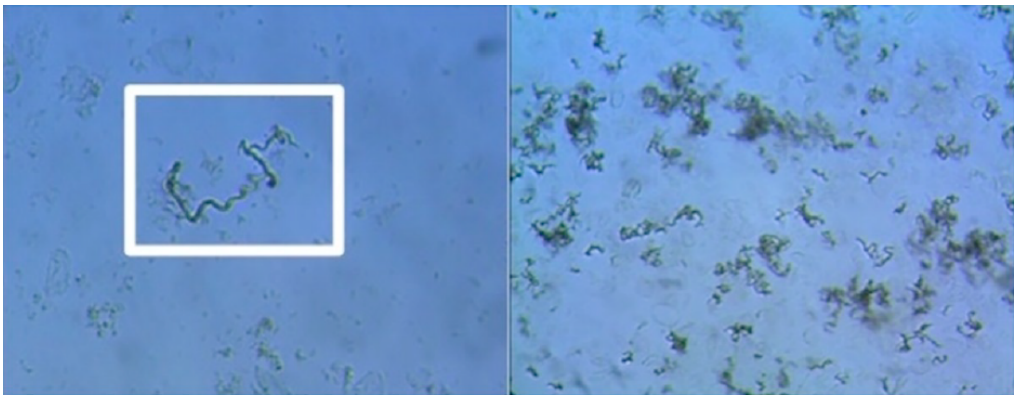


FIGURE 2.6: The left image shows a tuberculosis cord in a white highlighted box. The right part is a positive tuberculosis culture [3]

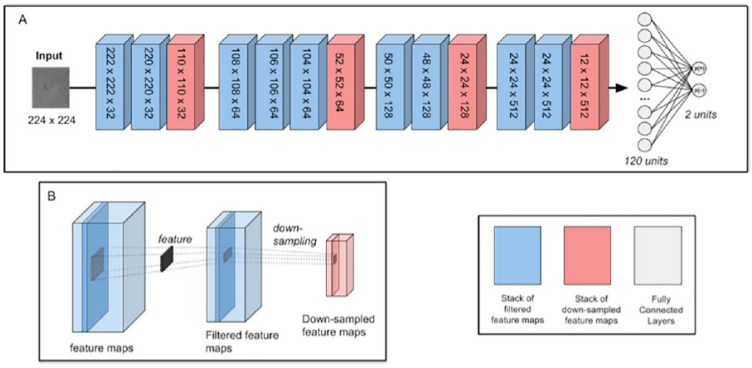


FIGURE 2.7: "Simplified network architecture. (A) Input to the network is a 224 x 224 grayscale image of a MODSM. tuberculosis culture. The image is passed through the network, and the output of the second fully-connected layer is a probability distribution over the two classes (positive (+): 1 and negative (–): 0). Each block is a stack of feature maps, of dimensions (width x height x number of feature maps). Layer operations take place between each block (see (B)) and are identifiable by the feature map volume produced. Kernels are 3 x 3 and 2 x 2 for convolutional and pooling layers, respectively. The network is trained and evaluated on a dataset of 1008 train/validation and 2502 test images. (B) A schematic representation of the convolution and pooling operations on an input volume." [3]

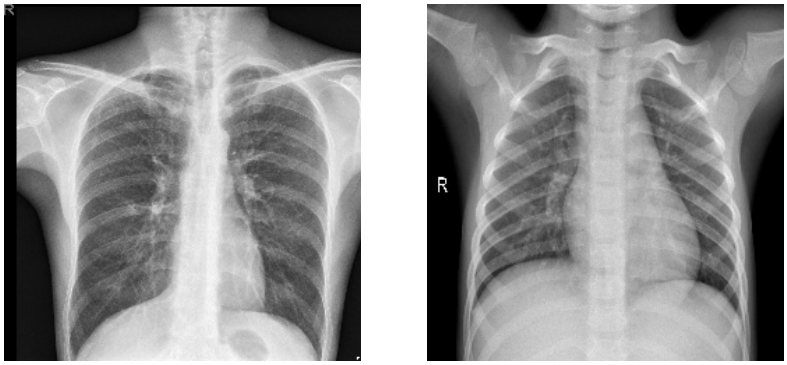
2.4 Deep Learning Approaches in Chest X-Ray analysis

Recent applications of Deep Neural Networks [44][43][45][8][7] lead to major improvements in medical imaging. The efficiency of dimensionality reduction algorithms like lung segmentation was demonstrated in the Chest X-Ray image analysis [46]. Researchers in [4] aimed at improving tuberculosis detection on relatively small data sets ($< 10^3$ images per class) by incorporating deep learning segmentation and classification methods from [46]. The further exploration of these techniques is the topic of this thesis. Therefore we focus on various deep learning methods in lung diseases classification in chapters 3 and 4

2.5 Dataset

This work combines two relatively small datasets ($< 10^3$ images per class) datasets for the classification (pneumonia and tuberculosis detection) and segmentation purposes. We selected 306 examples per "diseased" class (306 images containing marks of tuberculosis and 306 images with pneumonia) and 306 of healthy patients contributing to a set of 918 samples coming from different patients. Sample images coming from both datasets are presented in Figure 2.8.

The Shenzhen Hospital dataset (SH) [13][47] containing CXR images was created by People's Hospital in Shenzhen (China). It includes both abnormal (containing marks of tuberculosis) and the standard CXR images. Unfortunately, the dataset is not well-balanced in terms of absence or availability of



(A) tuberculosis case

(B) pneumonia case

FIGURE 2.8: Sample Chest X-Ray images containing marks of tuberculosis (A) and pneumonia (B).

disease, gender, or age as presented on the chart 2.9. Here, we extracted only 153 samples of healthy patients (153 from both datasets) and 306 of those labeled with marks of tuberculosis. Selecting information about one class from different resources ensures that the model does not learn features typical for the method of taking images, e.g., lens.

Pneumonia is an inflammatory condition of the lung affecting the little air sacs known as alveoli. Standard symptoms comprise of a blend of a dry hack, inconvenience breathing, chest agony, and fever. The Labeled Optical Tomography and Chest X-Ray Images for Classification dataset [48] include selected images of patients from the Medical Center in Guangzhou. It consists of data with two classes - normal and those containing marks of pneumonia. All data comes from the patient's routine clinical care. The volume of the complete dataset includes thousands of validated OCT and X-Ray images yet for our analyze we wanted to keep the dataset tiny and evenly distributed thus only 153 images were taken (another 153 images come from the tuberculosis dataset) from the resources labeled as healthy and 306 as pneumonia - both chosen randomly. The exact dataset class distribution is presented in

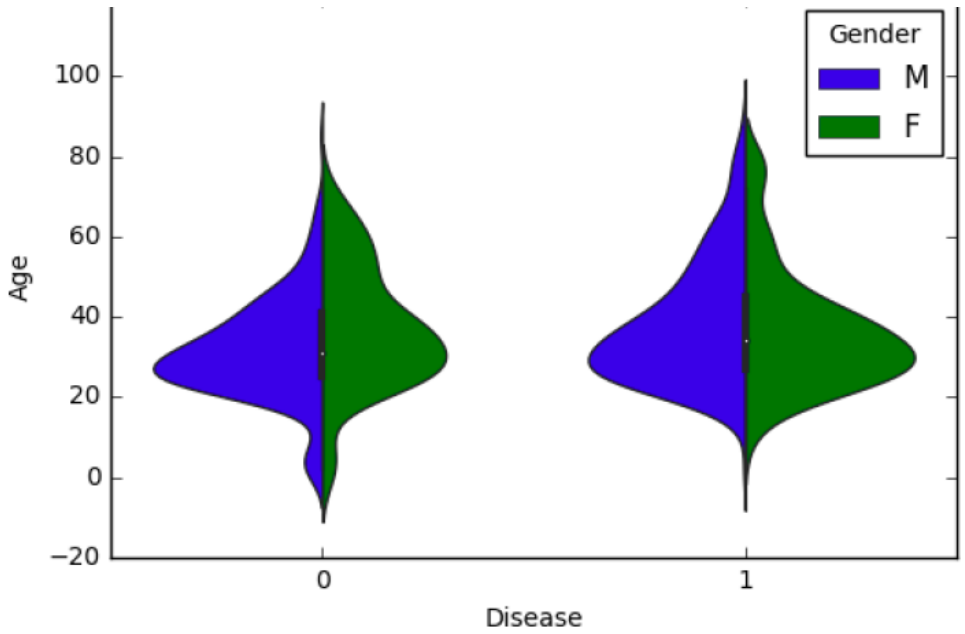


FIGURE 2.9: "Combined distribution of genders and ages among images without (0) and with (1) disease marks." [4]

Table 2.2

	Healthy	Pneumonia	Tuberculosis
Number of samples	306 (153 per dataset)	306	306

TABLE 2.2: Dataset class distribution.

External segmentation of left and right lung images (exclusion of redundant information; bones, internal organs, etc. - Fig. 2.10) was proven to be effective in gaining better prediction accuracy [4].

To extract lungs information and exclude outside regions, we used the manually prepared masks included in the extension of the SH dataset, namely, the segmented SH dataset. Due to nonidentical borders and lung shapes, the segmentation data has high variability although its distribution is much similar to the regular one, comparing to image area distribution as presented in

(A) X-ray image

(B) extracted lungs

FIGURE 2.10: Example of an original image in SH dataset (left) and the segmented result (right).

FIgure 2.11.

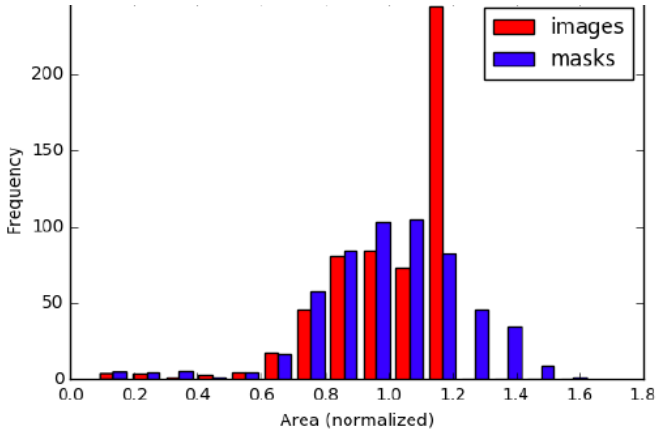
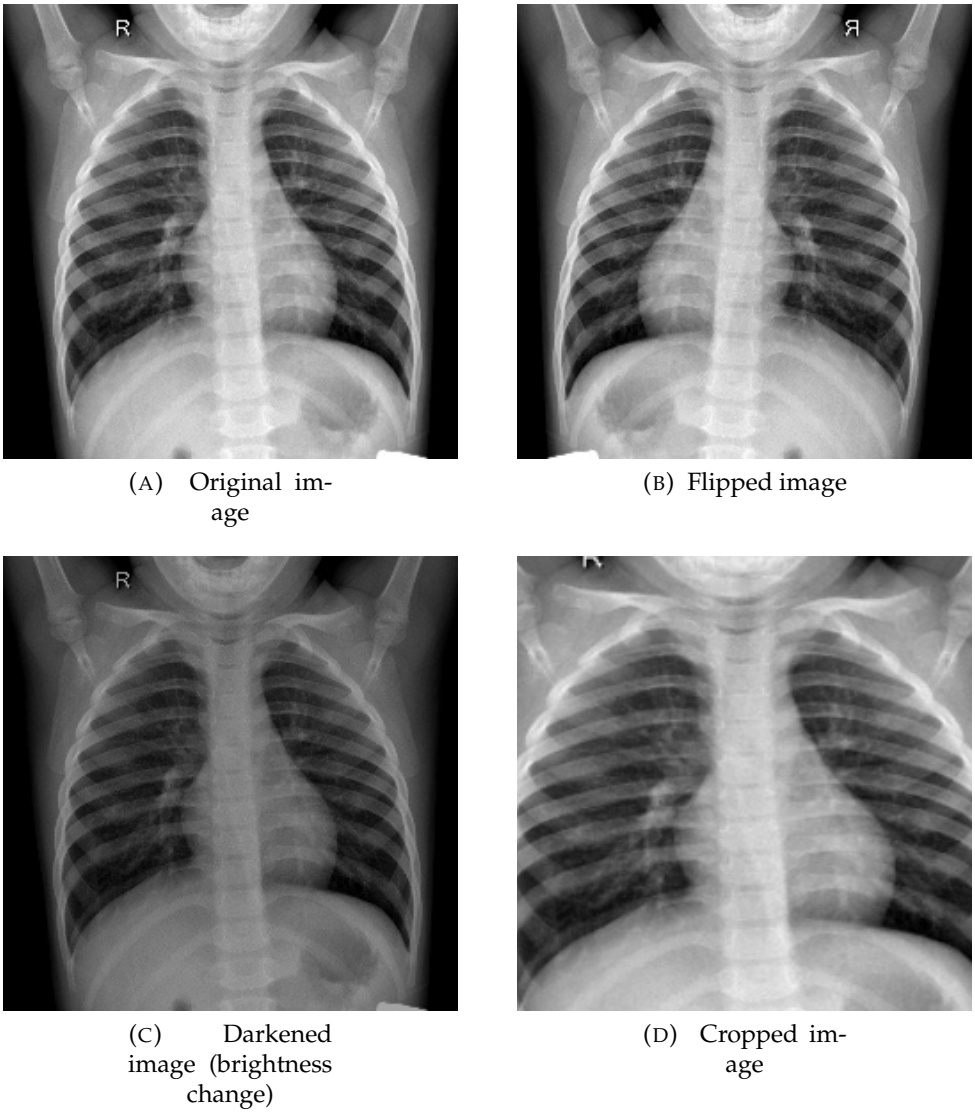


FIGURE 2.11: Distributions of image (Fig. 2.10, left) and mask (Fig. 2.10, central) areas. [4]

2.6 Image Data Augmentation

Model-based methods greatly improve their predictions when increasing the number of training samples. When a limited amount of data is available, some transformations have to be applied to the existing dataset to synthetically increase the volume of the training set. Researchers in [44] incorporates three techniques to augment the training dataset size. The first approach was

to randomly crop of a 224x224 pixel fixed-size window from a 256x256 pixel image. The next technique was flipping the image horizontally, which allowed capturing information about reflection invariance. Finally, the third method added randomly generated lightning to capture color and lightning variation. Image 2.12 shows sample operations on the image data.



(A) Original image

(B) Flipped image

(C) Darkened image (brightness change)

(D) Cropped image

FIGURE 2.12: Selected image transformation methods. Image A) shows the original image, B) flipped, C) darkened (brightness change) and finally D) zoomed(cropped and upsized)

2.7 Evaluation

Evaluation of a machine learning model plays a crucial part in all projects. One metric might not be satisfying when dealing with the effectiveness of classifiers, and thus in this thesis, we will use the following metrics; accuracy, F1-score, precision, sensitivity, specificity, a graphical performance - ROC (receiver operating characteristic curve) and AUC (Area under the ROC curve). In this section, we define metrics for binary problems. To extend it for a three-class problem we calculate them using the approach "one versus the others" per every label [49].

Accuracy is the number of correctly classify objects to the total predictions:

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made} \tag{2.7}$$

Now, let us consider a situation given in Table 2.3. We see that the data is completely imbalanced, therefore predicting just the class neutral provides us with the accuracy of 86% which is considered high. However, we are not supposed to assume that the model is valid when the decision is based on the dominating class. Therefore, we have to quantify our algorithms with the following properties: recall and precision.

Class	**Number of instances**
Healthy	600
Tuberculosis	50
Pneumonia	50

TABLE 2.3: Hypotetical data distribution

Recall (or sensitivity) R measures the number of classified samples with respect to all the relevant ones. It calculates the ratio of true positives T_p to

the sum of false negatives F_n and true positives T_p.

$$R = \frac{T_p}{T_p + F_n} \tag{2.8}$$

Precision P explains how selected items are relevant. It is defined as the ratio of true positives T_p to the sum of false positives and true positives:

$$P = \frac{T_p}{T_p + F_p} \tag{2.9}$$

That is to say; recall tells us how many samples we missed (in a positive class). Precision tells us what proportion of positive samples were correctly classified.
If one sample was diagnosed as tuberculosis (out of 50) and the rest as other, then the precision P equals 100%. However, recall would be as low as 2%.
There is likewise specificity which is characterized as the proportion of true negatives (T_n) to the total of true negatives and false positives

$$Specificity = \frac{Number\ of\ True\ Negatives}{Number\ of\ True\ Negatives + Number\ of\ False\ Positives} \tag{2.10}$$

If a model reaches 100% of specificity, then it missed no True Negatives, there were no False Positives - negative samples labeled as positive ones. However, there is still a high risk of having False Negatives.
Using the example in Table 2.3, the specificity for the class Neutral is 0 knowing that there were no true negatives (T_n)

The F1-score is defined by the following formula:

$$F1 = 2 * \frac{Precission * Recall}{Precission + Recall} \tag{2.11}$$

This metric is a combination of recall and precision, keeping a balance between them.

Another way to show a classification model performance is a graph called ROC (receiver operating characteristic curve) which creates a plot of the True Positive Rate against False Positive Rate.

True Positive Rate (TPR) is just another name for Recall (R). False Positive Rate (FPR) is defined by the following formula:

$$FPR = \frac{F_p}{F_p + T_n} \tag{2.12}$$

A ROC graph is a plot of True Positive Rate vs. False Positive Rate using various classification thresholds. Increasing the threshold assigns fewer samples as positive and thus decrease True Positives and False Positives. The reverse situation takes place when lowering the classification threshold.

A typical ROC curve can be seen on image 2.13

In order to compute the values in a ROC curve, we use a sorting-based algorithm called AUC. AUC means "Area under the ROC Curve," and it measures the area underneath the two-dimensional curve from (0, 0) to (1, 1).

A typical ROC curve can be seen on image 2.14

Area Under the ROC Curve provides us with an aggregated measure of

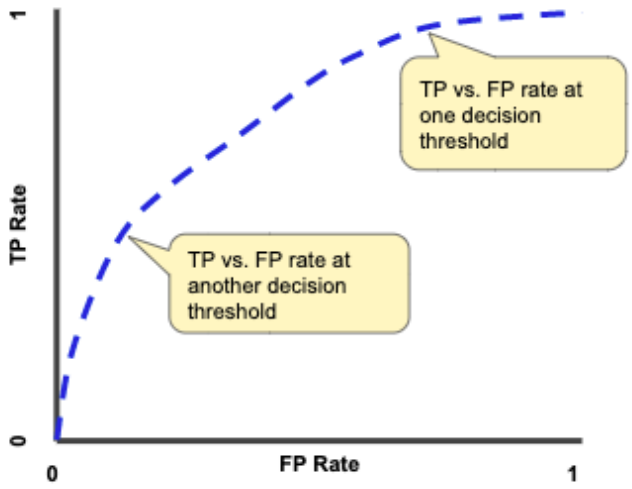


FIGURE 2.13: T_p vs. F_P using different thresholds of classification.[5]

performance considering any classification threshold. It might be interpreted as the probability that a given model assigns a random sample to a positive class.

The area defined by AUC varies from 0 to 1. A model who predicts a correct class in 100% of cases has an AUC of 1.0 and another whose predictions are 100% wrong has an AUC of 0.0.

This metric is desirable thanks to two reasons:

– classification-threshold-invariance

– scale-invariance

The first one measures how well model predictions are irrespective of the chosen threshold. The second one estimates the rank of the predictions, rather than their absolute values.

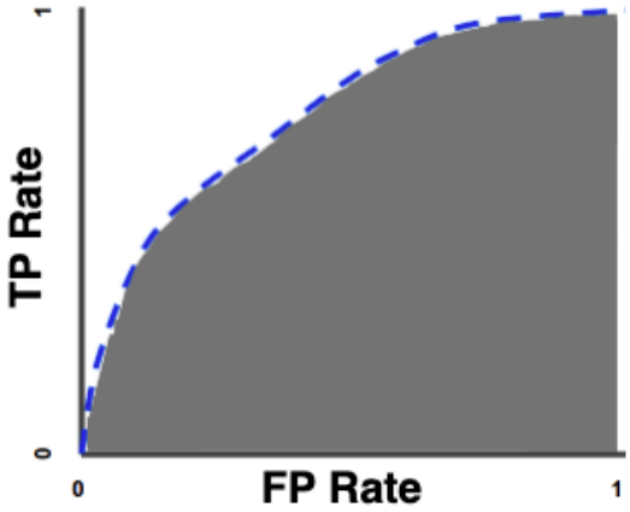


FIGURE 2.14: AUC (Area under the ROC).[5]

2.8 Convolutional Neural Networks

After publishing AlexNet ([44]) in 2012, convolutional neural networks (CNN) renewed the interest of the research community. CNN and subsequent deep models such as VGG ([43]) proved their usefulness, especially in computer vision-related tasks. The contribution of published work demonstrates that those models are suited much better at capturing different features than traditional algorithms which heavily rely on different feature engineering methods (e.g., gradient change).

In a classical formulation of a convolutional neural network used in classification, CNN consists of multiple convolution layers followed by pooling operators.

The convolution layers are made of kernels, small tensors compared to windows which process input and output information. Those operators can successfully capture the spatial and temporal dependencies in an image and

thus learn different local features like straight lines (horizontal or vertical) and curves while upper layers (hidden) can perform detection of more sophisticated information like rectangles or circles based on the received input, therefore, understand it better. As processed data flows higher to deeper layers, a network learns more "abstractive" combinations.

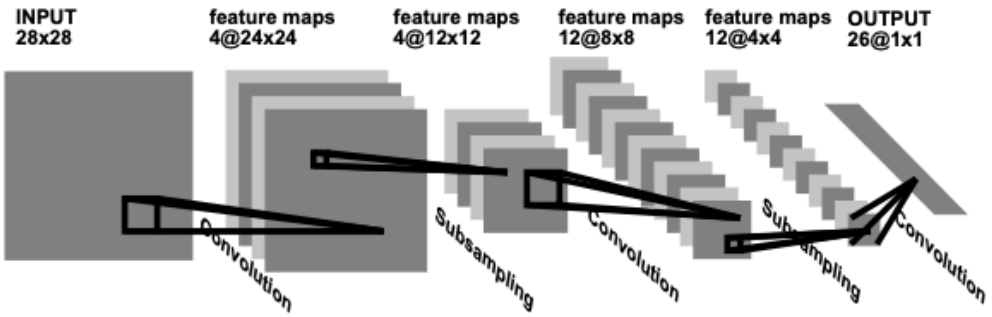


FIGURE 2.15: Convolutional Neural Network for image processing.[6]

Let us consider a hypothetical image presented in Figure 2.16 where we can distinguish three color layers: red, green, and blue respectively. The role of a convolutional neural network is to reduce the image space into a form much more comfortable to process without loss of any information crucial for obtaining a valid prediction. This example shows an image with small dimensionality (4x4), yet this aspect gains its importance along with the increasing size of an input, e.g., 32 Megapixel (6464 x 4864).

The example in Figure 2.17 shows the process of convolution operation, which extracts valuable input features and processes this information to the next level, whereas reducing the dimensionality.

Convolution can be viewed as a sequence of operations where a single operation centered on a group of surrounding values results with a single output o (provided we have only one kernel).

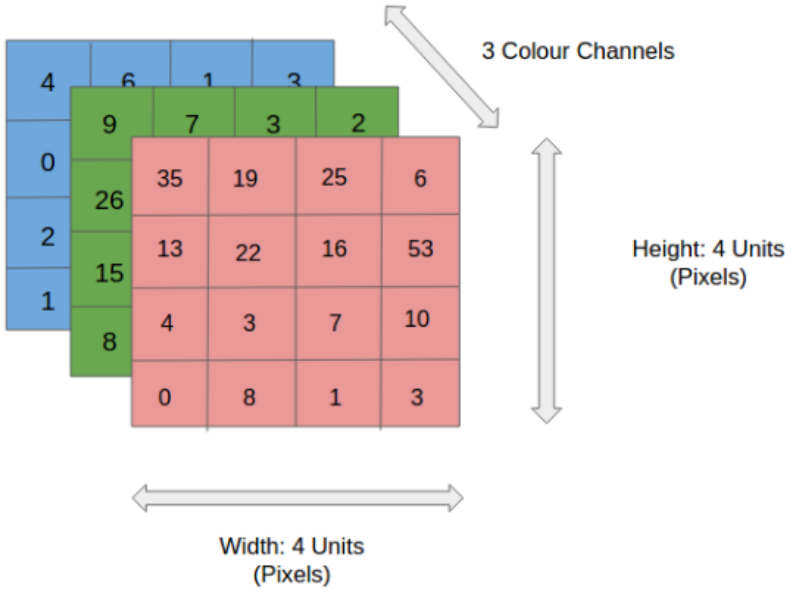
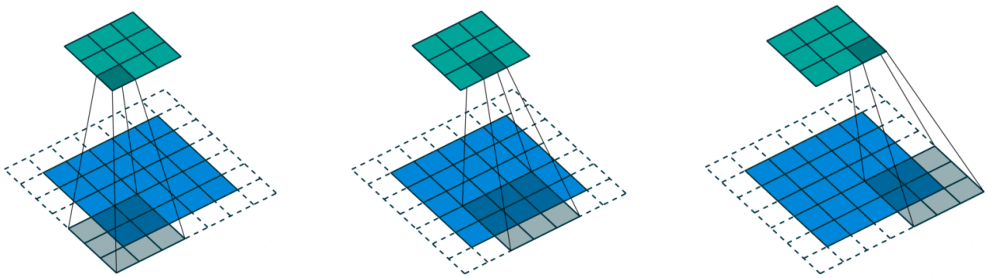


FIGURE 2.16: A hypothetical 4x4 image.



(A) (B) (C)

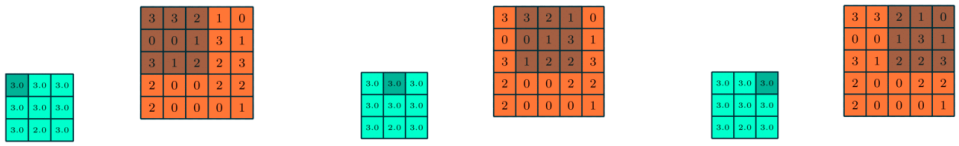
FIGURE 2.17: Convolution operation.

$$o = f\left(W_{x,j} + b\right) \tag{2.13}$$

where $W \in \mathbb{R}^{p,q}$ is a weight matrix (kernel), p is the output size of the convolution, q is the window size, f represents the non-linearity, and b is bias. Both parameters b and W are shared across all inputs.

Similarly to the convolution layer, the Pooling operator is also responsible for spacial size reduction and thus decreasing computational resources used in processing data, albeit the pooling layer contains no parameters (there is no

channel). Rather, pooling operators are deterministic, normally ascertaining either the most extreme or the average estimation of the components in the pooling window. These activities are max pooling and average pooling. Figure 2.18 presents the extraction of dominant, rotational invariant information - max pooling.



(A) (B) (C)

FIGURE 2.18: Max pooling operation.

The ConvNet effectively learns the relations between surrounding pixels throughout an image. Thanks to the convolution, the input is mapped into a constrained, abstracted representation which results in the output describing dominant features

2.9 Summary

This chapter briefly reviewed the work related to the problem: Extreme Learning Machines, Semi-Supervised Learning models, and Automatic CXR screening system. We also introduced previous deep learning methods used in Chest X-Ray analysis, pulmonary disease datasets, image data augmentation techniques, and different types of results measurements used in classification. We briefed deep convolutional neural networks and explained the operations they conduct in image data analysis.

Chapter 3

Transfer Learning in Lung Diseases Classification

3.1 Transfer learning

Transfer learning is a method of optimization of the training process by using tools pre-trained in a different task. A pre-trained model is reused as a base for a different task.

"Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned." - Chapter 11: Transfer Learning, Handbook of Research on Machine Learning Applications, 2009.

This is a very popular approach in computer vision related tasks using deep neural networks when data resources are limited. Therefore, to create a starting point for a new task, we incorporate the pre-trained models skilled in solving similar problems. This method is crucial in medical image processing due to the shortage of sample volume.

In deep neural networks, feature extraction is conducted but passing raw data through models specialized in other tasks. Here, we can refer to deep

learning models such as ResNet where the last layer information serves as input features to a new classifier.

3.1.1 Pre-trained models approach

Transfer learning in deep learning problems can be performed using a common approach called pre-trained models to approach.

Here we can distinguish the three following approaches:

- Reuse Model
- True Model
- Select Source Model

The first option, Reuse Model, states that a pre-trained model can produce a starting point for another model used in a different task. This involves the incorporation of the whole model or its parts.

In the second approach, an adopted model may or may not need to be refined on the input-output data for the new task.

The third option considers selecting one of the available models. It is very often that research institutions publish their algorithms trained on challenging datasets which may fully or partially cover the problem stated by a new task.

3.1.2 ImageNet

ImageNet [50] is a project that helps computer vision researches in classification and detection tasks by providing them with a large image dataset. This database contains roughly 14 million different images from over 20.000 classes. ImageNet also provides bounding boxes with annotations for over 1 million images, which are used in object localization problems.

In this work, we will focus on three models VGG, ResNet, and Inception pre-trained on the ImageNet dataset.

3.2 VGG16

VGG model is a deep convolutional neural network proposed by researchers A. Zisserman and K. Simonyan from the University of Oxford [43] containing over 138 million parameters. This model was able to achieve 7.4% error rate on the ImageNet dataset (see section 3.1.2). It improved the AlexNet [44] network by changing the kernel size and instead of 11x11 and 5x5 filters in the first two layers, it implemented multiple smaller ones 3x3 filters one after another.

3.2.1 VGG16 Architecture

The VGG convolutional network is a model with 16 layers trained on fixed-size images. The input is processed through a set of convolution layers which use small-size kernels with a receptive field 3x3. This is the smallest size allowing us to capture the notion of up, down, right, left, and center. The architecture also incorporates 1x1 kernels which may be interpreted as linear input transformation (followed by nonlinearity (see section 2.8). The stride of convolutions (number of pixels that are shifted in every convolution - step size) is fixed and set to 1 pixel; therefore the spatial resolution remains the same after processing an input through a layer, e.g., the padding is fixed to 1 for 3x3 kernels. Spatial downsizing is performed by five consecutive pooling (max-pooling) layers, which are followed by some convolution layers. However, not all of them are followed by max-pooling. The max-pooling operation is carried over a fixed 2x2 pixel window, with a stride of 2 pixels. This pile of convolutional layers ends with three Fully-Connected (FC) layers where the first two consist of 4096 channels each and the third one 1000 as it

performs the 1000-way classification using softmax. All hidden layers have the same non-linearity ReLU(rectification) [44].

Figure 3.1 visualises the architecture of the VGG model with 16 layers.

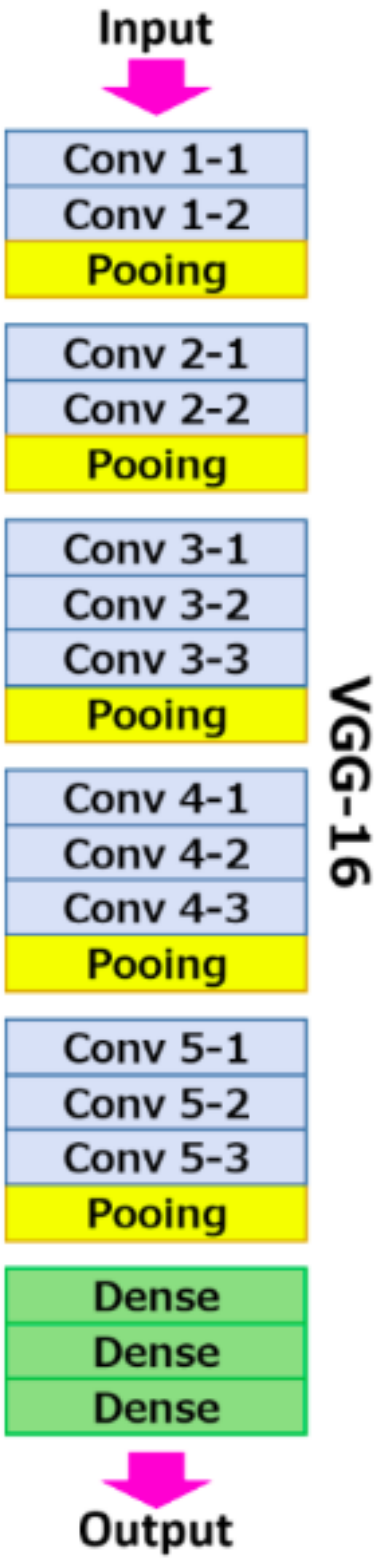


FIGURE 3.1: VGG network with 16 layers.

3.3 ResNet-50

The ResNet convolutional neural network is a 50 layers deep model trained on more than a million fix-sized images from the ImageNet dataset (see 3.1.2). The network classifies an input image into one of 1000 object classes like car, airplane, horse or mouse. The network has learned a plentiful amount of features thanks to the training images diversity and can achieve a 6.71% top-5 error rate on the ImageNet dataset (see section 3.1.2).

3.3.1 Residual Bloc

When expanding the number of layers in a convolutional neural system, a normal phenomenon is to see a decreasing error. Unfortunately, the opposite effect appears where accuracy saturates and eventually degrades. This, however, is not caused by overfitting yet vanishing gradient [45].

Because of the effect related to vanishing, gradient, researchers were not able to build deeper networks as they did not perform better than their shallower counterparts. The main idea of the ResNet model is introducing an "identity shortcut connection"(also residual bloc or skip connection) which skips one or more layers. An example of such operation is presented in Figure 3.2

Let us consider a deep neural network block whose accurate output distribution is denoted as $H(x)$ transformation of input x. The following formula defines the difference or the residue between those arguments:

$$R(x) = Output - Input = H(x) - x \tag{3.1}$$

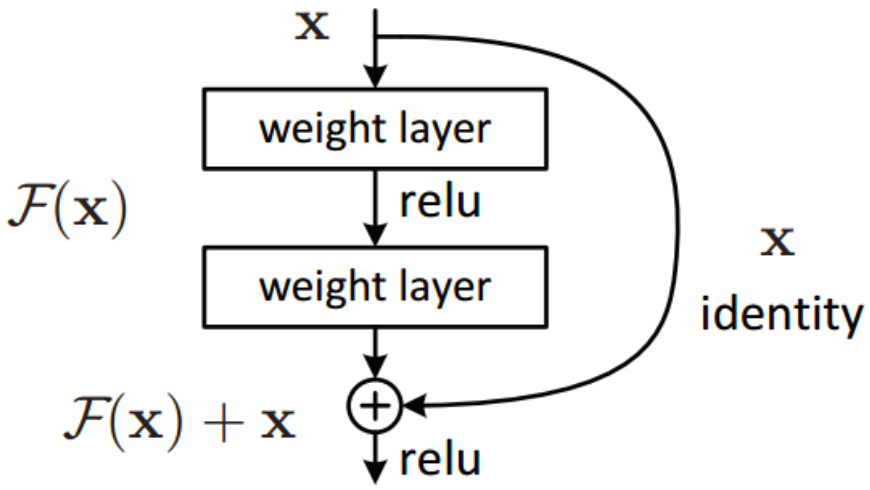


FIGURE 3.2: Residual bloc.

After rearrangement, we obtain:

$$H(x) = R(x) + x \tag{3.2}$$

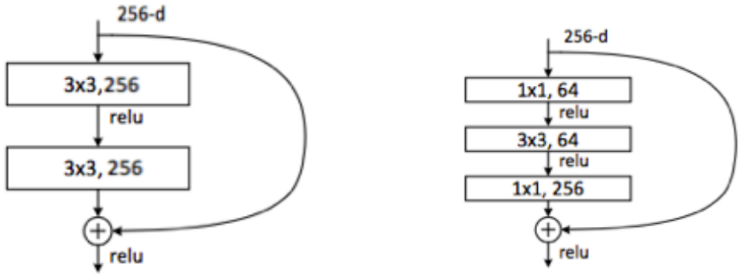
The residual block tries to learn the correct output $H(x)$ and since there is an identity connection there through skipping x to the output, the block learns the residual $R(x)$. Traditional networks try to predict output distribution within their layers, whereas a residual deep convolutional neural network learns the residual. Therefore, those blocks are called Residual.

3.3.2 ResNet50 Architecture

The ResNet-50 convolutional neural network is built of 5 stages, each having convolutions and identity blocks. Every convolution block consists of 3 convolutional layers. Figure 3.4 shows an architecture of a slightly different network (ResNet-34). However, the idea behind its sibling model remains

the same. The only difference is in residual blocks; unlike those in ResNet-34 (Figure 3.3 A) ResNet-50 replaces every two layers in a residual block with a three-layer bottleneck block and 1x1 convolutions, which reduce and eventually restore the channel depth. This allows reducing a computational load when a 3x3 convolution is calculated (Figure 3.3 B).

The model input is first processed through a layer with 64 filters each 7x7 and stride 2 and downsized by a max-pooling operation, which is carried over a fixed 2x2 pixel window, with a stride of 2 pixels. The second stage consists of three identical blocks, each containing a double convolution with 64 3x3 pixels filters and a skip connection block. The third pile of convolutions starts with a dotted line (Figure 3.4) as there is a change in the dimensionality of an input. This effect is achieved through the change of stride in the first convolution bloc from 1 to 2 pixels. The fourth and fifth groups of convolutions and skip connections follow the pattern presented in the third stem od input processing, yet they change the number of filters (kernels) to 256 and 512, respectively[45]. This model has over 25 million parameters.



(A) Resnet-34 residual bloc.

(B) Resnet-50 residual bloc.

FIGURE 3.3: Residual blocs

Visualisation of the network architecture can be found in Figure 3.4

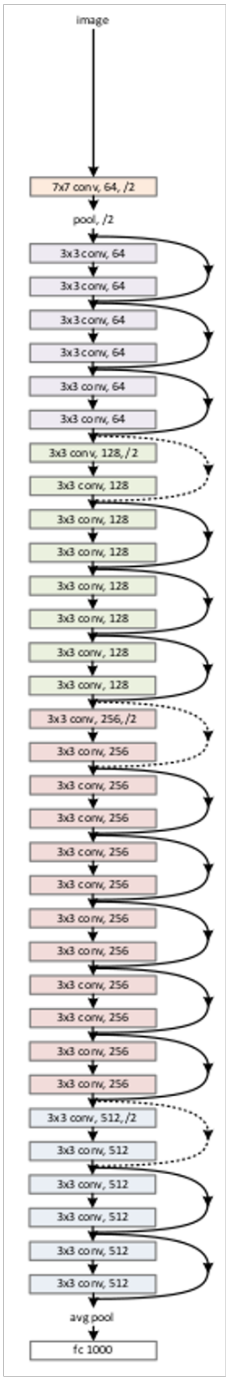


FIGURE 3.4: Resnet-34 architecture

3.4 Inception

The researchers from Google introduced the first Inception (InceptionV1) [7] neural network in 2014 during the ImageNet competition (See subsection 3.1.2). The model consisted of blocs called "inception cell" that was able to conduct convolutions using different scale filters and afterward aggregate the results as one. Thanks to 1x1 convolution which reduces the input channel depth the model saves computations. Using a set of 1x1, 3x3, and finally, 5x5 size of filters, an inception unit cell learns extracting features of different scale from the input image. Although inception cells use max-pooling operator, the dimension of a processed data is preserved due to "same" padding, and so the output is properly concatenated. A sample inception unit introduced with InceptionV1 is presented in Figure 3.5

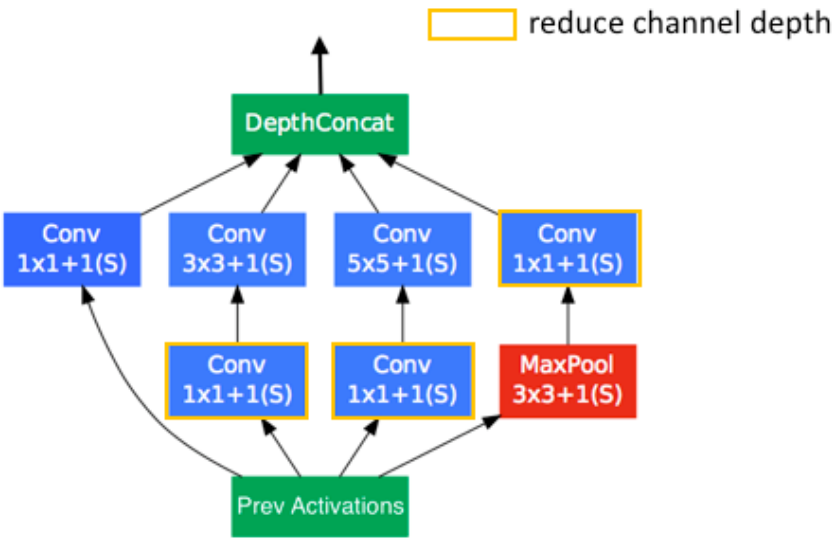


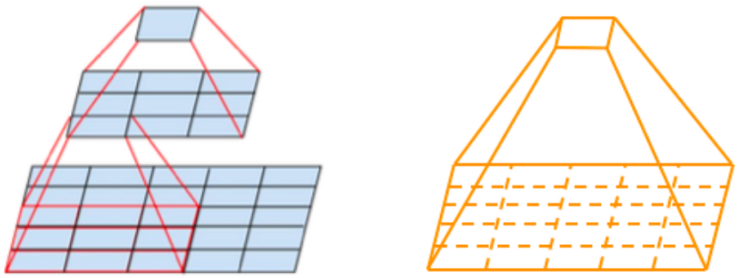
FIGURE 3.5: Inception cell introduced with the fist Inception model[7]

A follow-up paper was released not long after introducing a more efficient solution to the first version of the inception cell. Large filters sized 5x5, and

7x7 are useful in extensive spatial features extraction, yet their disadvantage lies in the number of parameters and therefore computational disproportion.

3.4.1 Factorizing Convolutions

The researchers from Google found a way to save computations and reduce number of parameters without dicreasing model's efficiency. In the proposed architecture [8] all 5x5 (Figure 3.6 (A)) convolutions were factorized to two consecutive 3x3 (Figure 3.6 (B)) operations, improving the computational speed. New inception unit is presented in Figure 3.7



(A) Double $3x3$ concolution.

(B) $5x5$ convolution.

FIGURE 3.6: $5x5$ convolution replaced by two $3x3$ convolutions.

By processing input through a layer with a 5x5 pixel filter, the number of parameters equals 25 ($5x5 = 25$). Whereas introducing two consecutive 3x3 pixel filters, the number of parameters decreases by 28% ($2x3x3 = 18$).

3.4.2 Factorization Into Asymmetric Convolutions

The researchers went even further with a decreasing number of filter parameters showing another double asymmetric convolutions 3x1 and 1x3 (see Figure 3.8) which deconstruct each 3x3 kernels.

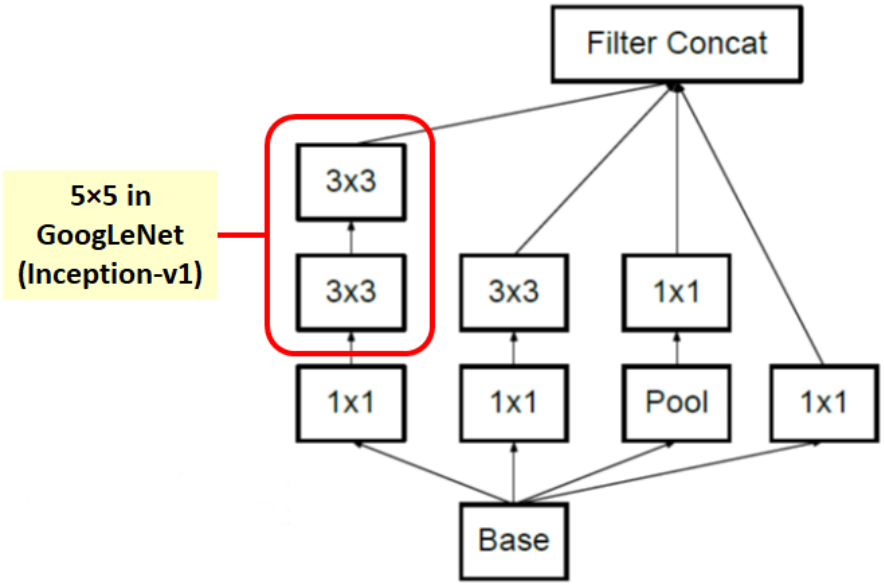


FIGURE 3.7: Inception cell introduced in [8]

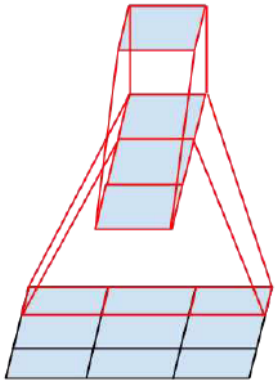


FIGURE 3.8: Asymmetric convolution [8]

Using deconstructed 3x3 filters the number of parameters decreases by roughly 33% since instead of 9 ($3x3 = 9$) we need only 6 ($2x3x1 = 6$). The application of asymmetric convolution can be seen in Figure 3.9. To achieve the results of a bloc presented in Figure 3.14, n needs to be set to 3.

The filter banks were furthermore expanded, making them wider, not

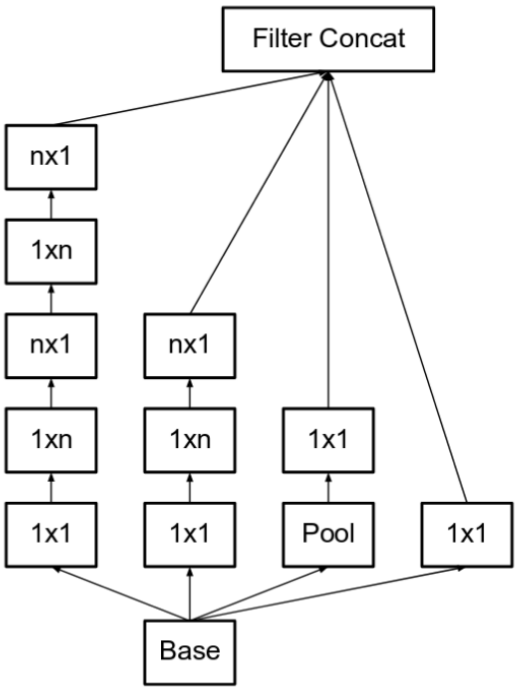


FIGURE 3.9: Inception cell using aymmetric convolutions[8]

deeper, which removed the representational bottleneck. Increasing the depth of a module decreases the dimensionality and introduces information loss. This is illustrated in Image 3.10

The suggested 3 different kinds of inception modules with factorization, drastically reduces the number of parameters in the whole network. Therefore, models incorporating such techniques are less prone to overfit and consequently can get deeper.

3.4.3 Auxiliary Classifiers

First auxiliary classifiers (see image 3.11) were proposed along with the InceptionV1 model [7]. Although the new models use the intuition behind them, they are slightly modified. Instead of using 2 auxiliary classifiers [7], only one is used on top of the 17x17 pixels layer (see Figure 3.14). The reason

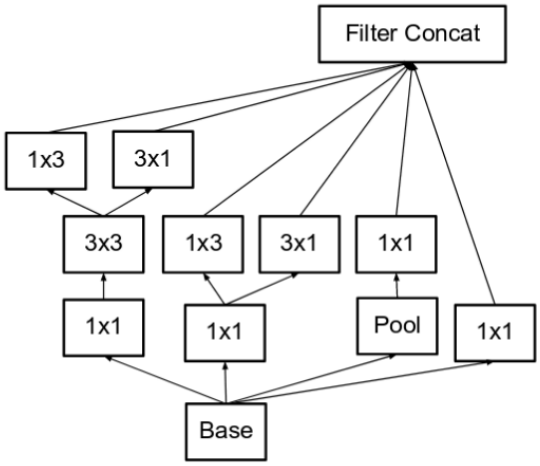


FIGURE 3.10: Wider inception module[8]

for introducing this difference lies in their purpose. The first version of the Inception deep neural network used auxiliary classifiers in order to make the model deeper. Here, for instance, one classifier serves as a regularizer.

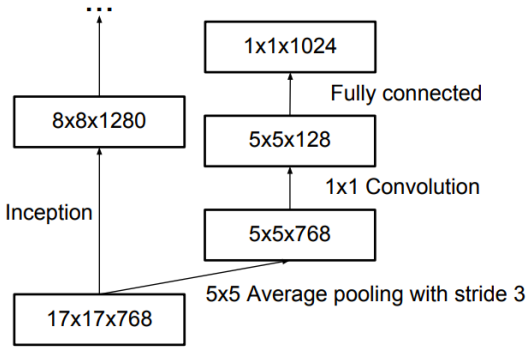


FIGURE 3.11: "Auxiliary classifier on top of the last 17x17 layer. Batch normalization of the layers in the side head results in a 0.4% absolute gain in top-1 accuracy. The lower axis shows the number of iterations performed, each with batch size 32." [8]

3.4.4 Effective Grid Size Reduction

The standard convention presented in AlexNet, VGG or ResNet uses a pooling operator (see Section 2.8) in order to downsize the feature map (See image

3.12). However, the drawback of this approach is either having a computationally expensive convolution operation followed by pooling or a greedy by max-pooling procedure proceeding with a convolution layer. Therefore, an alternative solution proposed effective grid size reduction.

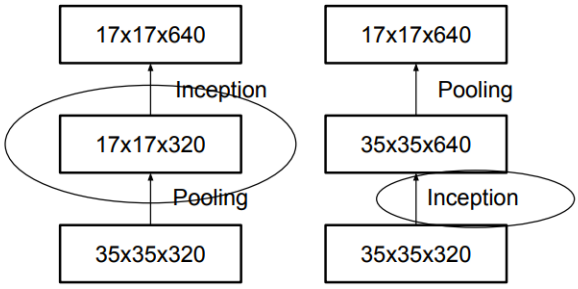


FIGURE 3.12: "Two alternative ways of reducing the grid size. The solution on the left violates principle 1 of not introducing a representational bottleneck from Section 2. The version on the right is 3 times more expensive computationally." [8]

The output from the effective grid size-reduction bloc is a concatenation of two sets of feature maps, together with having 640 channels. The first set having 320 feature maps is an output from a convolution bloc with stride equal 2. The second set constituting another 320 channels is obtained by max pooling.

The effective grid size reduction is an efficient operation, although less expensive.

The Figure 3.13 an inception module redicing grid-size.

3.4.5 Architecture

The InceptionV3 model [8] contains over 23 million parameters. The architecture can be divided into 5 modules, as presented in Figure 3.14. The

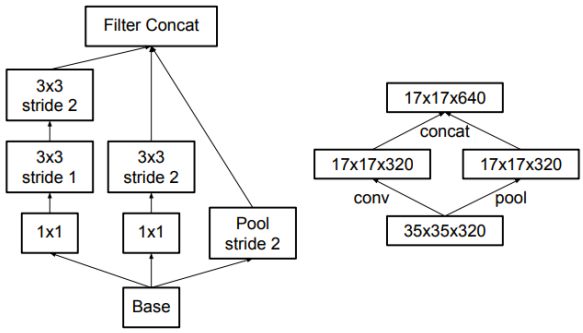


FIGURE 3.13: "Inception module that reduces the grid-size while expands the filter banks. It is both cheap and avoids the representational bottleneck as is suggested by principle 1. The diagram on the right represents the same solution but from the perspective of grid sizes rather than the operations." [8].

first processing block consists of 3 inception modules visualized in the image 3.6. Then, information is passed through the effective grid size reduction (see 3.4.4) and processed through four consecutive inception cells with asymmetric convolutions (see image 3.8). Moving forward, information flows to the 17x17 pixels convolution layer connected to an auxiliary classifier (see 3.4.3) and another effective grid size-reduction block. Finally, data progresses through a series of two blocs with wider filter banks (see image 3.10) and consequently gets to a fully-connected layer ended with a Softmax classifier.

Visualization of the network architecture can be found in Figure 3.14

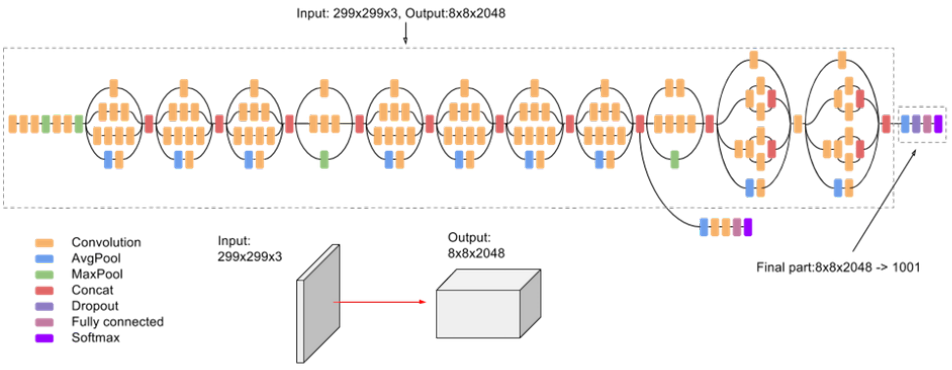


FIGURE 3.14: InceptionV3 architecture. Batch normalization and ReLU non-linearity are used after every convolution layer.)

3.5 Experiments

3.5.1 Dataset

The first part of the experiments compares three modified versions of neural networks introduced earlier; VGG16, ResNet-50 and InceptionV3 described in 3.2, 3.3 and 3.4, respectively. We train those models on the database containing X-Ray lungs images introduced in Section 2.5. All images 2.2 (918 samples, 306 per class) were resized to the same shape before training, 256x256 pixels. The dataset constituting of X-Ray images and one-hot encoded labels was partitioned into three different categories for training, validation, and testing, respectively. The training dataset consisted of 80% of randomly selected images. The validation set used 10% of all data, and the remaining 10% served for testing (the same approach was used in [4]), and its class distribution was kept in an even proportion, e.g., a third of samples were labeled as 'healthy', a third contained marks of tuberculosis and the remaining part came from patients suffering from pneumonia. During the training process, the input data is augmented [51] [52][53] by randomly selecting one of the following operations: rotation, brightness change, and random cropping (See section 2.6). The validation set used to control the accuracy and overfitting to training data is created independently in each training using Monte Carlo Cross Validation [54].

3.5.2 Models

The analysis in this chapter compares three models using different transfer learning described in 3.2, 3.3 and 3.4 for lung disease classification. The models were expanded with the same neural networks based classifier consisting

of a global average pooling layer, three fully connected layers having 1024, 512 and 256 neurons and a softmax classifier. The number of trainable parameters in the deep neural networks is 1,182,211 for VGG16 and 2,755,075 for both ResNet50 and InceptionV3. Before passing input images through deep neural networks which serve as feature extractors, the batches were adjusted to the same formats (batch size, input scale, etc.) the mentioned models were trained with [43][45][8]. Furthermore, our neural network-based classifiers learn bias and weights parameters by backpropagating the error to minimize the categorical cross-entropy using Adam [55] optimizer at training time.

The models were later on validated using different sets of hyper-parameters (only appended layers as pre-trained networks remain frozen) observed during the training process. The output from the softmax classifier is a vector of probabilities with which an input image belongs to one of the classes. The final class is the one corresponding to the highest value, and its position is then mapped back to a corresponding class.

The code for the transfer-learning models is publicly available through a python API, Keras. Our algorithms were trained on servers equipped with GPU provided by Helios Calcul Québec, which consists of fifteen computing nodes each having eight Nvidia K20 GPUs and additionally six computing nodes with eight nVidia K80 boards each. Every K80 board includes two GPU's and so the total of 216 GPU's in the cluster.

3.6 Results and analysis

Training three models repeatedly ten times for 150 epochs took roughly one day. This relatively short period is caused by setting parameters of pre-trained networks to non-trainable, and thus the gradient caused by misclassification flows only through the appended layers. The initial image preprocessing was also advantageous for the duration since the real size images of

hundreds of thousands or million pixels were initially downsized to a fixed format. The biggest problem with training was related to the maximum platform usage time, which is up to twelve hours.

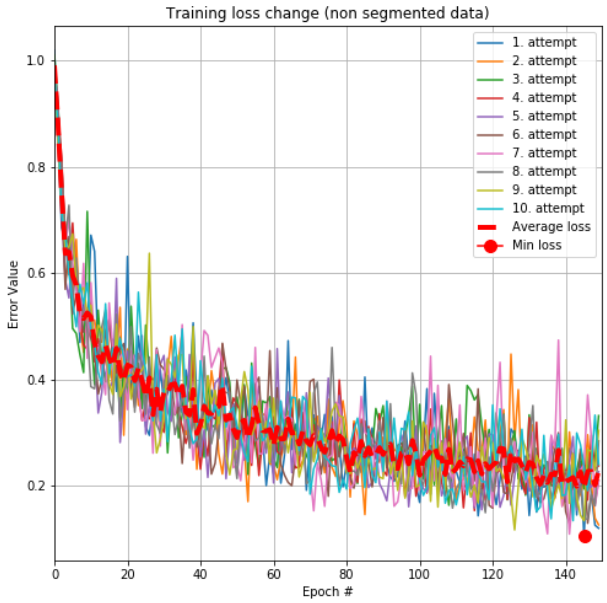
The examination of error calculated on a validation set allowed us to find a relatively good epoch after which models were overfitting training data. The training process was then stopped, and the final results were measured as an average of all results obtained at that step. The last step was to show the performance of selected models on the unseen data (test set).

3.6.1 VGG results

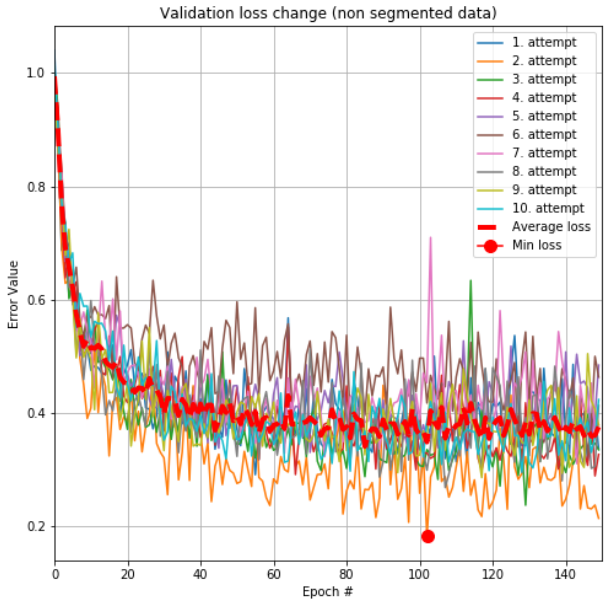
The following results were generated for ten independent training runs to observe a similar training pattern. Each of the ten training and validation curves (see Figures 3.15 and 3.16) were plotted on the same charts based on the tape (training or validation). To maintain a high level of readability, the results were separated. The wider, dotted curve is an averaged result of all obtained at the particular epoch. The red dot in Figures in 3.15 represent the lowest loss value on training and validation data sets, whereas in 3.16 it corresponds to the maximum accuracy obtained.

Figure 3.15 shows that the model slowly starts overfitting on the training dataset around the 90th epoch yet then the validation error falls again and eventually after 150 epoch achieves the best average results. Similar behavior is experienced when examining Figure 3.16. Here, the average validation accuracy slows down but then slightly increases.

Eventually, the models were evaluated on the test set and scored an average accuracy of 63.85% In order to visualize results, we selected a network which obtained the best accuracy score. We see on the confusion matrix in Figure 3.17 A) that the model had the biggest problems with classifying 'tuberculosis images' to the corresponding class and tend to mistake it as

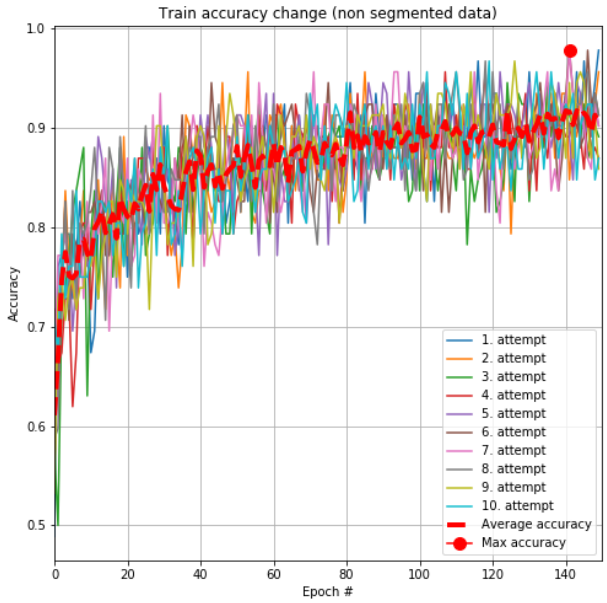


(A) Training error change

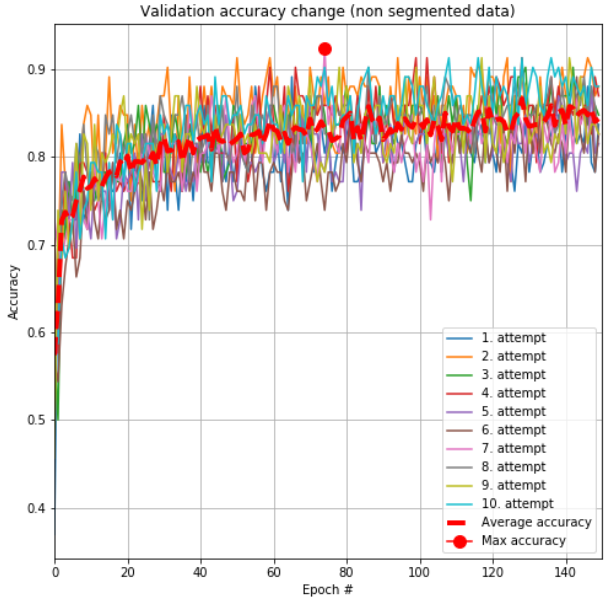


(B) Validation error change

FIGURE 3.15: VGG16 based model training and validation loss change.



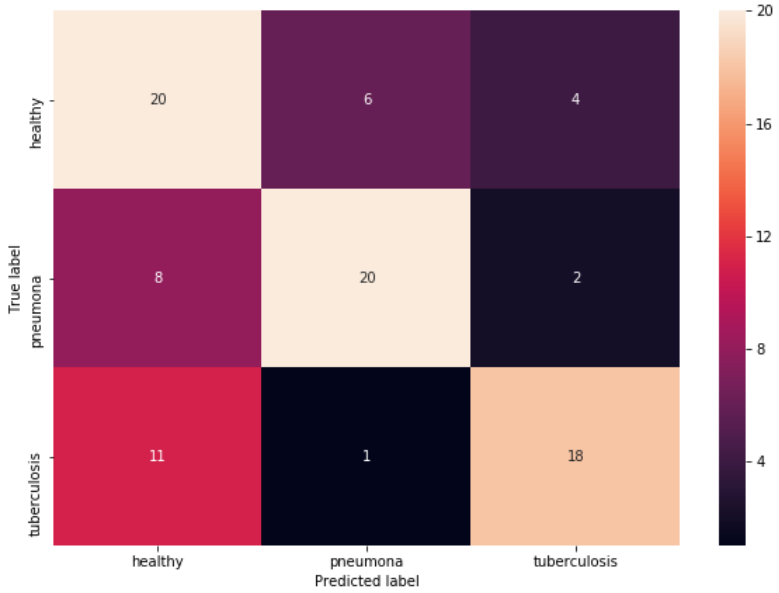
(A) Training accuracy change



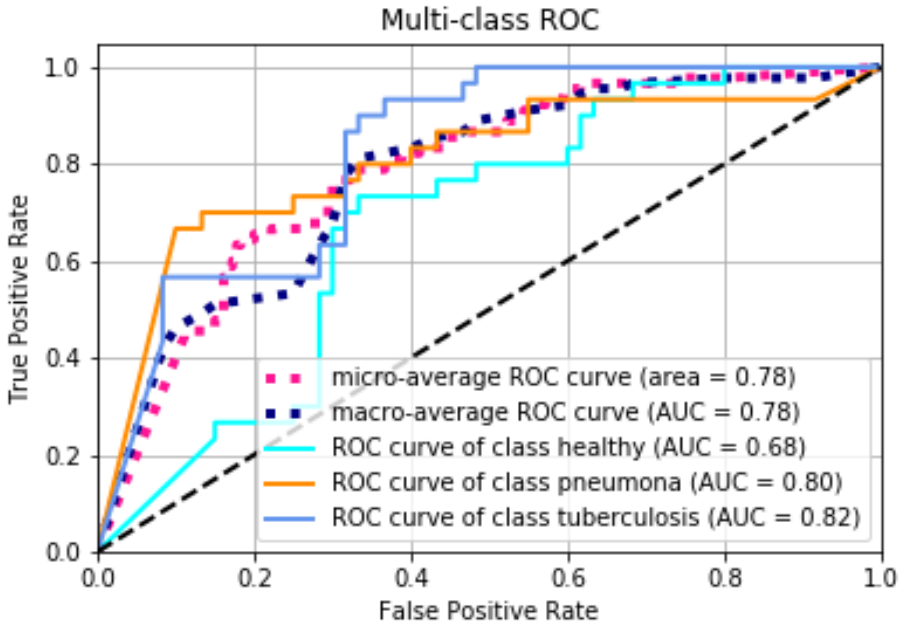
(B) Validation accuracy change

FIGURE 3.16: VGG16 based model training and validation accuracy change.

healthy. Fortunately, the majority of samples containing marks of tuberculosis were correctly classified. The image B) in Figure 3.17 shows that AUC score for classes healthy, tuberculosis and pneumonia were equal 0.68, 0.82 and 0.80, respectively. Additionally, we present sample classification results in Figure 3.18. Although both images A) and C) were correctly classified, their corresponding class activation maps (images B) and D)) show that their determinative regions were not related to the problem. Instead of investigating lung regions, our trained model focused on waist curves, armpits, or collarbones.

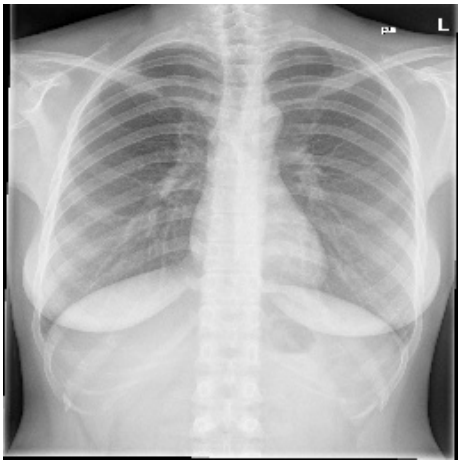


(A) Confusion matrix

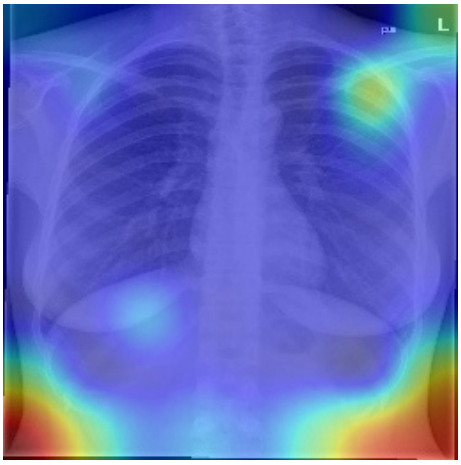


(B) Per class ROC curves

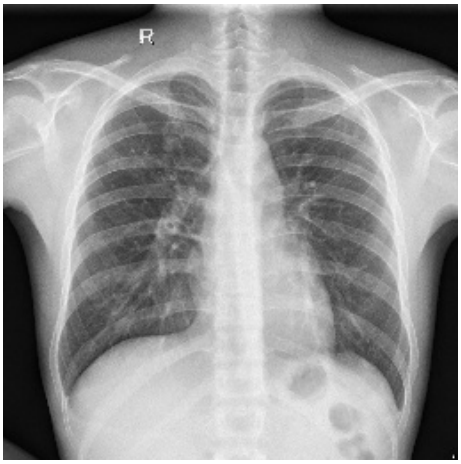
FIGURE 3.17: Image A shows the confusion matrix of the obtained results with the model which reached the highest accuracy during the training. Image B show its per class ROC curves.



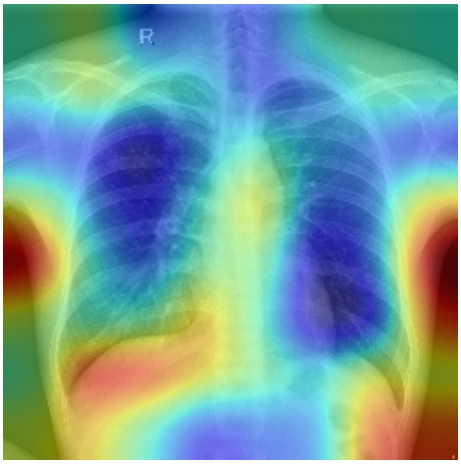
(A) Correctly classified image as 'tuberculosis'



(B) Class activation map



(C) Correctly classified image as 'pneumonia'



(D) Class activation map

FIGURE 3.18: Two pairs of correctly labelled images containing marks of tuberculosis and pneumonia with their class activation maps.

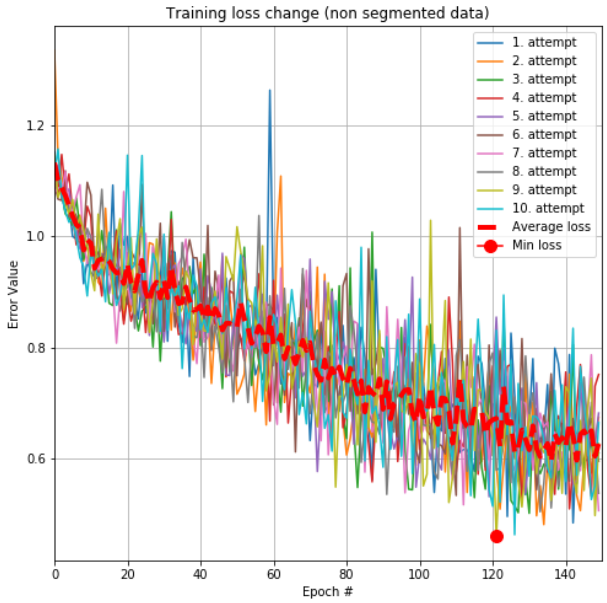
3.6.2 ResNet results

The following results were generated for ten independent training runs in order to observe a similar pattern in model behavior during training. All the results achieved during training and validation processes were plotted in Figures 3.19 and 3.20. By splitting the results by type (separately training and validation), we maintained a high level of visibility, allowing us to simplify

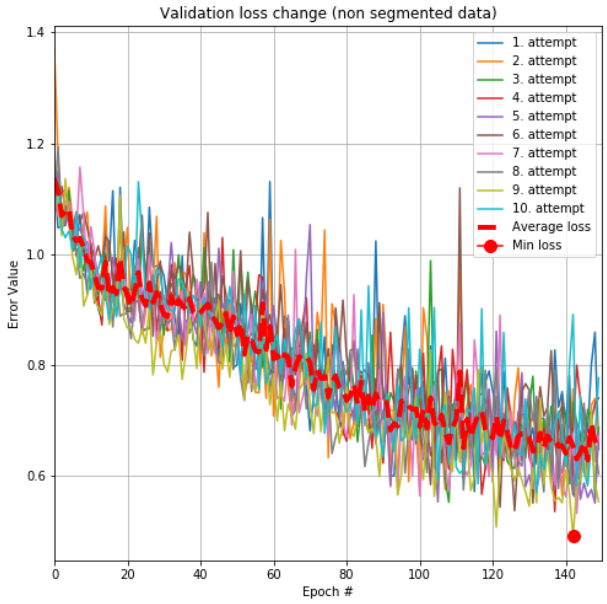
the analysis. The wider, dotted curve is an average of all results obtained at the corresponding epoch. The red dot in Figures 3.19 and 3.20 stands for minimum loss and maximum accuracy value, respectively. The Figure 3.19 shows that the model decreases its loss throughout the whole training yet seems to reach the validation accuracy plateau around 120th epoch (see Figure 3.20 B)).

Finally, after 150 epochs, all the ResNet-50 based models were evaluated on the test set and scored an average accuracy of 72.22%, which is almost a ten points improvement to 3.6.1. In order to visualize the results, we selected a network which achieved the best accuracy score. The confusion matrix in Figure 3.21 A) shows that, similarly to 3.6.1, the model had problems with correctly labelling 'tuberculosis images' although the majority of images were correctly classified. Image B) shows that the AUC score for healthy, tuberculosis and pneumonia were equal 0.84, 0.76, and 0.84, respectively. This is an improvement in comparison to results in the previous subsection (3.6.1).

Additionally, we present sample classification results in Figure 3.22. Similarly to 3.6.1, both images A) and C) were correctly classified yet the overlapping class activation maps (images B) and D)) show that the determinative regions were not related to lungs. Areas like collarbones or heart decided of the final label.

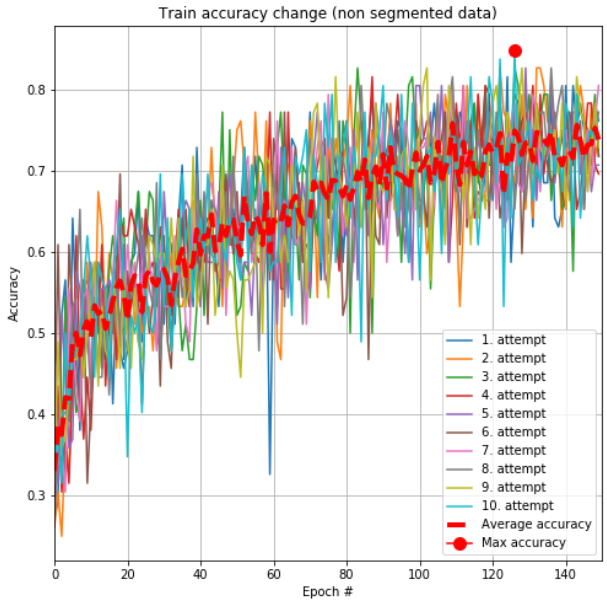


(A) Training error change

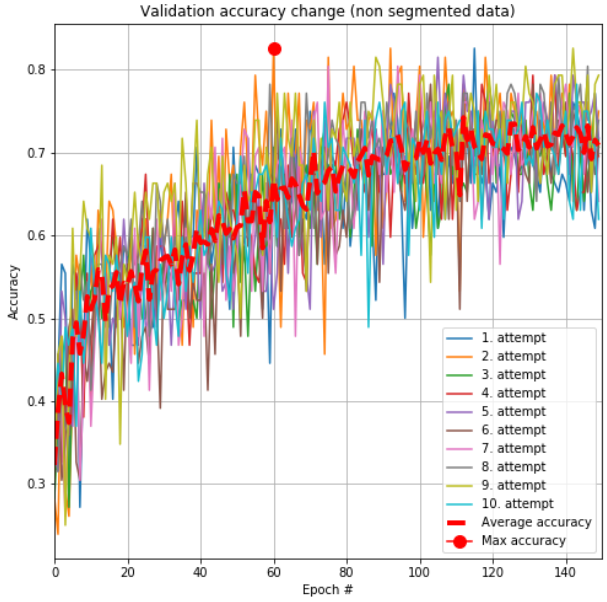


(B) Validation error change

FIGURE 3.19: ResNet-50 based model training and validation loss change.

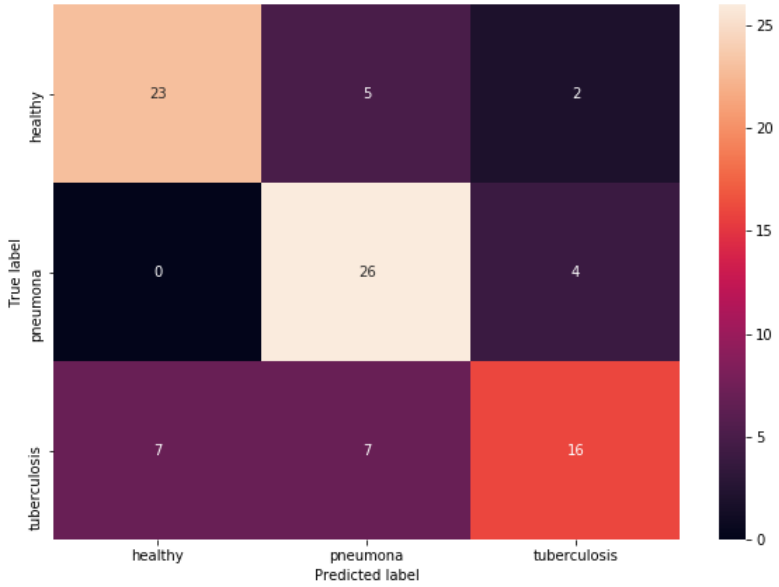


(A) Training accuracy change

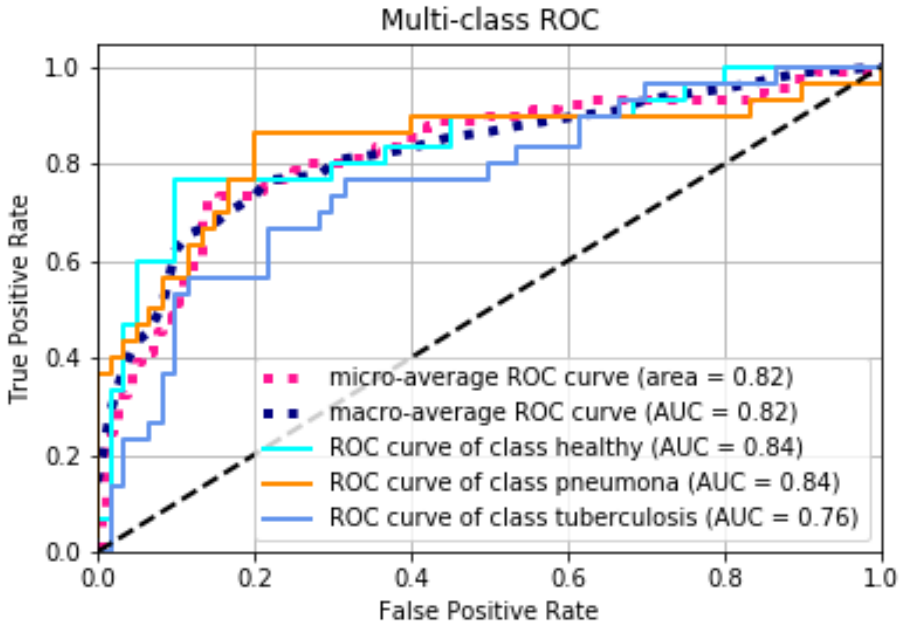


(B) Validation accuracy change

FIGURE 3.20: ResNet-50 based model training and validation accuracy change.



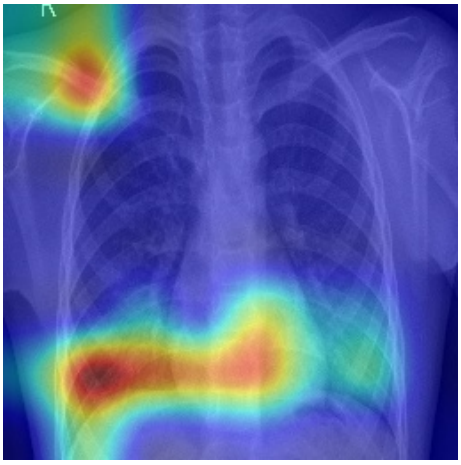
(A) Confusion matrix



(B) Per class ROC curves

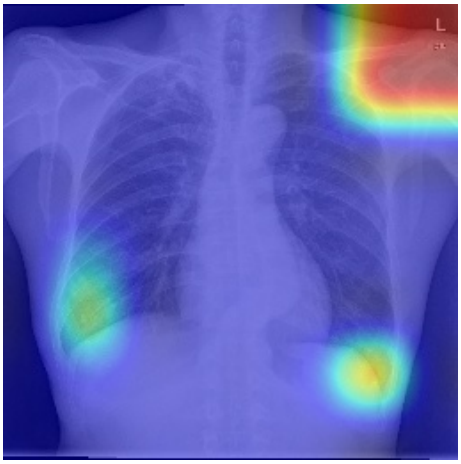
FIGURE 3.21: Image A shows the confusion matrix of the obtained results with the model which reached the highest accuracy during the training. Image B show its per class ROC curves.

(A) Correctly classified image as 'tuberculosis'



(B) Class activation map

(C) Correctly classified image as 'pneumonia'



(D) Class activation map

FIGURE 3.22: Two pairs of correctly labelled images containing marks of tuberculosis and pneumonia with their class activation maps.

3.6.3 Inception results

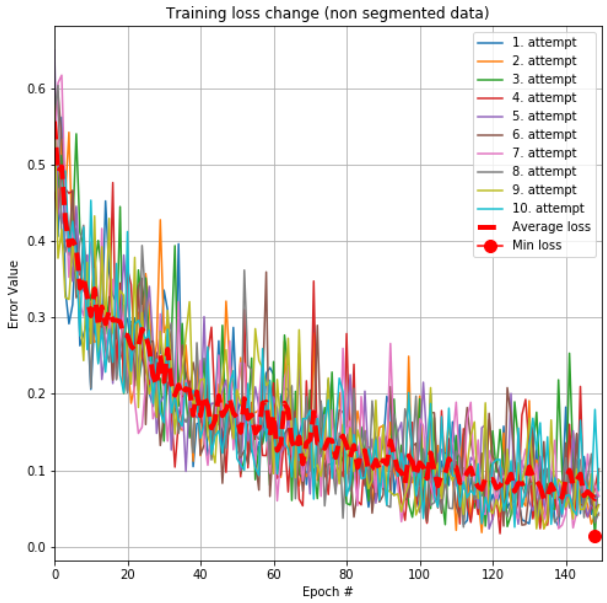
The following results were generated for ten independent training runs in order to observe a similar training pattern. Each of the ten training and validation curves (see Figures 3.23 and 3.24) were plotted on the same charts based on the tape (training or validation). The same as before, we separate the results to maintain a high level of readability. The wider, dotted curve

is an averaged result of all ten runs at the particular epoch. The red dot in Figures 3.23 3.24 stand for the lowest loss value on training and validation data sets the maximum accuracy obtained, respectively.

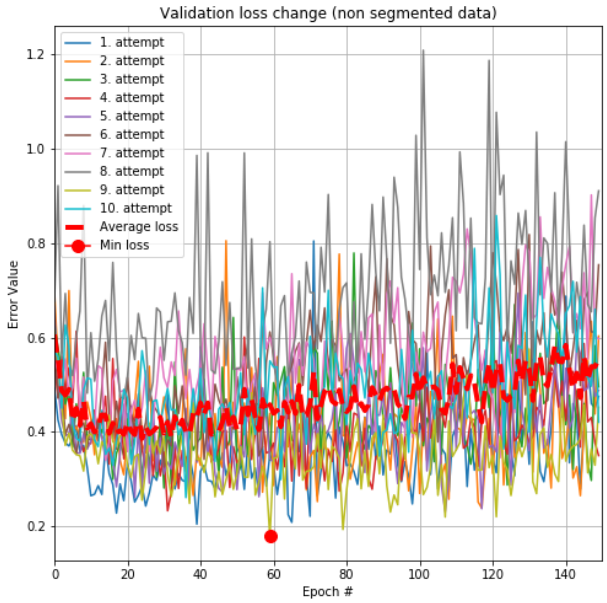
The image 3.23 shows that the model starts overfitting on the training dataset around the 20th epoch which can be witnessed by examining both validation error and accuracy change. The validation error curve slowly increases its value, whereas the accuracy level remains similar (see Figure 3.24).

Finally, we took all InceptionV3 based models at the 20th epoch and evaluated them on the test set and scored an average accuracy of 80.55%, which is a significant improvement comparing to 3.6.1 and 3.6.2. To visualize the final results, similarly to the previous chapter, we selected a model which achieved the best accuracy score after the 20th epoch. The confusion matrix in Figure 3.25 A) shows that the new model improved the number of true positives (TP) in all classes comparing to previous algorithms, although the network still faced minor problems with classifying all 'tuberculosis images' to their corresponding label. Image B) shows that AUC score for healthy, tuberculosis and pneumonia were equal 0.89, 0.87, and 0.92, respectively. This is an improvement in comparison to results in the previous subsections (3.6.1).

Additionally, we present sample classification results in Figure 3.26. Similarly to previous subsections (3.6.1 and 3.6.2), both images A) and C) were correctly classified yet the overlapping class activation maps (images B) and D)) show that the determinative regions were not necessarily related to lungs. Areas like armpits and internal organs decided on the final label.

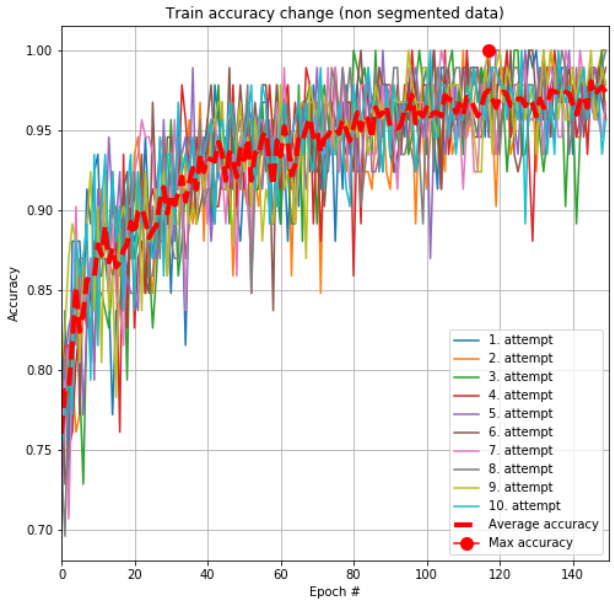


(A) Training error change

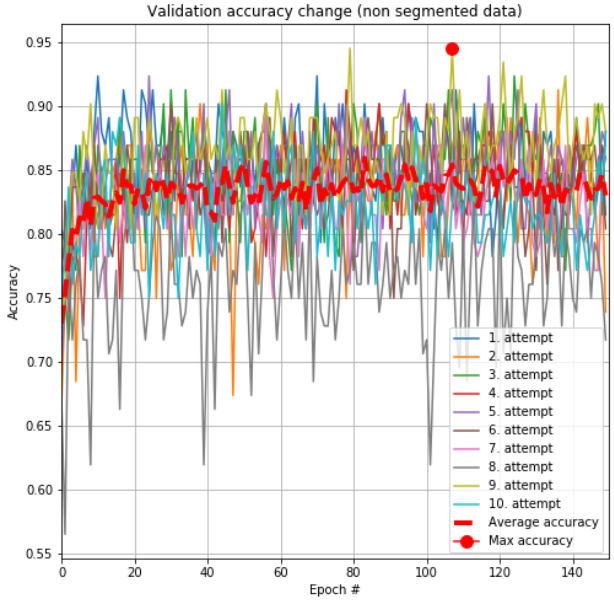


(B) Validation error change

FIGURE 3.23: InceptionV3 based model training and validation loss change.

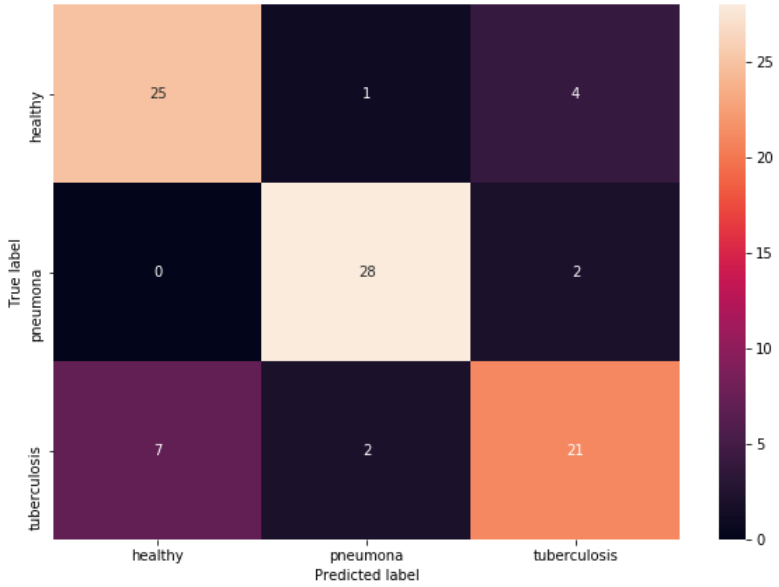


(A) Training accuracy change

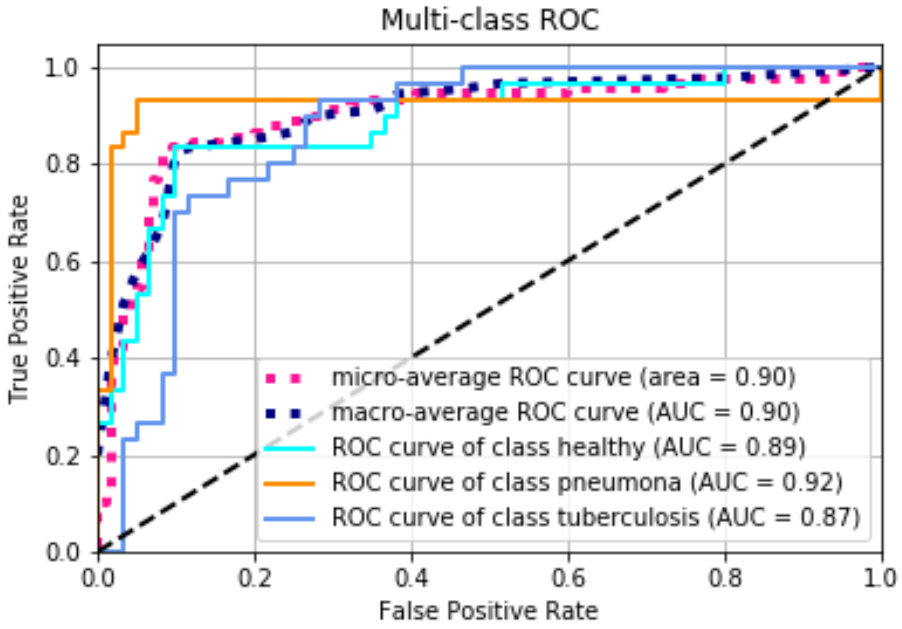


(B) Validation accuracy change

FIGURE 3.24: InceptionV3 based model training and validation accuracy change.



(A) Confusion matrix



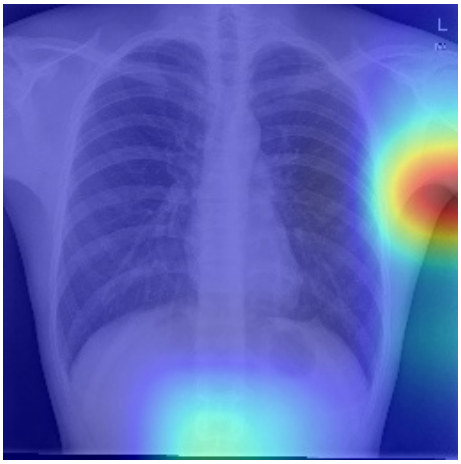
(B) Per class ROC curves

FIGURE 3.25: Image A shows the confusion matrix of the obtained results with the model which reached the highest accuracy during the training. Image B show its per class ROC curves.

(A) Correctly classified image as 'tuberculosis'

(B) Class activation map

(C) Correctly classified image as 'pneumonia'



(D) Class activation map

FIGURE 3.26: Two pairs of correctly labelled images containing marks of tuberculosis and pneumonia with their class activation maps.

3.6.4 Results comparison

After comparing the results obtained in sections 3.6.1, 3.6.2 and 3.6.3 we can observe that transfer-learning models perform decently in lung diseases classification tasks even when the data resources are limited. Table 3.1 shows the comparison of all trained algorithms. The InceptionV3 based model scored the best, reaching better accuracy than VGG16 algorithms by over 16 points.

What is more, we see major improvement in AUC, F1 score, precission and sensitivity. The most important results are related to the "infected" classes and here we see that using InceptionV3 model a random instance containing marks of either tuberculosis or pneumonia has a high probability to be assigned to one of those classes. The interpretability of our methods is not certain because the generated class activation maps show that our models focused on regions outside the lungs area - hips or collarbones. However, we can clearly state that using transfer-learning based algorithms on small datasets allows achieving competitive classification scores on the unseen data. There was no work done with a similar dataset (Chest X-Ray multiclassification problem with small dataset) therefore we did not compare ourselves with available solutions.

	VGG16	ResNet-50	InceptionV3
Accuracy	0.64	0.72	**0.81**
AUC (healthy)	0.68	0.84	**0.89**
AUC (pneumonia)	0.80	0.84	**0.92**
AUC (tuberculosis)	0.82	0.76	**0.87**
F1 score (healthy)	0.55	0.66	**0.76**
F1 score (pneumonia)	0.64	0.82	**0.90**
F1 score (tuberculosis)	0.72	0.66	**0.75**
precision (healthy)	0.49	0.77	**0.80**
precision (pneumonia)	0.87	0.76	**0.89**
precision (tuberculosis)	0.71	0.67	**0.74**
sensitivity (healthy)	0.62	0.60	**0.73**
sensitivity (pneumonia)	0.53	0.88	**0.92**
sensitivity (tuberculosis)	**0.77**	0.68	**0.77**

TABLE 3.1: Comparison of transfer-learning based algorithms in terms of accuracy, AUC, F1 score, precision and sensitivity for healthy, pneumonia and tuberculosis classes.

3.7 Summary

This chapter introduces three models which achieved the highest scores in the ImageNet competition; VGG16, ResNet-50, and InceptionV3. We also

present our initial work in lung diseases classification using those pre-trained deep neural networks as feature extractors for a simple 3-layers deep neural network. The significant and promising results on small datasets show that there is no need to build sophisticated, multiple-layers networks in order to achieve high scores on the test dataset.

Chapter 4

Transfer Learning Models Accuracy Improvement in Lung Diseases Classification Using Segmentated X-Ray Images

4.1 U-Net - Image Segmentation using Deep Neural Networks

Many vision-related tasks, especially those from the field of medical image processing, expect to have a class assigned per pixel, i.e., every pixel is associated with a corresponding class. To conduct this process, we propose a neural network architecture described in [9] and showed in Figure 4.1.

This model works well, as proven in [9] and [4], with very few training image examples yielding precise segmentation. The thought behind this network is to utilize progressive layers instead of a building system, where upsampling layers are utilized as instead of pooling operators. Therefore, increasing the output resolution.

High-resolution features are combined with the upsampled output to do localize, as presented in Figure 4.1. The deconvolution layers consist of a high

number of kernels, which better propagate information and result in higher resolution output. Thanks to the described procedures, the deconvolution path is approximately symmetric to the contracting one, and so the architecture resembles a u-shape. There are no fully connected layers, therefore, making it possible to conduct the seamless segmentation of relatively large images, extrapolating the missing context by mirroring processed input.

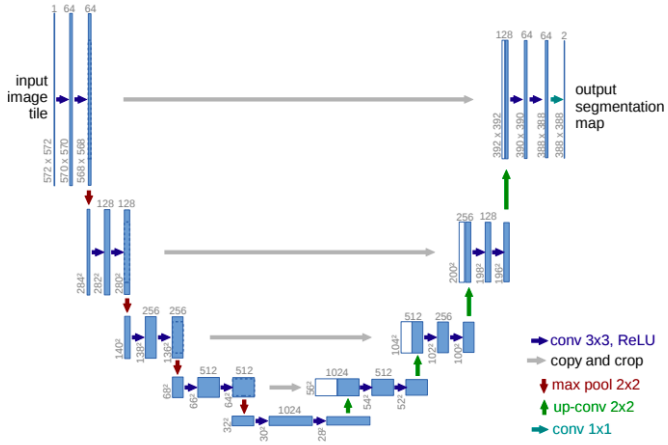


FIGURE 4.1: "U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations."[9]

4.1.1 Architecture

The network architecture showed in Figure 4.1 consists of an expansive path (right) and a contracting one(left). The first part (contacting)resembles a typical convolutional neural network; the repeated 3x3 convolutions followed by a non-linearity, here rectified linear unit (ReLU), and 2x2 poling with stride 2. Each downsampling operation doubles the number of resulting feature maps.

All expansive path operations are made of upsampling of the feature channels followed by a 2x2 deconvolution (or "up-convolution") which reduces

the number of feature maps twice. The result is then concatenated with the corresponding feature layer from the contracting path and convolved with 3x3 kernels, and each passed through a ReLU. The final layers apply a 1x1 convolution to map each feature vector to the desired class.

4.2 Lungs Segmentation

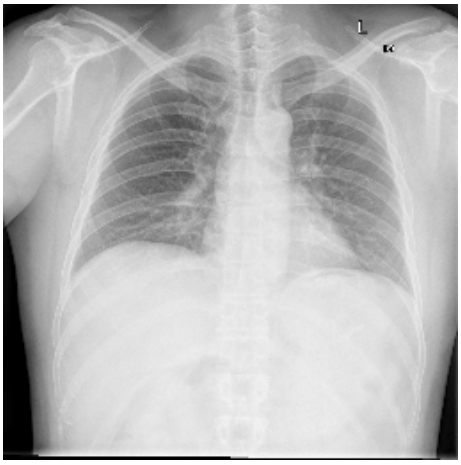
Following the approach presented by previous works summarized in Chapter 2, we wanted to use deep convolutional neural networks to segment lungs [56] before processing it through the classification models mentioned in 3.5.2. Researchers in [56] indicate that U-Net architecture and its modifications outperform the majority of CNN based models and achieve excellent results by easily capturing spacial information about the lungs. As an outcome, we propose a pipeline that consists of two stages; first segmentation and then classification.

4.2.1 Dataset

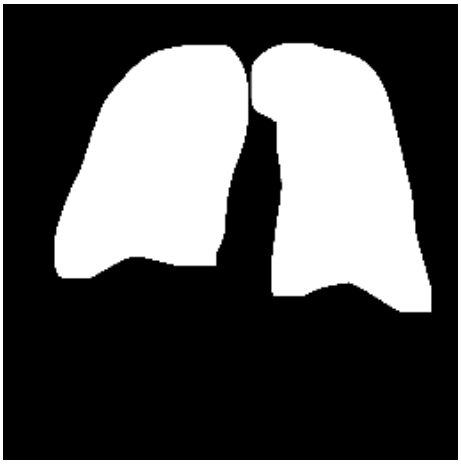
The phase of extracting valuable information (lungs) is conducted with a model presented in 4.1. Our algorithms trained for 500 epochs on an extension of the SH dataset described in 2.5. The input to our u-shaped deep neural network is a regular Chest X-Ray image, whereas the output is a manually prepared binary mask of lung shape, matching the input. Figure 4.2 A) presents an X-Ray image and B) its corresponding mask.

4.2.2 Training

As mentioned before, our model was trained for 500 epochs using a dataset divided into 80%, 10%, and 10% parts, for training, validation and test parts respectively on the same machines introduced in section 3.5.2 using the batch size o 8 samples, augmentation techniques briefed in subsection 2.6, Adam optimizer [55] and categorical cross-entropy as a loss function for pixel-wise binary classification. The training results are visible in Figure 4.3. As we can easily notice, the validation error is slowly falling throughout the whole training, whereas there is no major change after the 100th epoch. The final



(A) X-Ray image



(B) Corresponding lungs mask

FIGURE 4.2: An X-Ray image and its corresponding lungs mask.

error on the validation set is right below 0.05 and slightly above 0.06 on the test set.

4.2.3 Results

Our algorithm learns shape-related features typical for lungs and can generalize well further over unseen data. Figure 4.4 shows the results of our U-Net trained models.

It is clear that the network was able to learn chest shape features and exclude regions containing internal organs such as the heart. The incredibly promising results allowed us to process the whole dataset presented in subsection 3.5.1 and continue our analysis on the newly processed images.

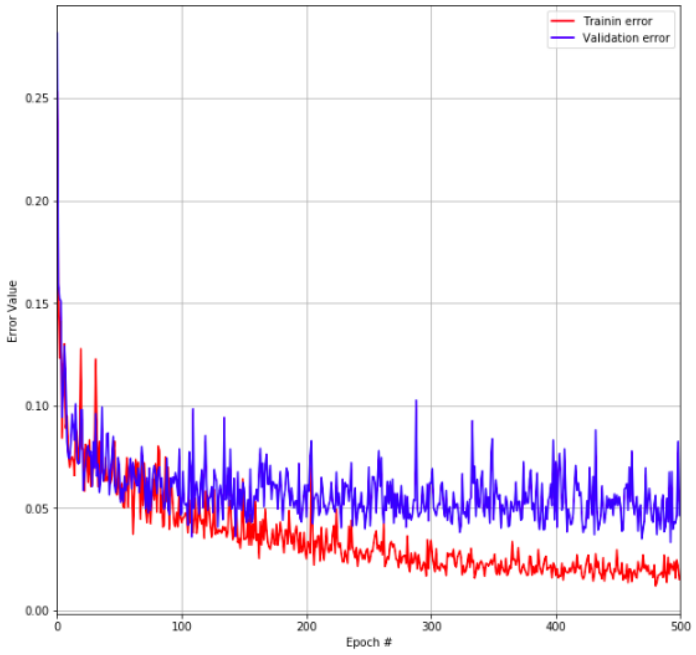
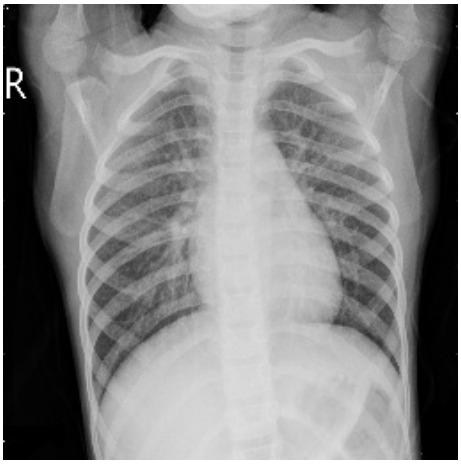
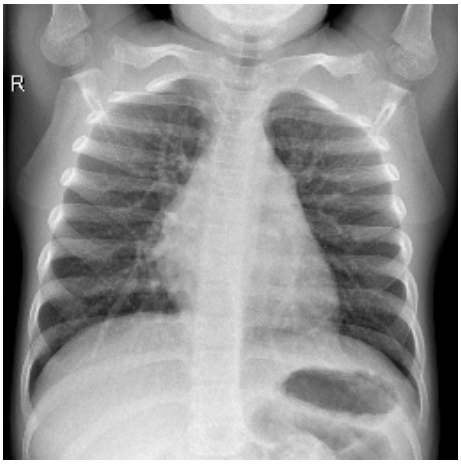


FIGURE 4.3: U-Net training and validation losses change during training.



(A) X-Ray image

(B) Segmented lungs



(C) X-Ray image

(D) Segmented lungs

(E) X-Ray image

(F) Segmented lungs

FIGURE 4.4: Three pairs of CXR images with corresponding, segmented (extracted) lungs.

4.3 Training Deep Learning Models On Segmented Images

As disclosed in the prologue to this section, we propose a two-stage pipeline consisting of Chest X-Ray images segmentation and lung disease classification. The first phase (segmentation) is conducted based on experiments presented in the previous section 4.2. The second stage is to train the models presented in the previous chapter 4 to investigate the potential improvement in performance. Our classification models were trained using the same setup as shown in 3.5.2

4.3.1 Dataset

Here, we conduct our experiments using the same data as in chapter 3. The difference is in previous segmentation, which extracts valuable for the task information - lungs. Figure 4.4 shows the training samples; the left and right column correspond to input and output, respectively.

4.4 Results and analysis

As in the previous chapter, training all three models repeatedly ten times for 150 epochs took about one day. This short time is a result of setting all parameters of pre-trained models as non-trainable. Therefore the gradients flow only through the concatenated layers. The segmented images were resized, which also beneficially influences the execution duration. The biggest problem related to training our models was the maximum platform usage time, which is up to twelve hours.

The examination of training and validation error curves allowed us to find a relatively good number of epochs after which models were overfitting

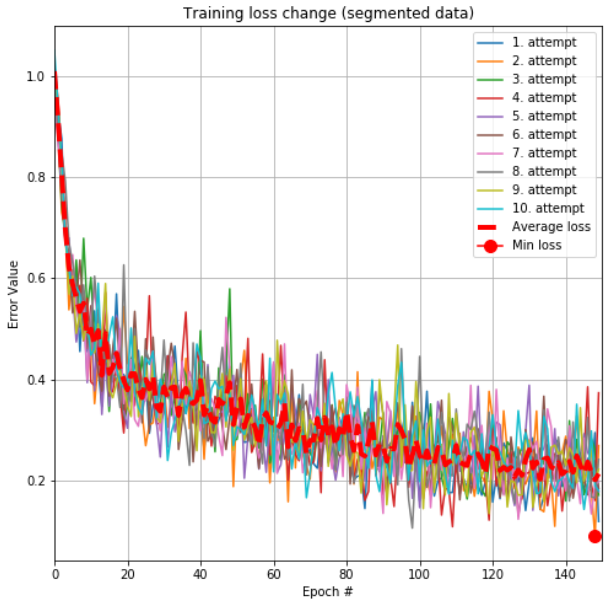
the training data. The training process was then stopped, and the final results were measured as an average of all results obtained at that step.

4.4.1 VGG results

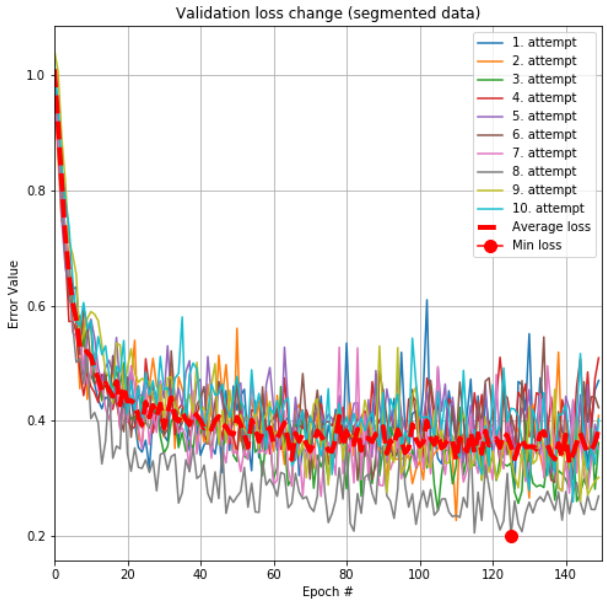
The following results were generated for ten independent training runs to observe a similar training pattern. Each of the ten training and validation curves (see Figures 4.5 and 4.6) were plotted on the same charts based on the tape (training or validation). So as to keep up a high state of readability, the results were isolated. The wider, dotted curve is an averaged result of all obtained at the particular epoch. The red dot on Figures in 4.5 represent the lowest loss value on training and validation data sets, whereas on 4.6 it corresponds to the maximum accuracy obtained.

The image 4.5 shows that the model stops dropping the validation error around the 70th epoch and maintains it's level throughout the remaining roughly 80 epochs. A similar behavior is experienced when examining Figure 4.6. Here, the average validation accuracy slows down and only slightly increases.

Eventually, the models were evaluated on the test set and scored an average accuracy of 69.99%, which is over 6 percentage of improvement comparing to the results received in 3.6.1. To visualize results, we selected a network which obtained the best accuracy score after 150 epochs. The confusion matrix in Figure 4.7 A) shows that the model had the biggest problems with classifying 'pneumonia images' to the corresponding class and tend mistaken it as healthy. The other problem is related to labeling healthy images as healthy, although in this case, it is not a major issue since the cost of assigning healthy patients to a category of sick ones is more acceptable than otherwise. Additionally, the majority of samples containing marks of pneumonia were correctly classified. The image B) on Figure 4.7 shows that AUC score for classes

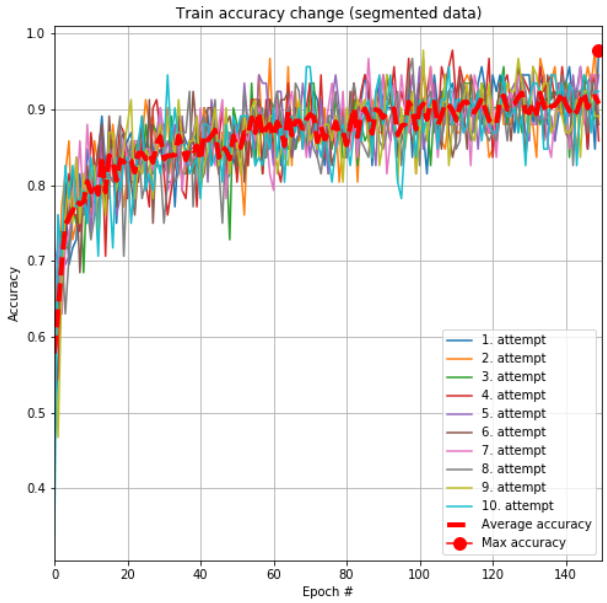


(A) Training error change

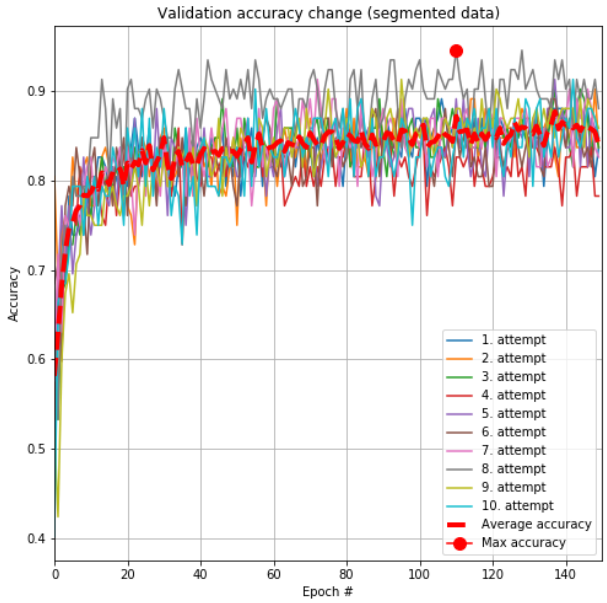


(B) Validation error change

FIGURE 4.5: VGG16 based model training and validation loss change.



(A) Training accuracy change

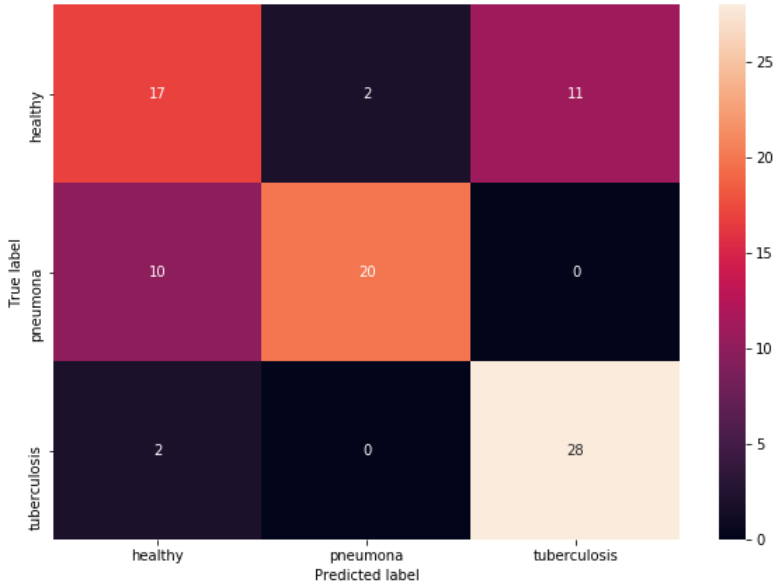


(B) Validation accuracy change

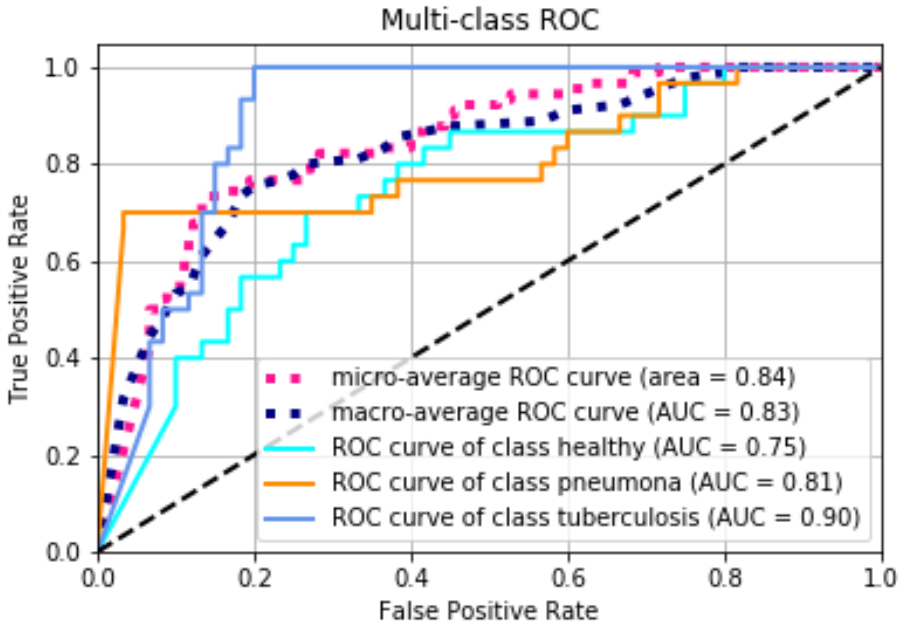
FIGURE 4.6: VGG16 based model training and validation accuracy change.

healthy, pneumonia and tuberculosis were equal 0.75, 0.81 and 0.90, respectively. This is also an improvement with the results received in 3.6.1.

Additionally, we present sample classification results on the Figure 4.8. Both images A) and C) were correctly classified and corresponding class activation maps (images B) and D)) show that their determinative regions were related to the problem, unlike in 3.6.1. The network investigated lungs regions and made the final decision based on extracted features.



(A) Confusion matrix



(B) Per class ROC curves

FIGURE 4.7: Image A shows the confusion matrix of the obtained results with the model which reached the highest accuracy during the training. Image B show its per class ROC curves.

(A) Correctly classified image as 'tuberculosis'

(B) Class activation map

(C) Correctly classified image as 'pneumonia'

(D) Class activation map

FIGURE 4.8: Two pairs of correctly labelled images containing marks of tuberculosis and pneumonia with their class activation maps.

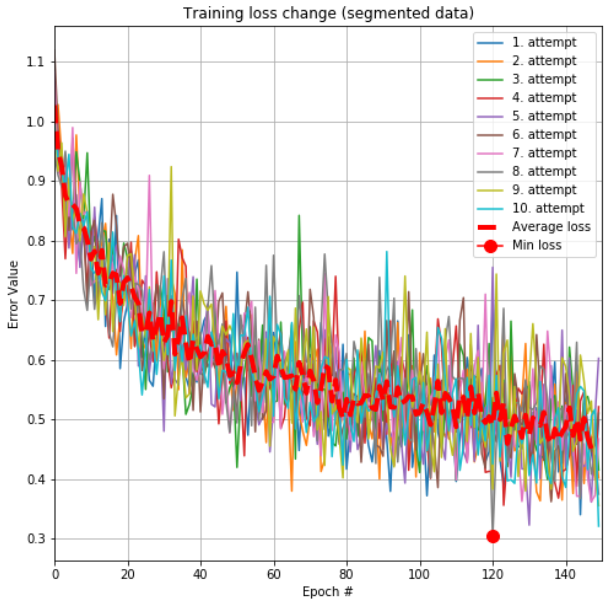
4.4.2 ResNet results

The following results were generated for ten independent training runs in order to observe a similar pattern in model behavior during training. All the results achieved during training and validation processes were plotted in Figures 4.9 and 4.10. By splitting the results by type (separately training and validation), we maintained a high level of visibility, allowing to simplify the

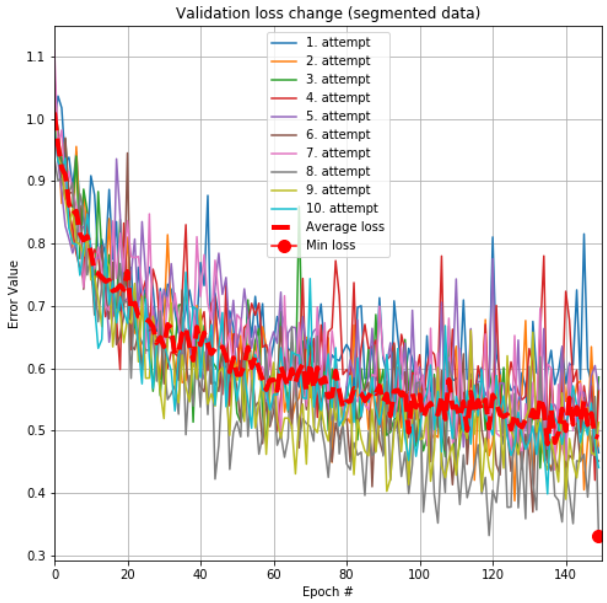
analysis. The wider, dotted curve is an average of all results obtained at corresponding epoch. The red dot on Figures 4.9 and 4.10 stands for minimum loss and maximum accuracy value, respectively. The image 4.9 shows that the model decreases its loss throughout the whole training.

Finally, after 150 epochs all the ResNet-50 based models were evaluated on the test set and scored an average accuracy of 74.99%, which is almost a 5 percent improvement to 4.4.1 and over two to 3.6.2. In order to visualize the results, we selected a network which achieved the best accuracy score. The confusion matrix in Figure 4.11 A) shows that, similarly to 3.6.2, the model had problems with correctly labelling 'tuberculosis images' although the majority of images were correctly classified. Image B) shows that AUC score for healthy, pneumonia and tuberculosis were equal 0.77, 0.91, and 0.82, respectively. Comparing to the results in the previous subsection 3.6.1, we were able to improve the results for healthy and pneumonia images classification. When looking at AUC scores in 3.6.2, we scored better in labeling pneumonia.

Additionally, we present sample classification results on the Figure 4.12. Similarly to 4.4.1, both images A) and C) were correctly classified and the overlapping class activation maps (images B) and D)) show that the determinative regions were related to lungs.

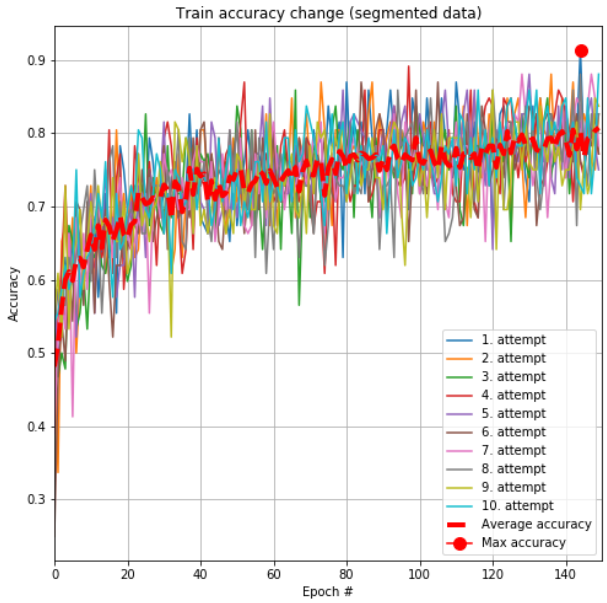


(A) Training error change

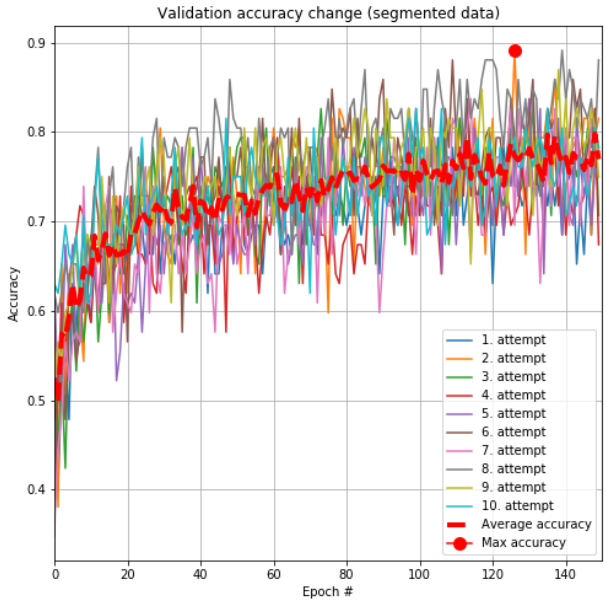


(B) Validation error change

FIGURE 4.9: ResNet-50 based model training and validation loss change.

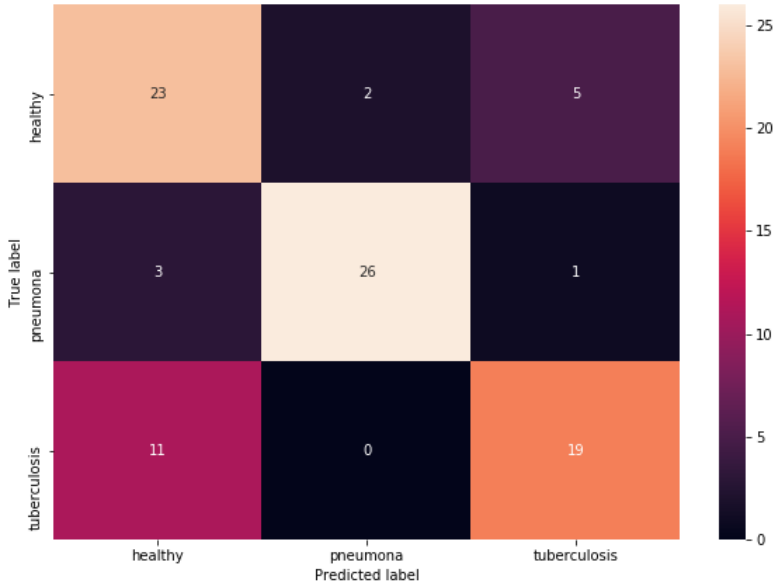


(A) Training accuracy change

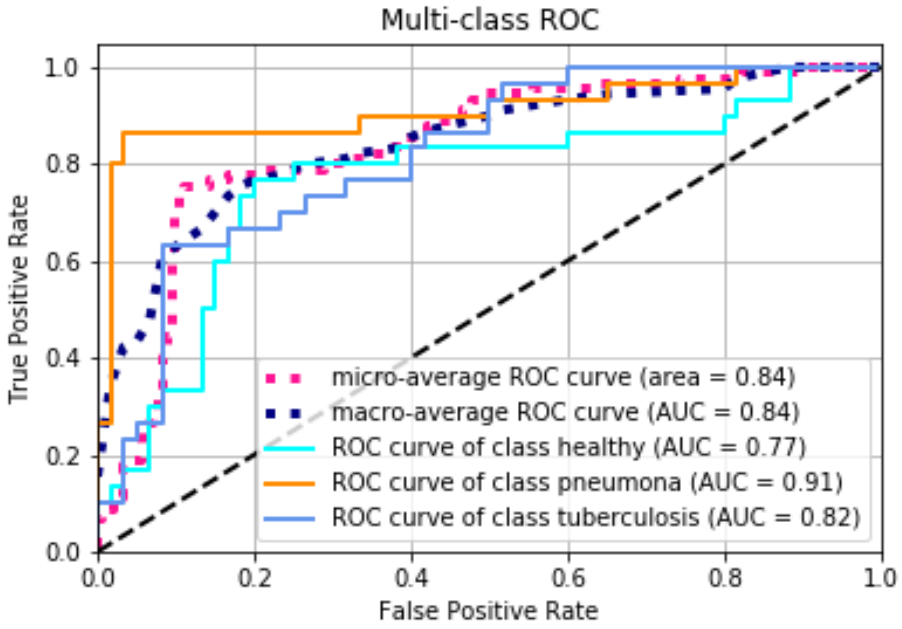


(B) Validation accuracy change

FIGURE 4.10: ResNet-50 based model training and validation accuracy change.



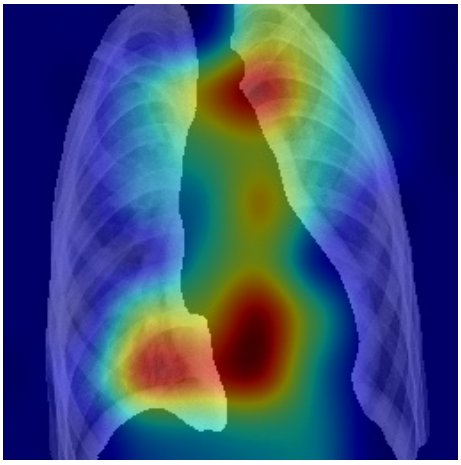
(A) Confusion matrix



(B) Per class ROC curves

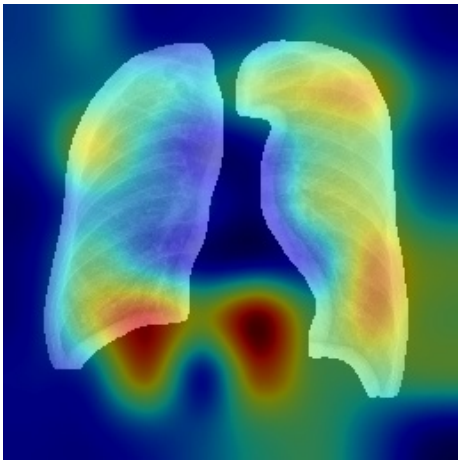
FIGURE 4.11: Image A shows the confusion matrix of the obtained results with the model which reached the highest accuracy during the training. Image B show its per class ROC curves.

(A) Correctly classified image as 'tuberculosis'



(B) Class activation map

(C) Correctly classified image as 'pneumonia'



(D) Class activation map

FIGURE 4.12: Two pairs of correctly labelled images containing marks of tuberculosis and pneumonia with their class activation maps.

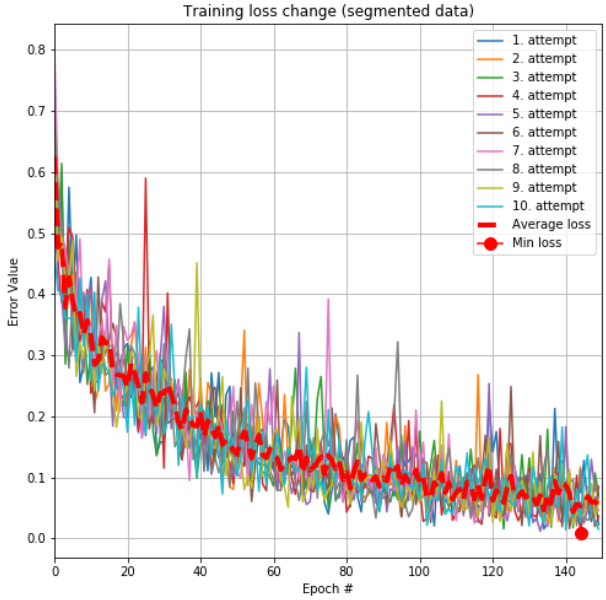
4.4.3 Inception results

The following results were generated for ten independent training runs in order to observe a similar training pattern. Each of the ten training and validation curves (see Figures 4.13 and 4.14) were plotted on the same charts based on the tape (training or validation). The same as before, we separate the results to maintain a high level of readability. The wider, dotted curve

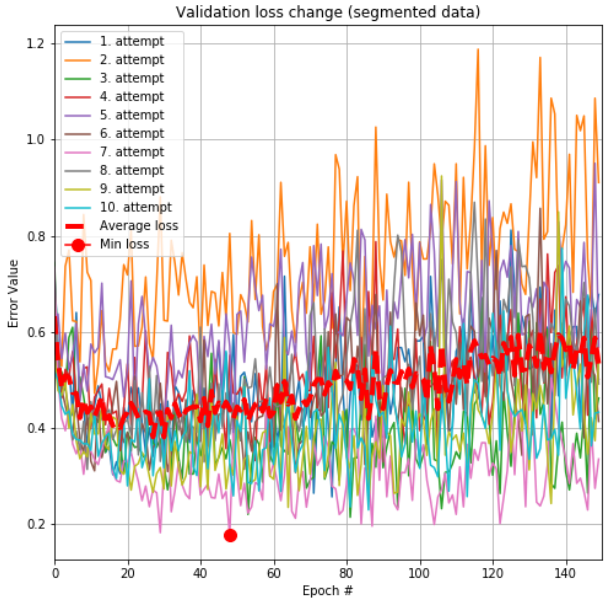
is an averaged result of all ten runs at the particular epoch. The red dot on Figures 4.13 4.14 stand for the lowest loss value on training and validation data sets the maximum accuracy obtained, respectively.

Figure 4.13 shows that the model starts overfitting on the training dataset around the 40th epoch, although we do not witness any noticeable drop in terms of accuracy. The validation error curve slowly increases its value, whereas the accuracy level remains similar (see Figure 4.14). Finally, we took all InceptionV3 based models at the 40th epoch and evaluated them on the test set and scored an average accuracy of 82.22%, which is a small improvement comparing to results obtained using non-segmented X-Ray images in3.6.3. However, this experiment provided us with the best average accuracy across all the models we trained using different techniques. In order to visualize the final results, similarly to the previous chapter, we selected a model which achieved the best accuracy score after the 40th epoch. The confusion matrix in Figure 4.15 A) shows that the new model improved the number of true positives (TP) in all classes comparing to previous algorithms. Image B) shows that the AUC score for healthy, tuberculosis and pneumonia were equal to 0.90, 0.93, and 0.99, respectively. This is a slight drop for the healthy class comparing to results in 4.4.1 and 3.6.3 albeit we greatly improved in comparison to our first results in 3.6.1.

Additionally, we present sample classification results in Figure 4.16. Similarly to previous subsections (4.4.1 and 4.4.2), both images A) and C) were correctly classified and the overlapping class activation maps (images B) and D)) show that the determinative regions were related to langs.

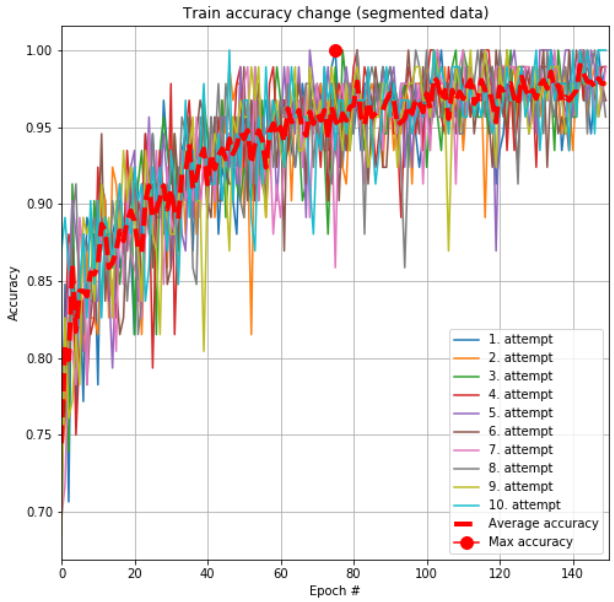


(A) Training error change

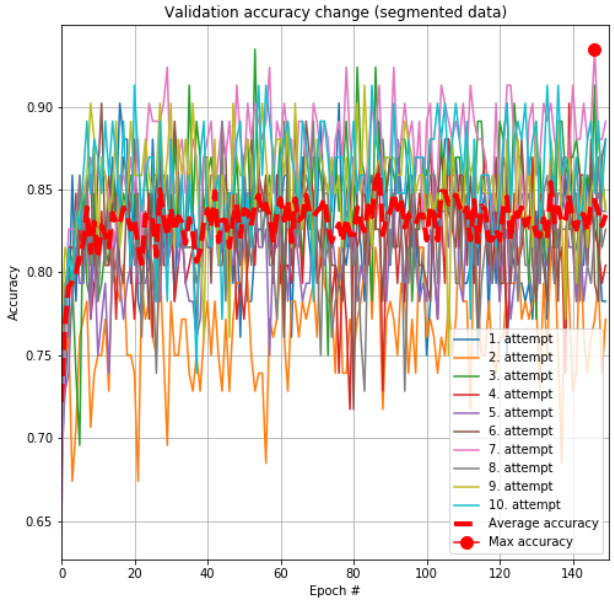


(B) Validation error change

FIGURE 4.13: InceptionV3 based model training and validation loss change.

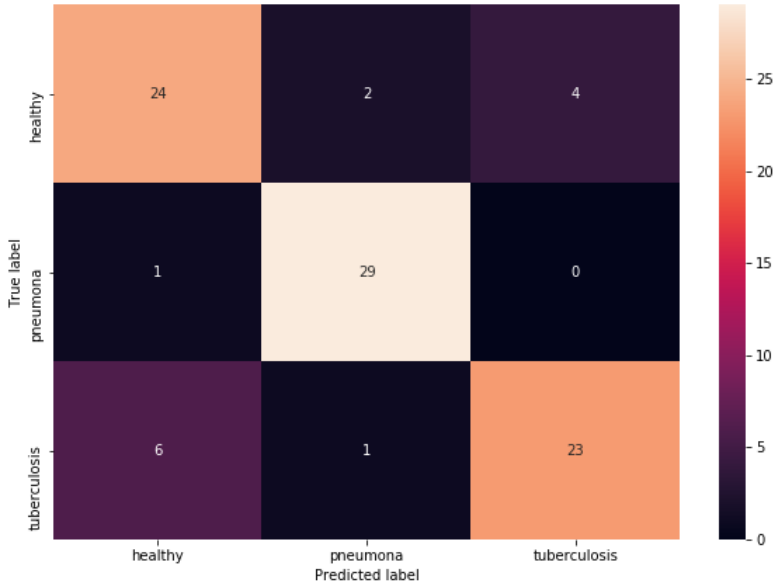


(A) Training accuracy change

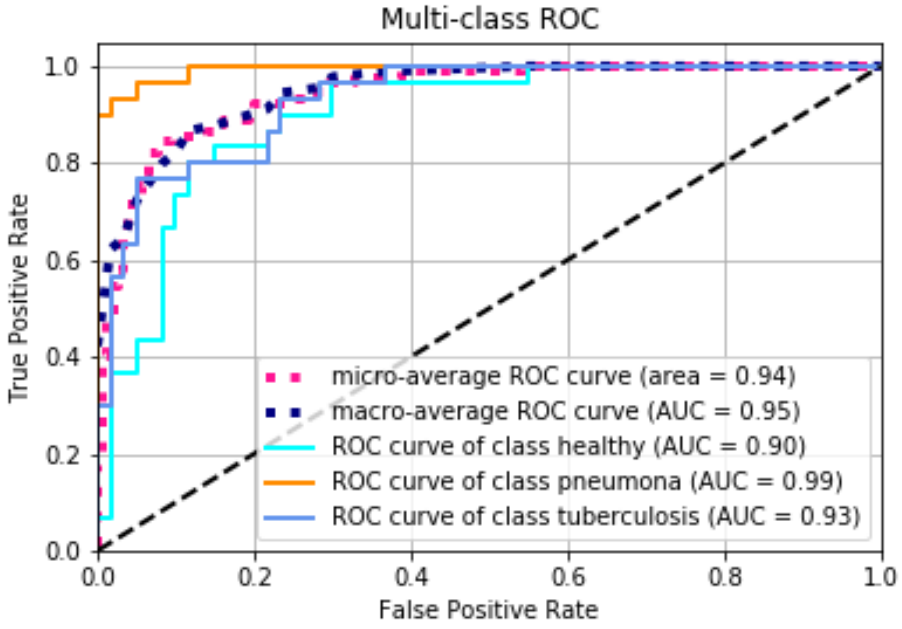


(B) Validation accuracy change

FIGURE 4.14: InceptionV3 based model training and validation accuracy change.



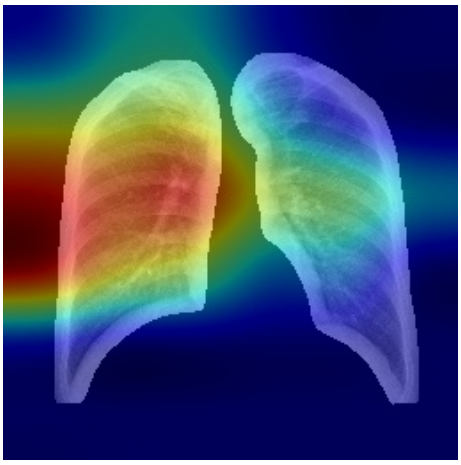
(A) Confusion matrix



(B) Per class ROC curves

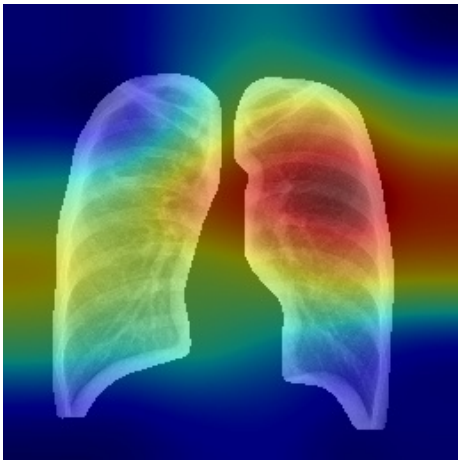
FIGURE 4.15: Image A shows the confusion matrix of the obtained results with the model which reached the highest accuracy during the training. Image B show its per class ROC curves.

(A) Correctly classified image as 'tuberculosis'



(B) Class activation map

(C) Correctly classified image as 'pneumonia'



(D) Class activation map

FIGURE 4.16: Two pairs of correctly labelled images containing marks of tuberculosis and pneumonia with their class activation maps.

4.5 Comparison of results

After comparing the results obtained in subsections 4.4.1, 4.4.2 and 4.4.3 we can observe that transfer-learning models perform well in lung diseases classification using segmented images tasks even when the data resources are limited. Not only is their accuracy improved, yet also the class activation maps support our conclusion. Table 4.1 shows comparison of results for all trained algorithms, both using segmented and non-segmented Chest X-Ray mages.

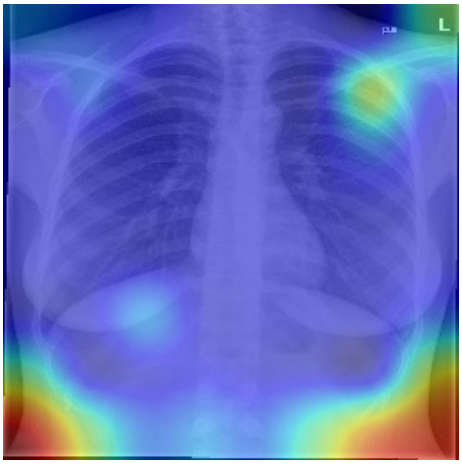
	Non segmented CXRs			Segmented CXRs		
	VGG16	ResNet-50	InceptionV3	VGG16	ResNet-50	Inception
Accuracy	0.64	0.72	0.81	0.70	0.75	**0.82**
AUC (healthy)	0.68	0.84	0.89	0.75	0.77	**0.90**
AUC (pneumonia)	0.80	0.84	0.92	0.81	0.91	**0.99**
AUC (tuberculosis)	0.82	0.76	0.87	0.90	0.82	**0.93**
F1 score (healthy)	0.55	0.66	**0.76**	0.55	0.62	**0.76**
F1 score (pneumonia)	0.64	0.82	0.90	0.84	0.89	**0.93**
F1 score (tuberculosis)	0.72	0.66	0.75	**0.79**	0.73	0.78
precision (healthy)	0.49	0.77	**0.80**	0.67	0.66	0.75
precision (pneumonia)	0.87	0.76	0.89	0.92	**0.93**	0.90
precision (tuberculosis)	0.71	0.67	0.74	0.68	0.71	**0.81**
sensitivity (healthy)	0.62	0.60	0.73	0.48	0.62	**0.77**
sensitivity (pneumonia)	0.53	0.88	0.92	0.78	0.85	**0.95**
sensitivity (tuberculosis)	0.77	0.68	0.77	**0.95**	0.78	0.75

TABLE 4.1: Comparison of all results received on segmented and non-segmented data.

The algorithm that scored the best in the majority results was InceptionV3 trained on the segmented images. What is more, it produced incredibly high scores for the "diseased" classes showing that a random instance containing marks of tuberculosis or pneumonia has over 90% probability to be classified to the correct class. Although the scores of the healthy class are worse than the diseased ones, its real cost is indeed lower as it is always worse to classify a sick patient as healthy.

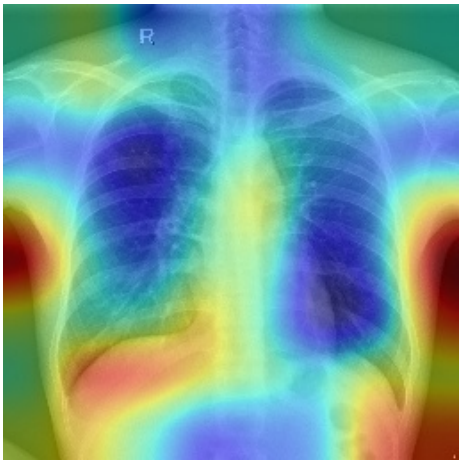
The InceptionV3 based model scored the best, reaching better accuracy than

VGG16 algorithms by over 12 percentage. Although the interpretability of our methods is not guaranteed, we can clearly state that using transfer-learning based algorithms on small datasets allows achieving competitive classification scores on the unseen data.



(A) Activation class map over a non segmented image

(B) Activation class map over a segmented image



(C) Activation class map over a non segmented image

(D) Activation class map over a segmented image

FIGURE 4.17: Two pairs of correctly classified images with their class activation maps. The left columns is non segmented images and the right is the segmented ones.

Furthermore, we compared the class activation maps and in order to investigate the reasoning behind decision making. The remaining features, here lungs, force the network to explore it and thus make decisions based on observed changes. That behavior was expected and additionally improved the interpretability of our models as the marked regions might bring attention in case of sick patients.

4.6 Comparison with other works

In this section, we would like to compare our models to the results achieved over different datasets. In order to do so we trained our algorithms on Shenzhen and Montgomery datasets [11] ten times, generated the results for all the models and averaged their scores: accuracy, precision, sensitivity, specificity, F1 score, and AUC 2.7

Table 4.2 shows the comparison of different deep learning models trained on the Shenzhen dataset [11]. Although our approach does not guarantee the best performance, it's close to the highest one yet less complex. Researchers in [57] use various pre-trained models in the pulmonary disease detection task, and the ensemble of them presents the highest accuracy and sensitivity. To compare, our InceptionV3 based model achieves accuracy smaller by only one percent and equal AUC, which means that our method gives an equal probability of assigning a positive case of tuberculosis to its corresponding class over a negative sample. Although we could not outperform the best solution, our approach is less complicated.

Furthermore, we compared the performance of our approach trained on the Montgomery dataset [11] (see Table 4.3). Our InceptionV3 based model tied with [12] in terms of accuracy yet showed a higher value of AUC. ResNet-50 and VGG16 based models performed worse, however not drastically as

	Accuracy	Precision	Sensitivity	Specificity	F1 score	AUC
[58]	0.82	-	-	-	-	-
[12]	0.84	-	-	-	-	0.90
VGG16 [57]	0.84	-	**0.96**	0.72	-	0.88
ReNet-50 [57]	0.86	-	0.84	0.88	-	0.90
ResNet-152 [57]	0.88	-	0.80	0.92	-	0.91
Ensemble [57]	**0.90**	-	0.88	0.92	-	**0.94**
VGG16	0.84	0.88	0.80	0.89	0.83	0.86
ResNet-50	0.85	**0.97**	0.73	**0.98**	0.83	0.92
InceptionV3	0.89	0.96	0.80	0.97	**0.88**	**0.94**

TABLE 4.2: Comparison of different deep learning based solutions trained on the Shenzhen datases. Although our result is not the best, it performs better than any single model (excluding Ensemble). Horizontal line means that those results were not provided in literature.

	Accuracy	Precision	Sensitivity	Specificity	F1 score	AUC
[12]	**0.790**	-	-	-	-	0.811
[59]	0.674	-	-	-	-	0.884
[13]	0.783	-	-	-	-	0.869
VGG16	0.727	**0.842**	0.581	**0.872**	0.669	**0.931**
ResNet-50	0.764	0.814	0.691	0.836	0.744	0.891
InceptionV3	**0.790**	0.822	**0.745**	0.836	**0.779**	0.884

TABLE 4.3: Comparison of different deep learning based solutions trained on the Montgomery dataset [11]. Our average performance is almost identical to [12].

they reached accuracies of 76% and 73% respectively, which is roughly 3 and 6 percent less than the highest score achieved.

Chapter 5

Conclusions and Future Work

5.1 Summary

In this thesis, we focused on exploring lung disease classification problem using deep neural networks preceded by segmentation and not under the supervision of the small size dataset (less than 10^3 examples). Moreover, we examined class activation maps to explore the reasoning of our models and investigate which regions are determinative. In Chapter 3 we first introduced different deep learning architectures which competed in ImageNet challenge [50]. Then, we use those networks as feature extractors to train our shallow algorithms. The results are summarized in section 3.6.4, here we only use accuracy to evaluate the performance since the test set is class equally distributed. Chapter 4 introduces the U-Net deep neural network and proposes a disease classification pipeline where Chest X-Ray images are first segmented before processing them through models described in Chapter 3. We train the same algorithms using a preprocessed dataset and compare the results with those obtained after learning features from non-segmented images. Here we also show how our solutions outperform deeper models trained on the same data. After comparing class activation maps in section 4.5, we conclude that segmentation not only improved the accuracy score yet also the reasoning behind the classification. Preprocessed Chest X-Ray images with remaining lungs force networks to explore only those areas which

lead to improvements in the interpretability of our models as the marked areas might bring attention in case of sick patients.

5.2 Contributions

The main contributions of this thesis are:

1. Implementation of transfer learning-based models such as VGG16, ResNet-50, and InceptionV3 in lung disease classification tasks.
2. Implementation of the U-Net model for lung segmentation.
3. Evaluation of algorithms trained on segmented and non-segmented Chest X-Ray images. Here, we compared both models' accuracies and their class activation maps.
4. Evaluation of the Shenzhen Hospital X-ray and Montgomery datasets for tuberculosis prediction and comparison to previous work done with deep neural networks.

5.3 Future Work

In Chapter 3, we examined only a small portion of available, pre-trained networks in the classification task. Even though the results are promising, other networks could be explored and examined in terms of class activation maps. Furthermore, we only used one classifier (3 layers deep neural network) due to computational and time limitations. Another direction would be the application of the introduced solutions to much bigger datasets such as ChestX-ray14 [60]. In Chapter 4, we propose a pipeline where classification is proceeded by segmentation. This part opens another area of exploration. As our models label Chest X-Ray images based on features extracted from

segmented data, yet it is beyond our expertise to decide whether the determinative regions truly contain marks of disease.

This work presents only a small part of research on lung disease classification. Considering the promising results we achieved and the relatively recent interest of deep neural network techniques in the medical field, there is plenty of room for improvements in other biomedical applications. Furthermore, we hope that one day computers will accelerate and help with a radiological examination and save the lives of millions.

Bibliography

[1] A. Karargyris, J. Siegelman, D. Tzortzis, S. Jaeger, S. Candemir, Z. Xue, K. Santosh, S. Vajda, S. Antani, L. Folio, and G. R Thoma, "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, pp. 563–576, June 2015.

[2] I. Livieris, A. Kanavos, V. Tampakas, and P. Pintelas, "An ensemble ssl algorithm for efficient chest x-ray image classification," *Journal of Imaging*, vol. 4, p. 95, July 2018.

[3] S. Lopez-Garnier, P. Sheen, and M. Zimic, "Automatic diagnostics of tuberculosis using convolutional neural networks analysis of mods digital images," *PLOS ONE*, vol. 14, pp. 1–16, 2019.

[4] S. Stirenko, Y. Kochura, O. Alienin, and O. Rokovyi, "Chest x-ray analysis of tuberculosis by deep learning with segmentation and augmentation," *2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO)*, pp. 422–428, 2009.

[5] Google, "Machine learning crash course with tensorflow apis." https://developers.google.com/machine-learning/crash-course/.

[6] Y. LeCun and Y. Bengio, *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 1998.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions,"

2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9, June 2015.

[8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, June 2016.

[9] O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, vol. 9351 of *LNCS*, (Cham), pp. 234–241, Springer International Publishing, 2015.

[10] K. Suzuki, "Pixel-based machine learning in medical imaging," *International Journal of Biomedical Imaging*, vol. 2012, pp. 1–1, February 2012.

[11] 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, December 2014.

[12] 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, pp. 1–9, December 2019.

[13] S. Jaeger, A. Karargyris, S. Candemir, L. Folio, J. Siegelman, F. Callaghan, Z. Xue, K. Palaniappan, R. K. Singh, S. Antani, G. Thoma, Y. Wang, P. Lu, and C. J. McDonald, "Automatic tuberculosis screening using chest radiographs," *IEEE Transactions on Medical Imaging*, vol. 33, pp. 233–245, February 2014.

[14] T. Ishigaki, S. Sakuma, and M. Ikeda, "One-shot dual-energy subtraction chest imaging with computed radiography: Clinical evaluation of film images," *Radiology*, vol. 168, pp. 67–72, August 1988.

[15] A. Karargyris, J. Siegelman, D. Tzortzis, S. Jaeger, S. Candemir, Z. Xue, K. Santosh, S. Vajda, S. Antani, L. Folio, and G. R Thoma, "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, pp. 563–576, June 2015.

[16] C. S. Pereira, H. Fernandes, A. M. Mendonça, and A. Campilho, "Detection of lung nodule candidates in chest radiographs," in *Pattern Recognition and Image Analysis* (J. Martí, J. M. Benedí, A. M. Mendonça, and J. Serrat, eds.), (Berlin, Heidelberg), pp. 170–177, Springer Berlin Heidelberg, 2007.

[17] G. Coppini, S. Diciotti, M. Falchini, N. Villari, and G. Valli, "Neural networks for computer-aided diagnosis: Detection of lung nodules in chest radiograms," *IEEE Transactions on Information Technology in Biomedicine*, vol. 7, pp. 344–57, January 2004.

[18] R. Hardie, S. Rogers, T. Wilson, and A. Rogers, "Performance analysis of a new computer aided detection system for identifying lung nodules on chest radiographs," *Medical Image Analysis*, vol. 12, pp. 240–58, July 2008.

[19] S. Chen, K. Suzuki, and H. Macmahon, "Development and evaluation of a computer-aided diagnostic scheme for lung nodule detection in chest radiographs by means of two-stage nodule enhancement with support vector classification," *Medical Physics*, vol. 38, pp. 1844–58, April 2011.

[20] K. Suzuki, I. Horiba, N. Sugie, and M. Nanki, "Extraction of left ventricular contours from left ventriculograms by means of a neural edge detector," *IEEE Trans. Med. Imaging*, vol. 23, no. 3, pp. 330–339, 2004.

[21] K. Suzuki, I. Horiba, and N. Sugie, "Neural edge enhancer for supervised edge enhancement from noisy images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1582–1596, December 2003.

[22] K. Suzuki, "Neural filter with selection of input features and its application to image quality improvement of medical image sequences," *IEICE Transactions on Information and Systems*, vol. E85-D, pp. 1710–1718, 2002.

[23] K. Suzuki, I. Horiba, and N. Sugie, "Efficient approximation of neural filters for removing quantum noise from images," *IEEE Transactions on Signal Processing*, vol. 50, pp. 1787–1799, July 2002.

[24] B. Sahiner, Heang-Ping Chan, N. Petrick, Datong Wei, M. A. Helvie, D. D. Adler, and M. M. Goodsitt, "Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images," *IEEE Transactions on Medical Imaging*, vol. 15, pp. 598–610, October 1996.

[25] S.-C. B. Lo, H. Li, Y. J. Wang, L. Kinnard, and M. T. Freedman, "A multiple circular path convolution neural network system for detection of mammographic masses," *IEEE Transactions on Medical Imaging*, vol. 21, pp. 150–158, 2002.

[26] S.-C. Lo, S.-L. Lou, J.-S. Lin, M. T. Freedman, M. V. Chien, and S. Mun, "Artificial convolution neural network techniques and applications for lung nodule detection," *IEEE Transactions on Medical Imaging*, vol. 14, pp. 711 – 718, December 1995.

[27] S.-C. B. Lo, H.-P. Chan, J.-S. Lin, H. Li, M. T. Freedman, and S. K. Mun, "Artificial convolution neural network for medical image pattern recognition," *Neural Networks*, vol. 8, pp. 1201–1214, December 1995.

[28] C. Nebauer, "Evaluation of convolutional neural networks for visual recognition," *IEEE Transactions on Neural Networks*, vol. 9, pp. 685–696, July 1998.

[29] S. Lawrence, C. Giles, A. C. Tsoi, and A. Back, "Face recognition: a convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, pp. 98–113, January 1997.

[30] F. Li, H. Arimura, K. Suzuki, J. Shiraishi, Q. H. Li, H. Abe, R. M. Engelmann, S. Sone, H. MacMahon, and K. Doi, "Computer-aided detection of peripheral lung cancers missed at ct: Roc analyses without and with localization.," *Radiology*, vol. 237 2, pp. 684–90, 2005.

[31] M. Loog, B. van Ginneken, and A. Schilham, "Filter learning: Application to suppression of bony structures from chest radiographs," *Medical Image Analysis*, vol. 10, pp. 826–40, January 2007.

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

[33] J.-S. Lin, S.-C. B. Lo, A. Hasegawa, M. T. Freedman, and S. K. Mun, "Reduction of false positives in lung nodule detection using a two-level neural classification," *IEEE Transactions on Medical Imaging*, vol. 15, pp. 206–17, 1996.

[34] W. Zhang, K. Doi, M. L. Giger, R. M. Nishikawa, and R. A. Schmidt, "An improved shift-invariant artificial neural network for computerized detection of clustered microcalcifications in digital mammograms," *Medical Physics*, vol. 23, no. 4, pp. 595–601, 1996.

[35] S. Oda, K. Awai, K. Suzuki, Y. Yanaga, Y. Funama, H. Macmahon, and Y. Yamashita, "Performance of radiologists in detection of small pulmonary nodules on chest radiographs: Effect of rib suppression with a massive-training artificial neural network," *AJR. American Journal of Roentgenology*, vol. 193, pp. W397–402, November 2009.

[36] U. D. of Health and H. Services, "Pneumonia can be prevented - vaccines can help." https://www.cdc.gov/features/pneumonia/index.html.

[37] G. Ramalho and L. Bezerra, "Lung disease detection using feature extraction and extreme learning machine," *Revista Brasileira de Engenharia Biomedica*, vol. 30, pp. 207 – 214, September 2014.

[38] F. J. Aherne, N. A. Thacker, and P. I. Rockett, "The bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetika*, vol. 34, pp. 363–368, 1998.

[39] G. Beylkin, "Discrete radon transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, pp. 162–172, February 1987.

[40] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 978–994, May 2011.

[41] T. Evgeniou and M. Pontil, "Support vector machines: Theory and applications," *Machine Learning and Its Applications: Advanced Lectures*, vol. 2049, pp. 249–257, January 2001.

[42] G. Kostopoulos, I. E. Livieris, S. B. Kotsiantis, and V. Tampakas, "Cst-voting: A semi-supervised ensemble method for classification problems," *Journal of Intelligent and Fuzzy Systems*, vol. 35, pp. 99–109, 2018.

[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computing Research Repository*, vol. abs/1409.1556, 2014.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems 25*, pp. 1097–1105, 2012.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[46] Y. Gordienko, P. Gang, J. Hui, W. Zeng, Y. Kochura, O. Alienin, O. Rokovyi, and S. Stirenko, "Deep learning with lung segmentation and bone shadow exclusion techniques for chest x-ray analysis of lung cancer," *Computing Research Repository*, vol. abs/1712.07632, 2017.

[47] 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, pp. 577–590, February 2014.

[48] K. Z. D. Kermany and M. Goldbaum, "Large dataset of labeled optical coherence tomography (oct) and chest x-ray images," *Cell*, pp. 1122–1131, 2018.

[49] M. Hossin and S. M.N, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining and Knowledge Management Process*, vol. 5, pp. 01–11, March 2015.

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

[51] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *ArXiv*, vol. abs/1712.04621, 2017.

[52] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, "Deep image: Scaling up image recognition," *ArXiv*, vol. abs/1501.02876, January 2015.

[53] D. Hana, Q. Liu, and W. Fan, "A new image classification method using cnn transfer learning and web data augmentation," *Expert Systems with Applications*, vol. 95, p. 57–71, November 2017.

[54] Q. Xu, Y.-Z. Liang, and Y.-P. Du, "Monte carlo cross-validation for selecting a model and estimating the prediction error in multivariate calibration," *Journal of Chemometrics*, vol. 18, pp. 112 – 120, February 2004.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computing Research Repository*, vol. abs/1412.6980, 2015.

[56] Q. Kang, Q. Lao, and T. Fevens, "Nuclei segmentation in histopathological images using two-stage learning," *Medical Image Computing and Computer Assisted Intervention (MICCAI 2019)*, October 2019. Accepted June 4, 2019.

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

[58] R. H. Anuj Rohilla and A. Mittal, "Tb detection in chest radiograph using deep learning architecture," *International Journal of Advance Research in Science and Engineering*, vol. 6, pp. 1073–1084, August 2017.

[59] J. J. M. H.-J. K. Sangheum Hwang, Hyo-Eun Kim, "A novel approach for tuberculosis screening based on deep convolutional neural networks," *Medical Imaging 2016: Computer-Aided Diagnosis*, vol. 9785, pp. 1–23, 2016.

[60] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3462–3471, 2017.