Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Engineering

Diploma Thesis

Classification of common thoracic disorders from X-ray images

Enkhtaivan GANBAT

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Bc. Enkhtaivan Ganbat

Systems Engineering and Informatics
Informatics

Thesis title

Classification of common thoracic disorders from X-ray images

---

Objectives of thesis

The objective of this thesis is to identify and classify 14 common thoracic disorders from chest X-ray images.
The dataset to be used is a publicly available dataset from the U.S. National Institute of Health (NIH).
The dataset contains about 112 thousand X-ray images of nearly 33 thousand patients.

Methodology

The multi-class classification task will be done using modified or custom deep convolutional neural networks.

There are existing implementations of similar tasks based on the same dataset, but those are implemented using the PyTorch framework.

In this thesis I will attempt to implement using Keras and Tensorflow frameworks.

The proposed extent of the thesis

60 pages

Keywords

Deep learning, convolution networks

Recommended information sources

GÉRON, A. *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems.* Beijing ; Boston ; Farnham ; Sebastopol ; Tokyo: O'Reilly, 2019. ISBN 978-1-492-03264-9.

Expected date of thesis defence

2020/21 SS – FEM

The Diploma Thesis Supervisor

doc. Ing. Arnošt Veselý, CSc.

Supervising department

Department of Information Engineering

Electronic approval: 9. 3. 2021

Ing. Martin Pelikán, Ph.D.

Head of department

Electronic approval: 9. 3. 2021

Ing. Martin Pelikán, Ph.D.

Dean

Prague on 11. 03. 2021

Declaration

I declare that I have worked on my diploma thesis titled "Classification of common thoracic disorders from X-ray images" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break any copyrights.

In Prague on 29th of March _____

Acknowledgement

I appreciate the efforts devoted to the collection of the ChestX-ray14 dataset.

Classification of common thoracic disorders from X-ray images

Abstract

I have trained and tested two models based on the ResNet-50 and MobileNet models. The models have been trained on the ChestX-ray14 dataset, which is currently one of the largest publicly available dataset of X-ray images, containing over 112 thousand images collected from the medical reports of nearly 31 thousand patients.

Each X-ray image is labelled either 'No Findings' (normal) or with at least one of the 14 thoracic disorders. AUC (Area Under Curve) scores were used to measure model accuracy.

Furthermore, three Voting Classifiers: Max Vote, Hard Vote, Simple Average were used along with AUC scores from the two models in an attempt achieve higher classification accuracy.

Keywords: model, training, image, dataset, label, classification, accuracy, X-ray, AUC, Voting Classifier.

Klasifikace běžných hrudních poruch z rentgenových snímků

Abstrakt

Vyškolil jsem a otestoval dva modely založené na modelech ResNet-50 a MobileNet. Modely byly proškoleny na datovém souboru ChestX-ray14, který je v současné době jedním z největších veřejně dostupných datových souborů rentgenových snímků, který obsahuje více než 112 tisíc snímků shromážděných z lékařských zpráv téměř 31 tisíc pacientů.

Každý rentgenový snímek je označen buď jako „Žádný nález“ (normální nález), nebo alespoň s jednou ze 14 hrudních poruch. K měření přesnosti modelu byla použita skóre AUC (Area Under Curve).

Kromě toho byly použity tři klasifikátory hlasování: Max Vote, Hard Vote, Simple Average a skóre AUC ze dvou modelů ve snaze dosáhnout vyšší přesnosti klasifikace.

Klíčová slova: model, trénink, obrázek, datová sada, štítek, klasifikace, přesnost, rentgen, AUC, klasifikátor hlasování.

Table of content

List of pictures

List of abbreviations

Cases

1 Introduction

Shortage of doctors, nurses is a very serious and increasing problem due to variety of factors such as: many doctors are reaching retirement age, not many doctors are being trained and many doctors leave their home country for higher paying jobs in wealthier countries, creating huge disparities among regions and countries.

Low fertility rate caused by variety of socioeconomic issues is the root of aging populations in developed countries. Increasing percentage of old people will naturally increase the demand for hospital beds, doctors, nurses and other hospital staff.

According to a report (European Commission, 2012) by the European Commission, number of elderly persons aged 65 and over in Europe is estimated to almost double from the 2010 figure of 87 million to 152.7 million by 2060.

Environmental pollution is a major factor exacerbating this problem further. Air pollution alone causes many life-threatening diseases such as ischemic heart disease, stroke, lung cancer and acute lower respiratory infection (World Health Organization, 2014) which often manifest as pneumonia. All these factors as well as the current COVID-19 pandemic is increasing the pressure on health care systems all over the world.

Radiologists are one of the most affected medical professionals (telemedicineclinic, 2016) by the sudden increase of workload and they can benefit from a computer-aided diagnosis (CAD) system that can identify thoracic disorders such as pneumonia. This is not to say that such a system will completely replace radiologists, but it can help reduce their workload. Advancements of the machine learning field in the last decade and the availability of massive datasets made it possible to implement such a system.

2 Objective and Methodology

2.1 Objective

The objective of this thesis is to train and test multiple state-of-the-art convolutional neural networks (CNNs) on the chosen dataset to compare their accuracies.
Moreover, use various Ensemble methods in conjunction with the models to compare the test results with the results of individual models.

2.2 Methodology

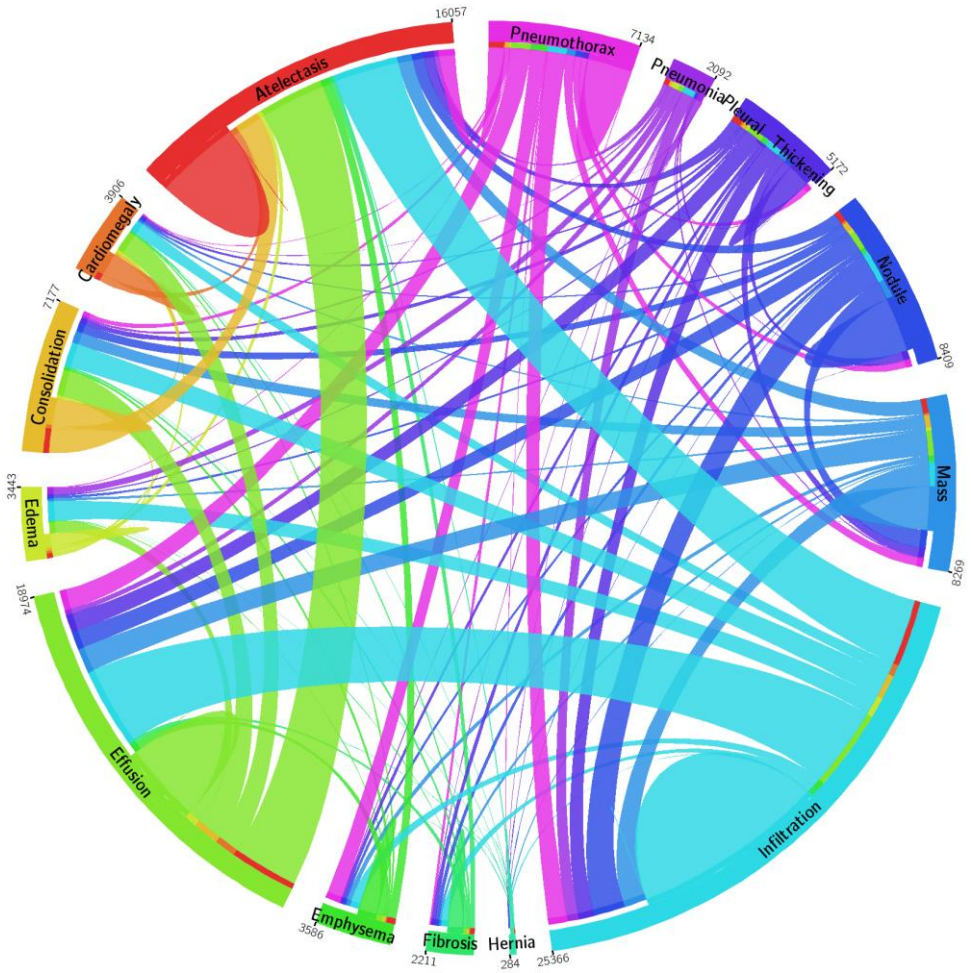
In this thesis, I have followed below steps:

1. Find data.
2. Discover and visualize the data to gain insights.
3. Prepare the data for CNN models.
4. Select multiple CNN models and train them on the original images.
5. Test each model on the test set and compare their accuracies.
6. Train each model further on augmented images.
7. Test each model on augmented images and compare the results with previous results.
8. Use Ensemble methods with the models to achieve higher accuracy.

2.3 Data

The chosen data used is a publicly available dataset provided by the U.S National Health Institute's (NIH) Clinical Centre and contains over 112 thousand anonymized frontal-view chest X-ray images from nearly 31 thousand patients (Wang, et al., 2017).
The image labels were mined from associated radiological reports using natural language processing (NLP). Each image can have multiple labels, the fourteen common thoracic disorders/pathologies represented by the labels are: Atelectasis, Consolidation, Infiltration, Pneumothorax, Edema, Emphysema, Fibrosis, Effusion, Pneumonia, Pleural thickening, Cardiomegaly, Nodule, Mass and Hernia. The text-mined disease labels are expected to have over 90% accuracy. Figure 1 shows the proportion of X-ray images with multi-labels for each of the 14 pathologies.

Figure 1. The circular diagram shows the proportions of images with multi-labes in each of 14 pathology classes and the labels' co-occurence statistics.



Source: (Wang, et al., 2017)

2.4 Implementation

Model training was done using a virtual machine on Google Compute Engine with the following specifications:

- 8 x vCPUs
- 52GB RAM
- 1 x NVIDIA Tesla P4 GPU
- Debian 10
- Tensorflow 2.4 pre-installed.

2.5 Model Architectures

Convolutional neural networks (CNNs) are special types of neural networks that are proven to be best for image processing tasks such as classification. CNNs are different from regular neural networks because they contain at least one convolutional layer, in which neurons are connected to pixels only in their receptive fields (called kernel) rather than every single pixel.

While it is possible to build a DCNN model from scratch by combining different layers such as convolutional, max pooling and dense, it is often more practical to use existing high-accuracy models as a base model and built on top of them.

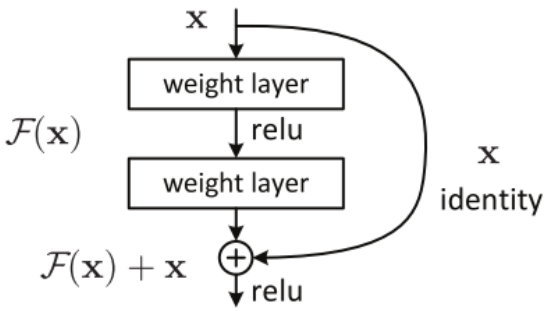
I have trained and compared two models based on ResNet50 and MobileNet model architectures.

2.5.1 ResNet

ResNet (stands for Residual Neural Network) won the 2015 ILSVRC competition by introducing a "residual block" which allowed one or more layers to be skipped. It's an extremely deep network with 152 layers and achieved top-5 error rate of 3.57%.

Figure 2 illustrates the residual block.

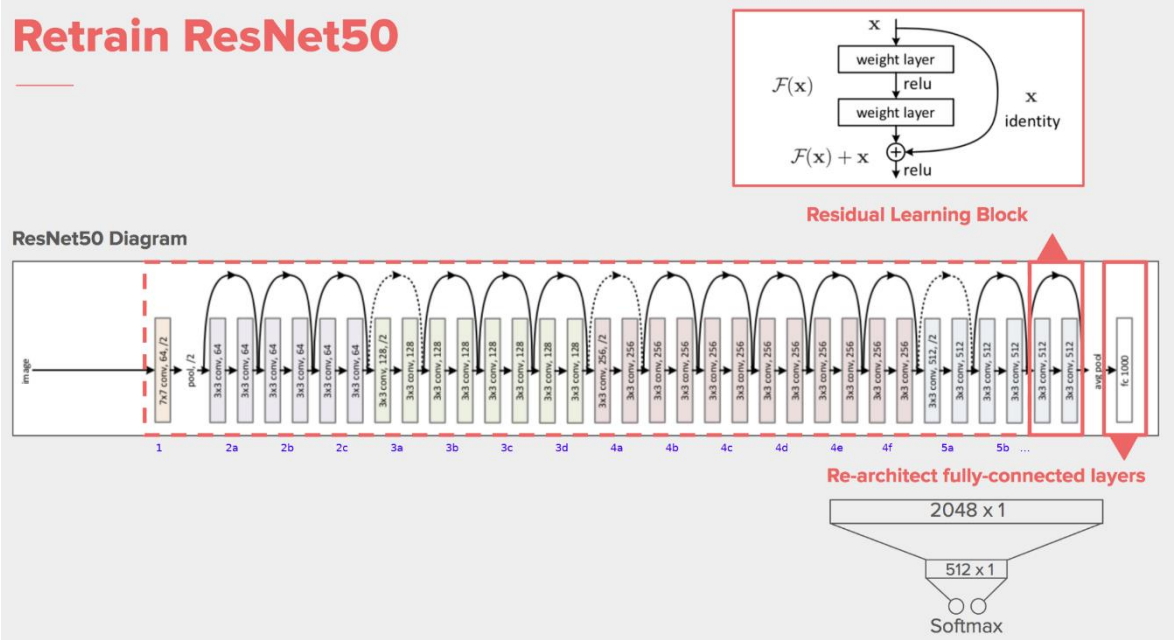
Figure 2. The residual block.



Source: (Garyfallos, et al., 2019)

The Top-5 error is the percentage of the time that the classifier did not include the correct class among its top 5 guesses. Figure 3 shows the architecture of ResNet-50 model.

Figure 3. Architecture of ResNet-50 model.

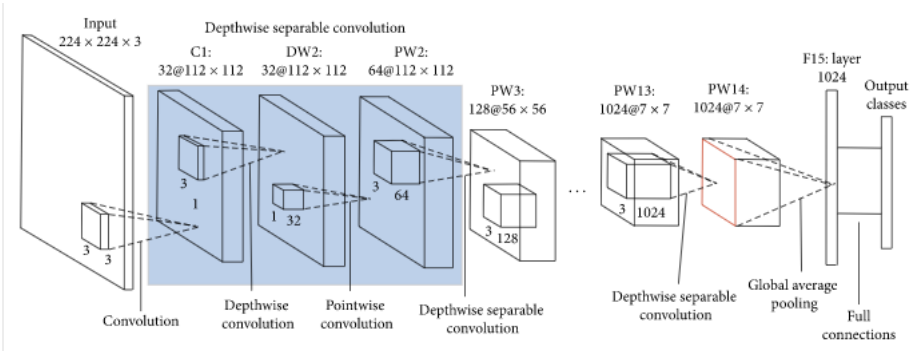


Source: www.stackoverflow.com

2.5.2 MobileNet

MobileNet is a light-weight deep convolutional neural network designed to run on embedded and mobile devices. MobileNet uses depth-wise separable convolutional layers. Figure 4 shows the architecture of MobilNet.

Figure 4. Architecture of MobileNet



Source: (Wang, et al., 2020)

2.5.3 Hyperparameters often used for tuning models

- Batch size

  Batch size is the size (number) of training instances used in a batch learning.

- Learning rate

  Learning rate is the size of the step in a Gradient Descent algorithm, if the learning rate is too small the algorithm will take too long (too many iterations) to reach the minimum value of the cost function, if the learning rate is too big, the algorithm might skip over the minimum and never converge.

- Momentum

  Momentum helps to know the direction of the next step with knowledge of the previous steps. It helps prevent oscillations.

- Optimizer

  Optimizers are algorithms or methods used for changing the attributes (weights, learning rate etc.) of a neural network to reduce loss faster.

- Image size

  Image size is important since many CNNs require the training images to be of uniform size or even have a specific resolution (e.g., 224 x 224).
  Moreover, CNNs generally train faster on smaller images.

3 Survey of Current Literature

Since the release of the ChestX-ray8 dataset (Wang, et al., 2017), the previous version of ChestX-ray14 in 2017, there has been many studies using this dataset for a multi-label classification of thorax diseases including the original paper.

The ChestX-ray8 dataset contained almost 109 thousand images, each labelled either with one or more of the 8 possible pathologies or "normal" in cases where no abnormality was detected.

3.1 ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases

In this paper (Wang, et al., 2017), the authors built a DCNN architecture called Unified Deep Convolutional Neural Network Framework by modifying pre-trained models such as AlexNet, GoogLeNet, VGG-16 and ResNet-50.

3.1.1 Model

The modification entailed removing the fully-connected and final classification layers and including a transition layer, global pooling layer, a prediction layer and finally a loss layer. They have used an 8-dimensional label vector for predictions. Indices in this vector represented a presence or a lack of pathologies with values 1 and 0 respectively, which transformed the multi-label classification problem to use a regression-like loss function. The role of the transition layer was to transform activations from previous layers into a uniform dimension of output since different pre-trained models has different settings, for example 1024 for GoogLeNet and 2048 for ResNet-50.

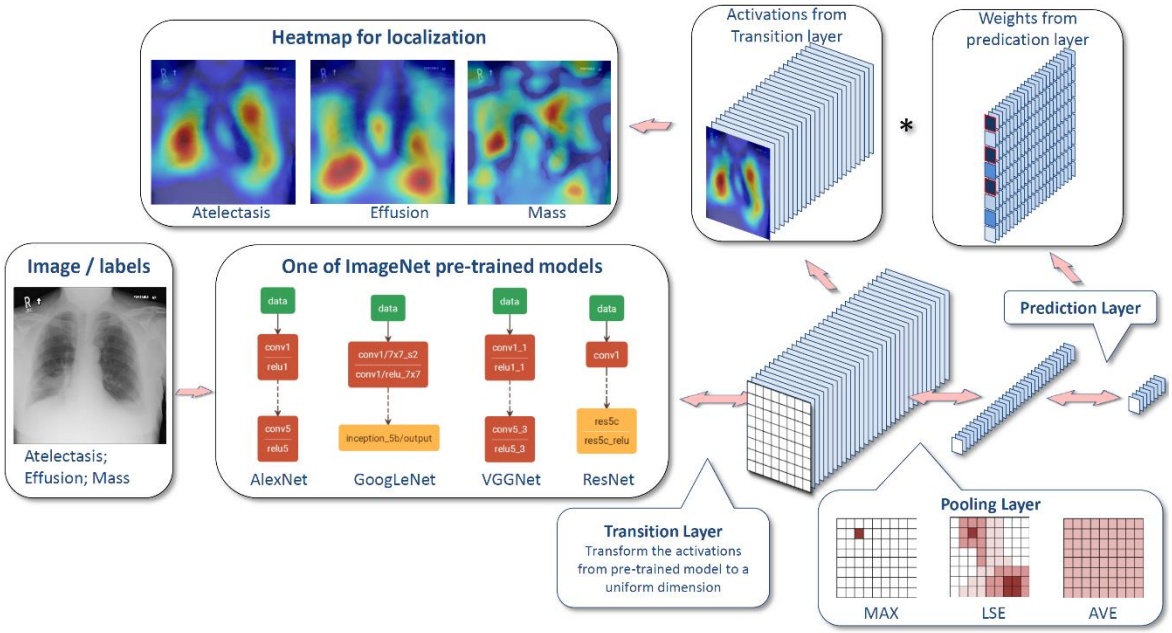
At the loss layer they first experimented with 3 standard loss functions: Hinge Loss, Euclidean Loss and Cross Entropy Loss. But the model had problems learning due to the rarity of positive (not "Normal") labels within the dataset.

Thus, they modified the Cross Entropy Loss (CEL) function to a Weighted Cross Entropy Loss (W-CEL) function by multiplying the 2 parts of CEL by $\beta_P = \frac{|P|+|N|}{|P|}$ and $\beta_N = \frac{|P|+|N|}{|N|}$ respectively, where |P| and |N| are total number of 1s and 0s in a batch of image labels.

They used the global pooling layer for not only classification, but also for generating heatmaps, and then used the heatmaps to generate bounding boxes.

Figure 5 shows the architecture of the Unified DCNN framework.

Figure 5. Unified DCNN framework



Source: (Wang, et al., 2017)

3.1.2 Experiments and Results

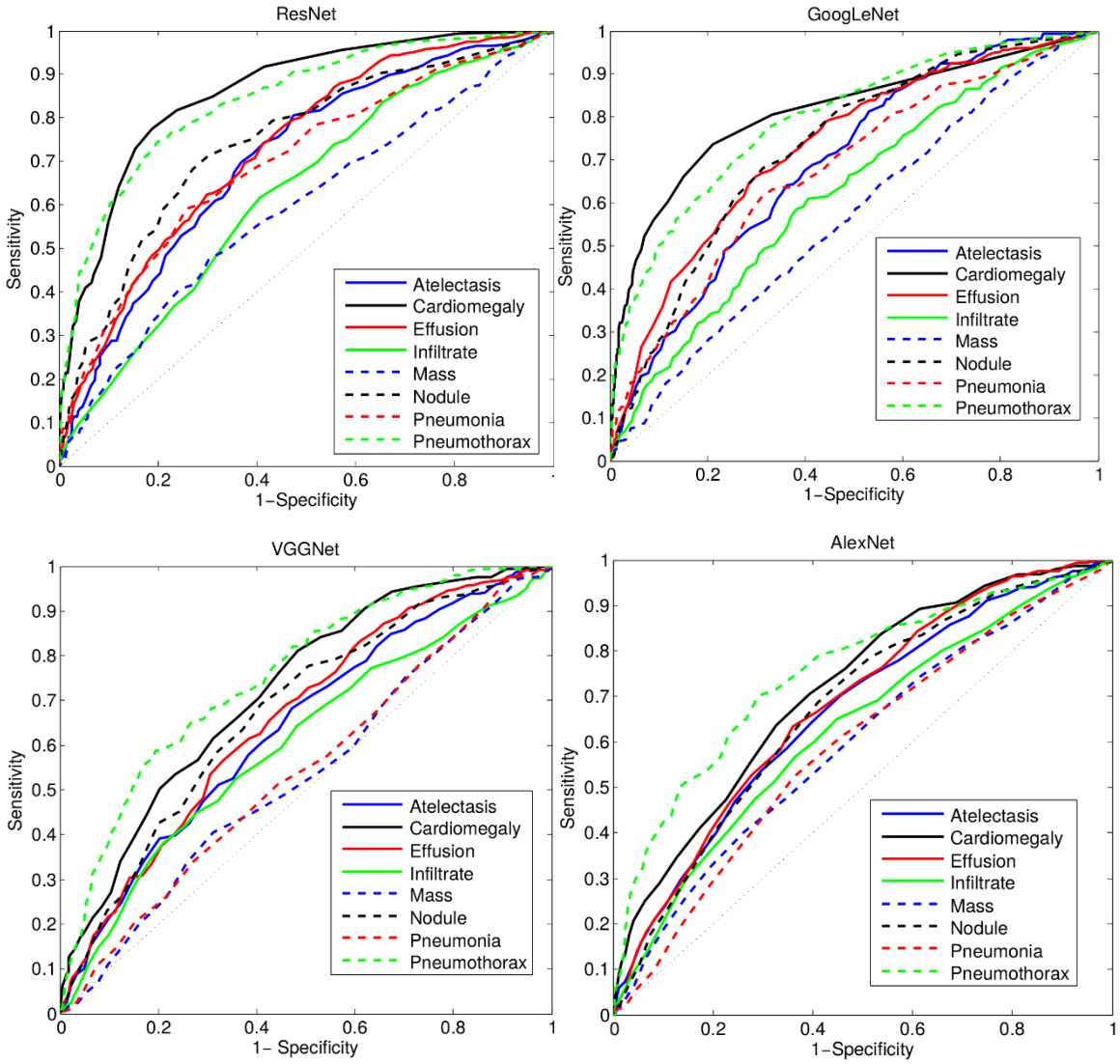
After experimenting with Unified DCNN frameworks based on four different DCNNs: AlexNet, GoogLeNet, VGG and ResNet-50, the one based on ResNet-50 achieved the highest Area-Under-Curve (AUC) value.

Figure 6 shows the ROC curve plots of Unified DCNNs based on ResNet, GoogLeNet, VGGNet and AlexNet.

The team further experimented using the ResNet-50 but with three different pooling schemes: Average Pooling, Max Pooling and LSE (stands for Log-Sum-Exp) pooling and found out that LSE outperformed average and max pooling schemes when the hyperparamter r = 10. Finally, the model performed better with W-CEL compared to CEL, especially on classes with few positive instances.

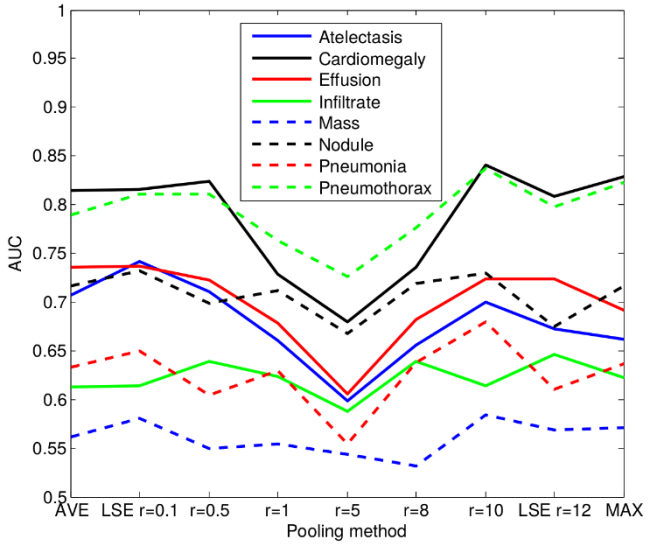
Figure 7 shows the comparison plot of three different pooling schemes: Average, Global and LSE.

Figure 6. Comparison of ROC curves.



Source: (Wang, et al., 2017)

Figure 7. Comparison of pooling schemes.



Source: (Wang, et al., 2017)

3.2 ChexNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning

ChexNet model was developed by a team of scholars from the department of Computer Science, Department of Medicine and the Department of Radiology at Stanford University.

3.2.1 Model

ChexNet is a 121-layer deep convolutional neural network that was trained on the ChestX-ray14 dataset. The model outputs the probability of pneumonia and a heatmap localizing the areas of the image most often associated with pneumonia. Figure 10 shows a heatmap on an X-ray image of patient with congestive heart failure and cardiomegaly (enlarged heart).
The team used a binary cross entropy loss function which is very similar to that of in the original paper (Wang, et al., 2017).

$$L(X, y) = -w_+ * y * log_p(Y = 1|X) - w_- * (1 - y) * log_p(Y = 0|X) \quad (1)$$

Where $p(Y = i|X)$ is the probability that the model assigns to the label i , $w_+ = \frac{|N|}{(|P|+|N|)}$, $w_- = \frac{|P|}{(|P|+|N|)}$ where |P| and |N| are the numbers of positive and negative cases of pneumonia in the training set respectively.
Unlike in the original paper (Wang, et al., 2017), the classification problem here is binary (pneumonia vs normal), thus the use of binary cross entropy loss function. Another significant difference is that the authors used diagnosis by four practicing radiologists to evaluate the model's accuracy.

3.2.2 Experiments and Results

The authors collected a test set of 420 frontal chest X-ray images. The labels were obtained independently from the four radiologists who had no information about the patient history or the test set. The team calculated the F1 scores (harmonic mean of precision and recall) for each radiologist and the model and used it as the ground truth. They also calculated the average F1 score of the radiologists. Moreover, the team also calculated 95% confidence intervals (CI) for both the radiologists and the model on 10 thousand bootstrap (bootstrapping is sampling method where a subset of the test set is randomly chosen, which means one instance can be sampled more than once) samples, sampled from the test set.

Figure 8 shows the comparison of the individual and average F1 scores of 4 radiologists and the ChexNet model.

Figure 8. Comparison of indivial and average F1 scores of radiologists against ChexNet

	F1 Score (95% CI)
Radiologist 1	0.383 (0.309, 0.453)
Radiologist 2	0.356 (0.282, 0.428)
Radiologist 3	0.365 (0.291, 0.435)
Radiologist 4	0.442 (0.390, 0.492)
Radiologist Avg.	0.387 (0.330, 0.442)
CheXNet	0.435 (0.387, 0.481)

Source: (Rajpurkar, et al., 2017)

To determine whether the model's accuracy was statistically significantly higher than radiologist diagnosis, the authors also calculated the difference between the average F1 score of the model and the radiologists on the same bootstrap samples, and they concluded that the difference was significant because the 95% confidence interval (0.051 (95% CI 0.005, 0.084)) did not include zero.

To compare the performance of ChexNet with models from other teams, the authors modified the ChexNet by changing the binary output to 14-dimensional vector to indicate the presence of the 14 pathology classes. This vector contained the predicted probabilities of each pathology class.

Finally, they modify the loss function to optimize the sum of unweighted binary cross entropy losses:

$$L(X, y) = \sum_{c=1}^{14} [-y_c log_p(Y_c = 1|X) - (1 - y_c) log_p(Y_c = 0|X)] \quad (2)$$

Where $p(Y_c = 1|X)$ is the predicted probability that the image contains the pathology 'c' and $p(Y_c = 0|X)$ is the predicted probability that the image does not contain the pathology 'c'. This modified model out-performed previous state-of-the art models on all 14 classes. Figure 9 shows the comparison of ChexNet's performance against previous state-of-the art models.

Figure 9. Comparison of ChexNet against previous state-of-the-art models

Pathology	Wang et al. (2017)	Yao et al. (2017)	CheXNet (ours)
Atelectasis	0.716	0.772	**0.8094**
Cardiomegaly	0.807	0.904	**0.9248**
Effusion	0.784	0.859	**0.8638**
Infiltration	0.609	0.695	**0.7345**
Mass	0.706	0.792	**0.8676**
Nodule	0.671	0.717	**0.7802**
Pneumonia	0.633	0.713	**0.7680**
Pneumothorax	0.806	0.841	**0.8887**
Consolidation	0.708	0.788	**0.7901**
Edema	0.835	0.882	**0.8878**
Emphysema	0.815	0.829	**0.9371**
Fibrosis	0.769	0.767	**0.8047**
Pleural Thickening	0.708	0.765	**0.8062**
Hernia	0.767	0.914	**0.9164**

Source: (Rajpurkar, et al., 2017)

Figure 10. Heatmap correctly localized on an X-ray image of a patient with congestive heart failure and cardiomegaly (enlarged heart)

Source: (Rajpurkar, et al., 2017)

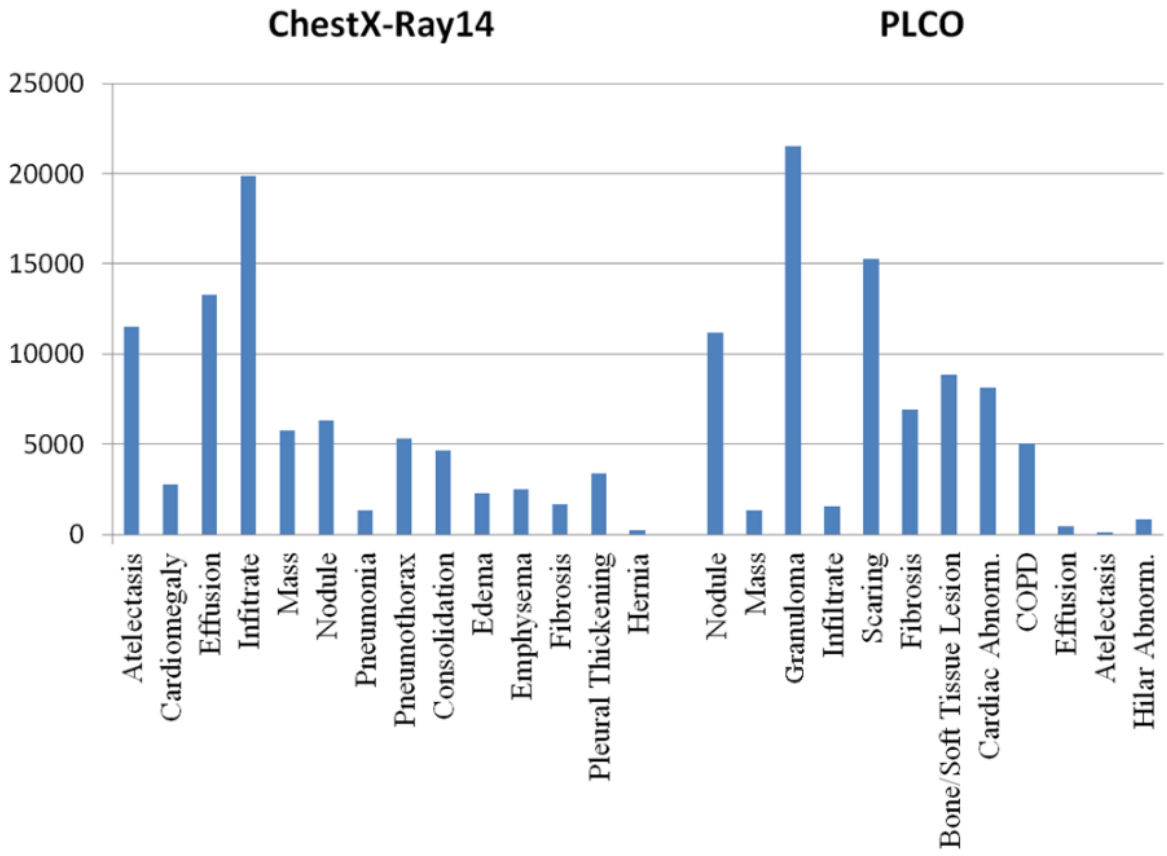
3.3 Learning to recognize Abnormalities in Chest X-Rays with Location-Aware Dense Networks

3.3.1 Data

Unlike previous, similar works, the authors of this paper used PLCO (Gohagan, et al., 2000) dataset in addition to the ChestXRay-14 dataset, a total of 297.541 images of 86.876 patients. From the PLCO dataset, 12 most prevalent labels were chosen in addition the 14 labels of ChestXray-14 dataset. The two datasets share 6 labels with same names, however, for simplicity these classes were treated as different. Also, the authors assumed that there is no patient overlap between the two datasets. All images were normalized to match the ImageNet definition.

Figure 11 shows the image distribution by labels except the 'No Finding'/'Normal' label.

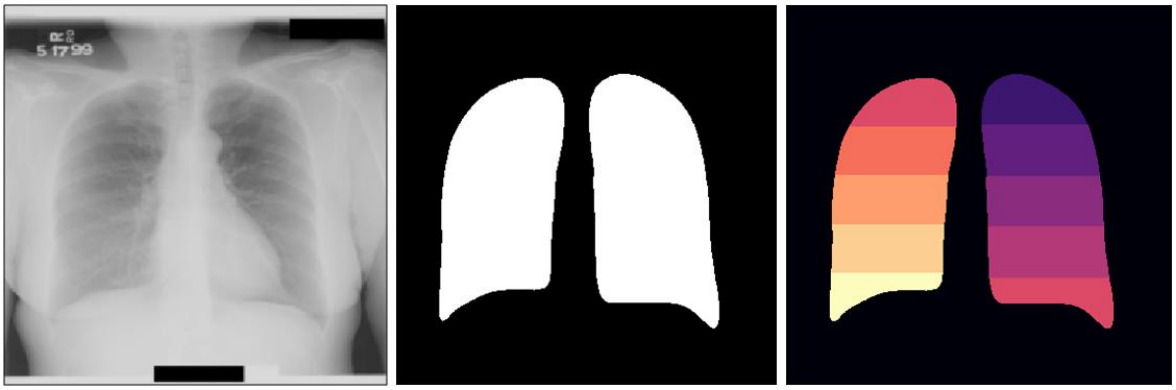
Figure 11. Image distribution by labels except the 'No Finding' label in both datasets.



Source: (Guendel, et al., 2018)

In the PLCO dataset, location information is available for 5 of the 12 pathologies. The location information consists of the information about the side (right or left lung), more detailed localization in each lung (divided by horizontal lines into 5 segments of equal height) and an additional label for diffuse disease. Figure 12 shows an example x-ray image from the PLCO dataset and it's corresponding lung side, lung segmentation information.

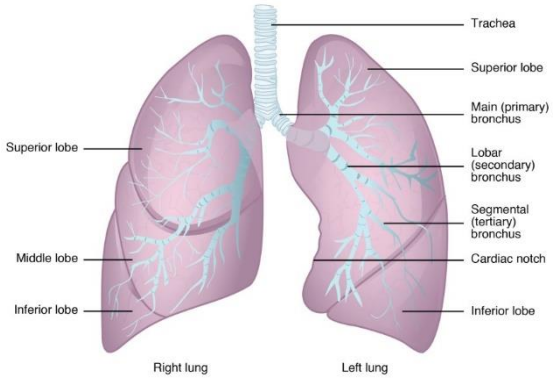
Figure 12. Input x-ray image, and it's corresponding lung side, lung segmentation information.



Source: (Guendel, et al., 2018)

Based on the localization information in the PLCO dataset, the authors created 9 additional classes: 2 for the lung sides, 1 for diffused diseases over multiple lung parts and 5 for each lobe, finally 1 more 'wildcard' label for a presence of pathology in multiple lung parts. Lobes are distinct units of a lung; right lung has 3 lobes, and the left lung has 2 lobes. Figure 13 shows an image of a lung denoting various parts, including the 5 lobes.

Figure 13. Image of a lung denoting various parts, including the 5 lobes.



Source: (Lumen Learning, n.d.)

3.3.2 Model

The authors used a pretrained (on the ImageNet dataset) DenseNet-121 model to classify the images. For each image in the ChestX-ray14 dataset, they assigned a C dimensional binary vector $[l_1, l_2, \dots\ l_C]$ where $C = 14$. They treated the classification problem as 14 independent binary classification problems by defining 14 binary cross entropy loss functions. Due to the high class-imbalance as shown in Figure 11, the authors included additional weights in the loss functions, based on the label frequency within each batch:

$$L(X, l_n) - (w_P * l_n \log(p) + w_N * (1 - l_n) \log(1 - p)) \quad (3)$$

Where $w_P = \frac{P_n + N_n}{P_n}$ and $w_N = \frac{P_n + N_n}{N_n}$, with P_n and N_n indicating the number of positive and negative samples. The model was trained with batch size of 128, the Adam optimizer $(\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8})$ and an adaptive learning rate initialized at 10^{-3} and reduced tenfold when the validation loss plateaus. The authors also split the data in a way to ensure each batch contained images from both datasets.

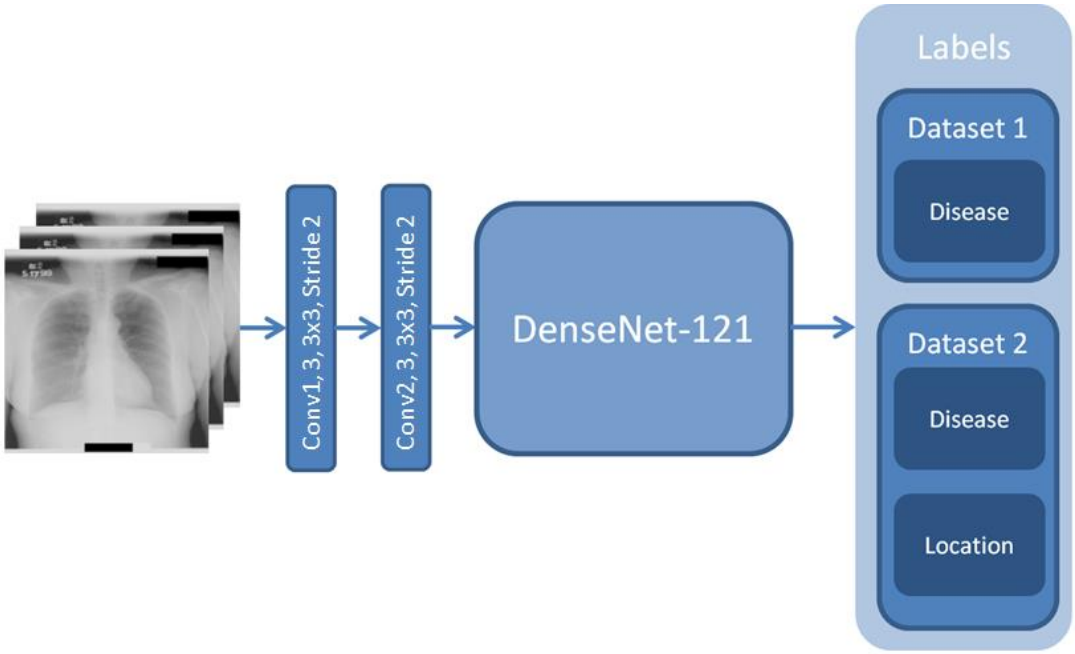
Following loss function was used for the combined dataset with C = 35 classes:

$$L(X) = -\frac{1}{C}\sum_{n=1}^{C} w(w_P * l_n \log(p) + w_N * (1 - l_n) \log(1 - p)) \quad (4)$$

Where w is either 0 or 1, depending on which dataset the image is coming from and whether a spatial information exists.

Figure 14 shows the proposed model architecture.

Figure 14. Model architecture.



Source: (Guendel, et al., 2018)

3.3.3 Experiments and Results

The combined dataset was split patient-wise, 70% for training, 10% for validation and 20% for testing. Figure 15 shows the test results on the two datasets, the table left shows results on the ChestX-ray14 dataset compared to the test result of the original paper (Wang, et al., 2017), the table on the right shows the test results on the PLCO dataset.

DNetLoc model version used the localization information in the PLCO dataset.

The 5 pathologies which the names were highlighted as bold (Nodule, Mass, Infiltrate, Atelectasis, Hilar Abnormality) had localization information.

The performance difference of DNet and DNetLoc models on those 5 pathologies are significant compared to the rest of the pathologies.

Figure 16 shows the corresponding ROC curves of the test results in Figure 15.

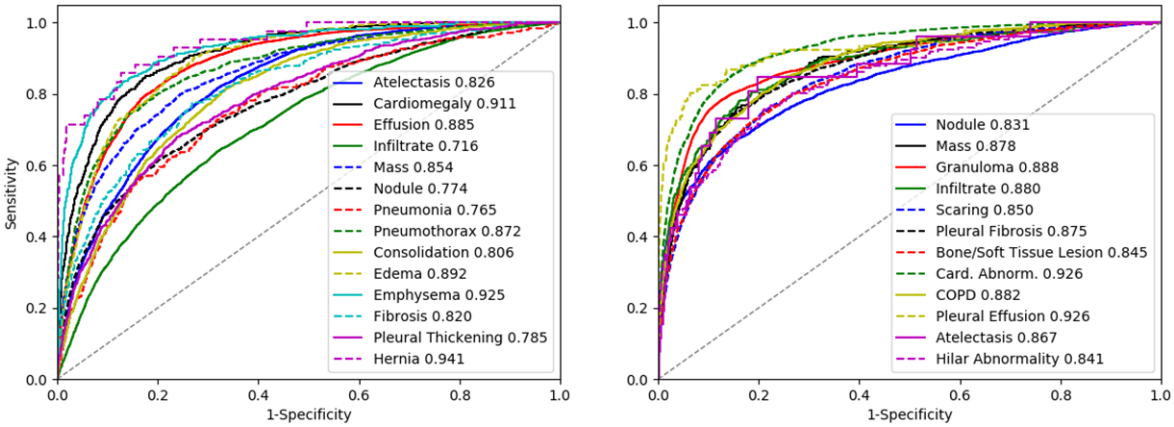
Figure 15. Table of the left shows test results on the ChestX-ray14 dataset, the table on the right shows test results on the PLCO dataset.

Method	Wang *et al.* [1]	Our DNet	Our DNet
Official Split	Yes	Yes	No
Atelectasis	0.7003	0.767	0.826
Cardiomegaly	0.8100	0.883	0.911
Effusion	0.7585	0.828	0.885
Infiltration	0.6614	0.709	0.716
Mass	0.6933	0.821	0.854
Nodule	0.6687	0.758	0.774
Pneumonia	0.6580	0.731	0.765
Pneumothorax	0.7993	0.846	0.872
Consolidation	0.7032	0.745	0.806
Edema	0.8052	0.835	0.892
Emphysema	0.8330	0.895	0.925
Fibrosis	0.7859	0.818	0.820
Pleural Thick.	0.6835	0.761	0.785
Hernia	0.8717	0.896	0.941
Mean	0.7451	0.807	0.841

Method	Our DNet	Our DNetLoc
Nodule	**0.817**	**0.831**
Mass	**0.845**	**0.878**
Granuloma	0.888	0.888
Infiltrate	**0.875**	**0.880**
Scaring	0.841	0.850
Fibrosis	0.873	0.875
Bone/Soft Tissue Lesion	0.853	0.845
Cardiac Abnormality	0.927	0.926
COPD	0.881	0.882
Effusion	0.933	0.926
Atelectasis	**0.831**	**0.867**
Hilar Abnormality	**0.812**	**0.841**
Mean (Location)	**0.836**	**0.859**
Mean	0.865	0.874

Source: (Guendel, et al., 2018)

Figure 16. ROC curves of the corresping tables in Figure 18.



Source: (Guendel, et al., 2018)

3.4 Weakly Supervised Medical Diagnosis and Localization from Multiple Resolutions

This study focuses on localization of Region of Interest (ROI) rather than classification. The authors emphasized the need for image analysis at multiple levels of resolution since thoracic disorders vary greatly in terms of the size and location of ROIs.
For example, cardiomegaly (enlarged heart) is determined to be present if the width of the heart is measured to be 50% or greater than the width of the thoracic cage, this can be detected by looking at the entire X-ray image rather than a localized region.
On the hand, lung nodules are usually as small as few millimetres in size and are often missed by radiologists, thus it is obviously preferable to analyse small, localized regions of an X-ray image to detect lung nodules. The authors further emphasized the importance of localization of ROI because it can immediately draw the attention of practicing radiologists, thus assisting them to provide faster and more accurate diagnosis. (Yao, et al., 2018).
Unlike in (Wang, et al., 2017) and (Rajpurkar, et al., 2017) which generated heatmaps, the authors of this study proposed a model that generates saliency maps in order to visualize the ROIs, to provide radiologists a form of transparency as to why the model made a particular prediction. Saliency map can be considered a form of image segmentation, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as superpixels).
The goal of segmentation is to simplify the representation of an image into something that is easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. (The Wikimedida Foundation, 2020).

3.4.1 Model

The authors proposed a model that can perform localization only from the use of global labels, global label is simply a label for the entire image as opposed to a segment or pixel label. They proposed such a model because medical training data is very hard to label as it often relies on the use of natural language processing to convert historic reports into global labels or employment of radiologists to meticulously read and label each report manually.

Segmentation information is even harder to obtain because the radiologists has to draw segmentations by hand. They framed the problem of weakly supervised classification and localization problem as a multi-instance learning (MIL) problem based on previous similar works. MIL is a type of supervised learning, instead of receiving set of training instances which are individually labelled, the model receives a set of labelled *bags*, each containing many instances. A bag is labelled positive if any of the instances it contains is positive, otherwise negative (The Wikimedia Foundation, 2020). In this case, bags are images and instances in within the bags are image patches.

It seems that the reason why the authors of this study and the authors of similar previous works framed the problem of localization with multi-resolution as MIL is because of how MIL labels bags of instances is very similar to how image segmentation works by assigning a single label/annotation to a segment/set of pixels. But, unlike previous similar works, the authors of this paper used a customized version of the Log-Sum-Exp pooling function with a learnable lower-bounded adaption which they called LSE_LBA to better handle the challenge of localizing pathologies of very different sizes using only image-level/global labels. This allowed the model to generate high-resolution saliency maps without using localization labels. Below function is the modified LSE (LSE_LBA):

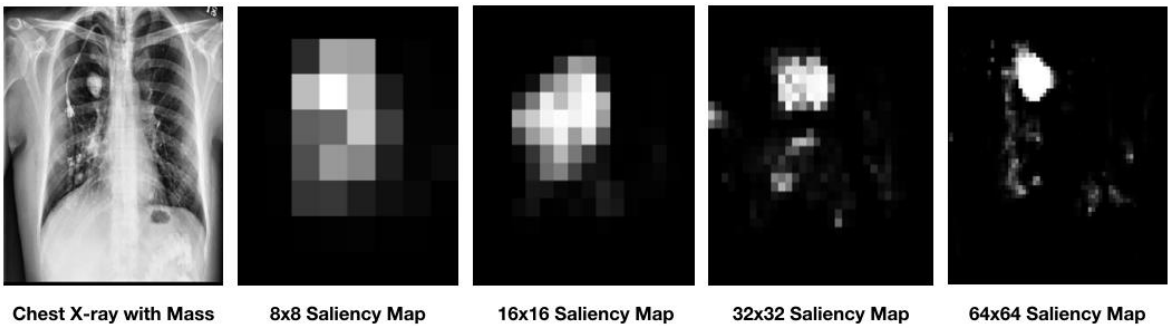
$$LSE\_LBA(S) = \frac{1}{r_0 + e^{\beta}} \log\{\frac{1}{wh}\sum_{i=1}^{w}\sum_{j=1}^{h} e^{[(r_0 + e^{\beta})S_{i,j}]}\} \quad (5)$$

Where S is a saliency map, r_0 is the lower-bound and β is a learnable parameter.

The authors noted that the key difference between their approach and the approaches of previous, similar works is that they specifically trained their model to localize, instead of trying to output localization cues from models trained to classify.

Figure 17 shows an X-ray image of a patient with Mass along with generated saliency maps of increasing resolutions.

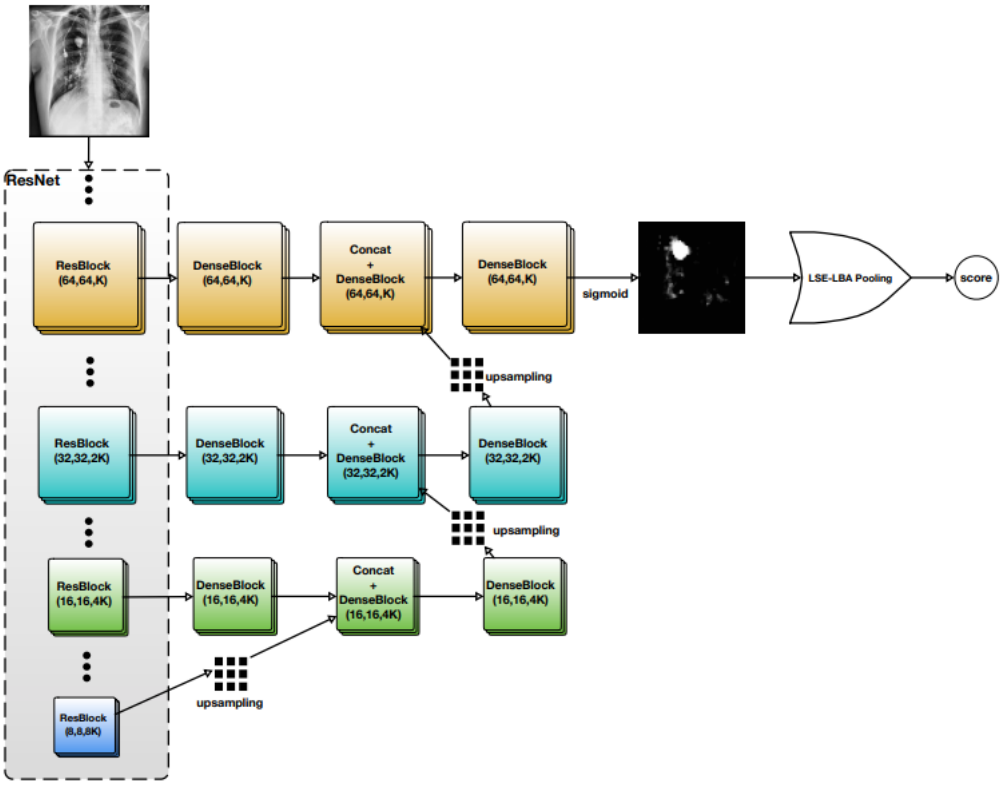
Figure 17. Chest X-ray image and it's multiple saliency map of increasing resolutions



Source: (Yao, et al., 2018)

The proposed model architecture uses ResNet to reduce image resoluton while also using DenseNet at each resolution level to preserve them. The model also uses upsampling (increasing resolution) in order to generate the saliency maps.

Figure 18. Model architecture



Source: (Yao, et al., 2018)

3.4.2 Experiments and Results

The authors applied data augmentation during model training by zooming by factors uniformly sampled from [0.25, 0.75], translating by [-50, 50] pixels (moving the image in one of four directions so that part of the image would be out of frame) and rotating by [-25, 25] degrees. And then normalized to the interval [0, 1] as neural networks work better on normalized/scaled input.

Data augmentation is useful for artificially increasing the size of training dataset if the dataset is small and if a model is trained on such irregular images (zoomed, out of frame and rotated) it would be better at generalizing if it receives similar irregular/poor quality images as an input. The model was trained from scratch with Adam optimizer and early stopping enabled. The team used the AUC metric to evaluate the performance of the classification task and the Dice coefficient for the localization task.

Dice coefficient is a measure of overlap of between 2 images or patches/segments:

$$Dice\ coefficient = 2 * \frac{Area\ of\ overlap}{Total\ number\ of\ pixels\ in\ both\ images} \quad (6)$$

But the exact formula the authors used to calculate the Dice coefficient is:

$$DICE = \frac{2 * S * G}{S^2 + G^2} \quad (7)$$

Where S is the saliency map generated by the model and G is the ground truth binary bounding box with same resolution as the input X-ray image (512 x 512).

Figure 19 shows the test results of three models trained with different values for the r_0, which is the lower-bound of the modified Log-Sum-Exp pooling function LSE-LBA.

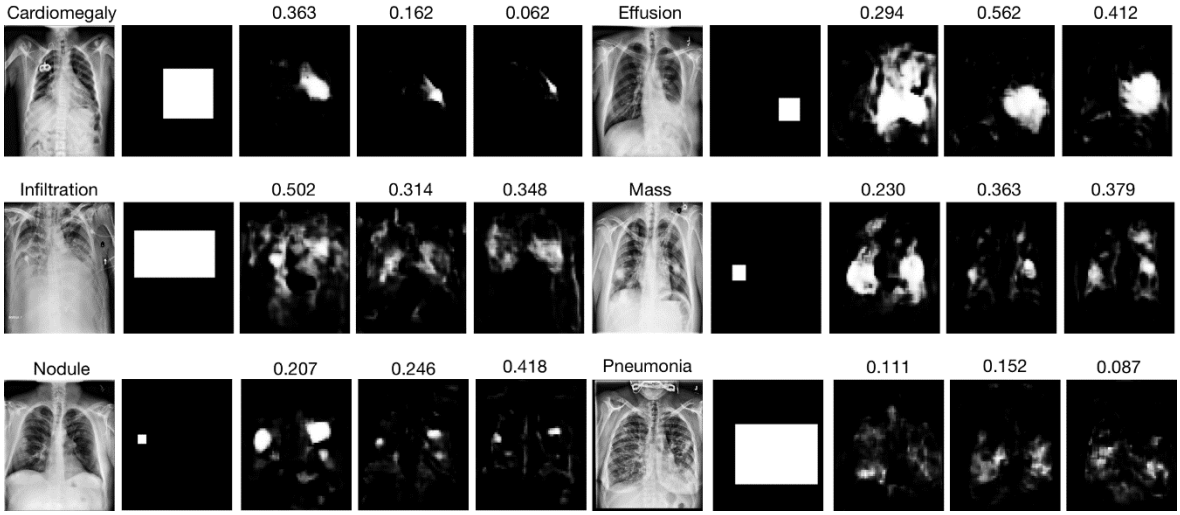
Figure 19. Comparison of test results against previous state-of-the-art model.

	AUC				DICE		
	[3]	$r_0 = 0$	$r_0 = 5$	$r_0 = 10$	$r_0 = 0$	$r_0 = 5$	$r_0 = 10$
Atelectasis	0.7003	**0.733**	0.728	0.724	0.204	**0.240**	0.211
Cardiomegaly	0.8100	0.856	**0.858**	0.854	**0.180**	0.114	0.076
Effusion	0.7585	**0.806**	0.803	0.795	0.293	**0.294**	0.242
Infiltration	0.6614	0.673	**0.675**	0.668	**0.325**	0.312	0.286
Nodule	0.6687	0.718	0.724	**0.727**	0.202	**0.238**	0.196
Mass	0.6933	0.777	0.777	**0.778**	**0.295**	**0.295**	0.241
Pneumonia	0.6580	0.684	**0.690**	0.687	**0.112**	0.104	0.072
Pneumothorax	0.7993	**0.805**	0.791	0.763	**0.039**	0.023	0.028
Consolidation	0.7032	0.711	0.714	**0.717**	-	-	-
Edema	0.8052	**0.806**	0.804	0.801	-	-	-
Emphysema	0.8330	**0.842**	0.822	0.771	-	-	-
Fibrosis	**0.7859**	0.743	0.757	0.731	-	-	-
Pleural thickening	0.6835	**0.724**	0.715	0.712	-	-	-
Hernia	**0.8717**	0.775	0.764	0.824	-	-	-
A.V.G.	0.738	0.761	0.760	0.754	-	-	-

Source: (Yao, et al., 2018)

The combined best results of three versions of the proposed model outperformed the previous state-of-the art model (Wang, et al., 2017) on 9 of the 14 pathologies.
The authors have noticed that AUC is more stable than DICE with respect to different values of r_0. Figure 20 shows example input images with their bounding boxes and saliency maps, the numbers above the saliency maps are the corresponding DICE coefficients.

Figure 20. Example input images with corresponding bounding box and saliency maps of varying DICE coefficients.



Source: (Yao, et al., 2018)

3.5 Comparison of Deep Learning Approaches for Multi-Label Chest X-Ray Classification

In this paper, the authors experimented with ResNet networks of varying depths to classify the ChestX-ray14 dataset (Wang, et al., 2017), as well as building and training a dedicated CNN for X-ray images from scratch. They have also experimented with transfer learning (use of pretrained models) with or without hyperparameter tuning. But what makes this work different from the previously discussed works in this chapter is that the authors also used the non-image data in the dataset such as patient age, gender etc, another difference is that they also performed a cross-validation.

3.5.1 Model

The authors framed the problem as a multi-label classification of 15 classes instead of 14, adding the *No Finding* as a class, thus used a binary vector of size = M for each image label, where M is the number of classes M = 15. After some experiments with different loss functions, the authors decided to use class-averaged binary cross entropy (BCE) as the loss function:

$$\varsigma(\vec{y}, \vec{f}) = \frac{1}{M} \sum_{m=1}^{M} H[y_m, f_m] \text{ where } H[y, f] = -y * logf - (1 - y)\log(1 - f) \quad (8)$$

$\vec{y}$ is the ground truth label and $\vec{f}: \mathrm{X} \rightarrow \Upsilon$ is the objective that minimizes the loss function. The authors modified ResNet-50 architecture by replacing the last dense layer with a new dense layer matching the number of labels (M = 15) and added a sigmoid activation function. Figure 21 shows the comparison of the original ResNet-50 and modified, fine-tuned architectures. As can be seen in Figure 21, the authors fine-tuned/retrained all the convolutional layers. They have also experiment with random weight initialization and pre-trained weights (on the ImageNet dataset).

Figure 21. Comparison of the original and modified, fine-tuned ResNet-50 architectures.

Layer name	Output size	Original 50-layer	Off-the-shelf	Fine-tuned
conv1	112 × 112	7 × 7, 64-d, stride 2	same	**fine-tuned**
pooling1	56 × 56	3 × 3, 64-d, max pool, stride 2	same	same
conv2_x	56 × 56	$\begin{bmatrix} 1\times1, 64\text{-d}, \text{stride}1 \\ 3\times3, 64\text{-d}, \text{stride}1 \\ 1\times1, 256\text{-d}, \text{stride}1 \end{bmatrix}\times3$	same	**fine-tuned**
conv3_0	28 × 28	$\begin{bmatrix} 1\times1, 128\text{-d}, \text{stride}2 \\ 3\times3, 128\text{-d}, \text{stride}1 \\ 1\times1, 512\text{-d}, \text{stride}1 \end{bmatrix}$	same	**fine-tuned**
conv3_x	28 × 28	$\begin{bmatrix} 1\times1, 128\text{-d}, \text{stride}1 \\ 3\times3, 128\text{-d}, \text{stride}1 \\ 1\times1, 512\text{-d}, \text{stride}1 \end{bmatrix}\times3$	same	**fine-tuned**
conv4_0	14 × 14	$\begin{bmatrix} 1\times1, 256\text{-d}, \text{stride}2 \\ 3\times3, 256\text{-d}, \text{stride}1 \\ 1\times1, 1024\text{-d}, \text{stride}1 \end{bmatrix}$	same	**fine-tuned**
conv4_x	14 × 14	$\begin{bmatrix} 1\times1, 256\text{-d}, \text{stride}1 \\ 3\times3, 256\text{-d}, \text{stride}1 \\ 1\times1, 1024\text{-d}, \text{stride}1 \end{bmatrix}\times5$	same	**fine-tuned**
conv5_0	7 × 7	$\begin{bmatrix} 1\times1, 512\text{-d}, \text{stride}2 \\ 3\times3, 512\text{-d}, \text{stride}1 \\ 1\times1, 2048\text{-d}, \text{stride}1 \end{bmatrix}$	same	**fine-tuned**
conv5_x	7 × 7	$\begin{bmatrix} 1\times1, 512\text{-d}, \text{stride}1 \\ 3\times3, 512\text{-d}, \text{stride}1 \\ 1\times1, 2048\text{-d}, \text{stride}1 \end{bmatrix}\times2$	same	**fine-tuned**
pooling2	1 × 1	7 × 7, 2048-d, average pool, stride 1	same	same
dense	1 × 1	1000-d, dense-layer	**15-d, dense-layer**	
loss	1 × 1	1000-d, softmax	**15-d, sigmoid, BCE**	

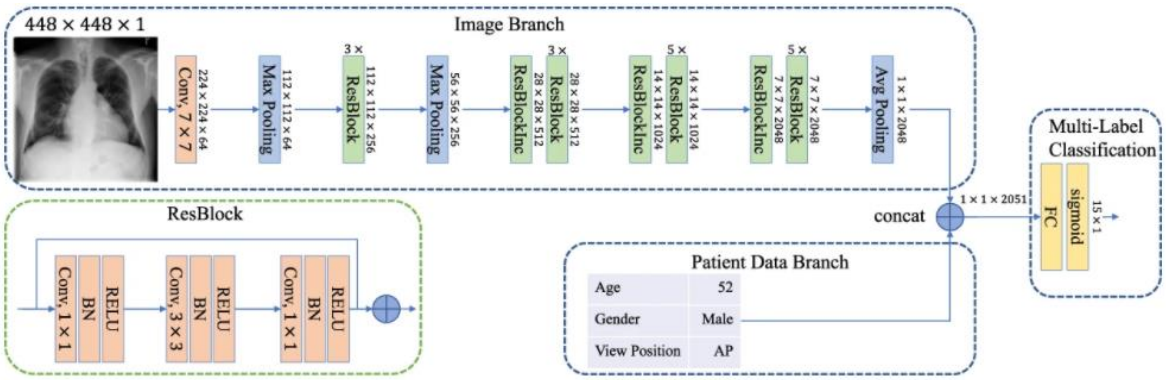
Source: (Baltruschat, et al., 2019)

Aside from the original ResNet-50, the authors also experimented with 2 variants:

- A variant with a reduced input channel to 1 down from 3 (RGB) since ResNet is designed to process RGB images.
- A variant with an increased input size of 448 x 448 from 224 x 224.

They also experimented with ResNet networks of different depths, namely ResNet-38 and ResNet-101 by increasing and decreasing the sizes of convolutional blocks.
Aside from the images, the authors also used three non-image features to further improve their architecture: patient age, gender and whether the x-ray images taken from the front/anterior-posterior (AP) or back/posterior-anterior (PA).
As shown in Figure 22, the non-image feature vector of dimension 3 x 1 was concatenated with the last pooling layer (1x1x2048), resulting in a 1x1x2051 dimensional output. Also, patient age data was scaled to [0, 1]

Figure 22. Architecture of ResNet-50 with non-image features used.



Source: (Baltruschat, et al., 2019)

3.5.2 Experiments and Results

The authors first extended the dataset with data augmentation as in (Szegedy, et al., 2015) which was used in all experiments. During training, image patches of sizes between 8% and 100% of the original image was sampled. They have also used random rotations between [-7, 7] degrees as well as horizontal flipping. Adam optimizer was used with default parameters of $\beta 1 = 0.9$ and $\beta 2 = 0.999$ and learning rates $lr = 0.001$, $lr = 0.01$ for transfer-learning and from scratch respectively.

The authors evaluated eight different model setups and divided them into 3 categories:

- With or without non-image features.
- Transfer learning with off-the-shelf (OTS) and fine-tuned.
- Modified ResNet models with 1-channel or enlarged 448 x 448 input sizes.

AUC values were calculated for all eight model setups along with their standard deviations. Figure 23 shows the results of the experiments by each class/label.

Figure 23. Results of all 8 model setups by each class.

Pathology	Without non-image features				With non-image features			
	OTS	FT	1channel	large	OTS	FT	1channel	large
Cardiomegaly	72.7 ± 1.8	88.5 ± 0.7	88.9 ± 0.5	89.7 ± 0.3	75.9 ± 1.4	88.4 ± 0.8	**90.2 ± 0.4**	89.8 ± 0.8
Emphysema	77.8 ± 2.1	89.2 ± 1.0	87.0 ± 0.8	88.3 ± 1.3	79.8 ± 1.9	**89.4 ± 1.2**	87.4 ± 1.3	89.1 ± 1.2
Edema	84.4 ± 0.6	**89.1 ± 0.4**	**89.1 ± 0.6**	88.8 ± 0.5	85.7 ± 0.5	**89.1 ± 0.7**	89.0 ± 0.6	88.9 ± 0.3
Hernia	78.8 ± 1.4	85.5 ± 3.8	88.1 ± 4.2	87.5 ± 4.5	81.9 ± 2.5	88.2 ± 3.2	89.3 ± 4.4	**89.6 ± 4.4**
Pneumothorax	77.3 ± 1.3	**87.0 ± 0.8**	85.7 ± 0.9	85.9 ± 0.9	79.1 ± 1.2	86.5 ± 0.6	85.4 ± 0.7	85.9 ± 1.1
Effusion	79.4 ± 0.4	87.1 ± 0.2	**87.6 ± 0.2**	**87.6 ± 0.2**	80.6 ± 0.4	87.2 ± 0.3	**87.6 ± 0.2**	87.3 ± 0.3
Mass	66.8 ± 0.6	82.2 ± 1.0	83.3 ± 0.6	**83.9 ± 0.9**	68.6 ± 0.6	82.2 ± 1.0	83.3 ± 0.7	83.2 ± 0.3
Fibrosis	72.0 ± 0.9	**80.0 ± 0.9**	79.9 ± 0.8	79.2 ± 1.6	73.9 ± 0.8	**80.0 ± 0.9**	79.6 ± 0.5	78.9 ± 0.5
Atelectasis	71.8 ± 0.6	**80.3 ± 0.7**	79.9 ± 0.4	79.2 ± 0.7	73.2 ± 0.7	80.1 ± 0.6	79.3 ± 0.6	79.1 ± 0.4
Consolidation	74.3 ± 0.3	79.5 ± 0.5	**80.6 ± 0.4**	80.0 ± 0.3	75.3 ± 0.3	79.6 ± 0.5	80.4 ± 0.5	80.0 ± 0.7
Pleural Thicken.	68.8 ± 1.0	**79.0 ± 0.7**	78.4 ± 0.9	78.0 ± 1.1	70.8 ± 1.1	78.6 ± 1.1	78.2 ± 1.3	77.1 ± 1.3
Nodule	65.0 ± 0.8	72.6 ± 0.9	73.3 ± 0.8	75.1 ± 1.3	66.5 ± 0.7	74.7 ± 0.6	74.0 ± 0.7	**75.8 ± 1.4**
Pneumonia	66.4 ± 2.7	74.4 ± 1.6	74.3 ± 1.5	75.3 ± 2.2	68.3 ± 2.3	73.3 ± 1.3	74.8 ± 1.5	**76.7 ± 1.5**
Infiltration	65.9 ± 0.2	69.9 ± 0.6	**70.2 ± 0.3**	**70.2 ± 0.5**	67.0 ± 0.4	**70.2 ± 0.2**	70.1 ± 0.5	70.0 ± 0.7
Average	73.0 ± 1.1	81.7 ± 1.0	81.9 ± 0.9	82.1 ± 1.2	74.8 ± 1.1	82.0 ± 0.9	82.0 ± 1.0	**82.2 ± 1.1**
No Findings	71.6 ± 0.3	76.9 ± 0.5	**77.3 ± 0.3**	77.1 ± 0.4	72.5 ± 0.3	76.8 ± 0.4	77.1 ± 0.4	77.1 ± 0.3

Source: (Baltruschat, et al., 2019)

As shown in the above image, models with non-image features performed only slightly better than their counterparts without non-image features on average.

The authors trained three more models: ResNet-50-large-age, ResNet-50-large-gender and ResNet-50-large-VP, where VP stands for view position (AP or PA) based on the best performing model ResNet-50-large to predict the age, gender and view position of each training instance.

ResNet-50-large-VP model reached AUC value of 0.9983, ResNet-50-large-gender reached a AUC value 0.9435. Finally, ResNet-50-large-age had a mean absolute error (MAE) of 9.13 ± 7.05 years. This very high AUC values indicate that the image features already encode information about the non-image features, thus the authors speculated that this is the reason why the models with non-image features did yield not reach significant improvement over their counterparts without non-image features.

4 Practical Part

4.1 Data exploration and pre-processing

The metadata of the ChestX-ray14 dataset has 10 columns, the index is a combination of the Patient ID and Follow-up # columns. Figure 24 shows the first five rows from the metadata file.

Figure 24. First five rows from the metadata file.

	Image Index	Finding Labels	Follow-up #	Patient ID	Patient Age	Patient Gender	View Position	OriginalImage[Width	Height]	Origin
0	00000001_000.png	Cardiomegaly	0	1	57	M	PA	2682	2749	
1	00000001_001.png	Cardiomegaly|Emphysema	1	1	58	M	PA	2894	2729	
2	00000001_002.png	Cardiomegaly|Effusion	2	1	58	M	PA	2500	2048	
3	00000002_000.png	No Finding	0	2	80	M	PA	2500	2048	
4	00000003_001.png	Hernia	0	3	74	F	PA	2500	2048	

Pre-processing required extraction of the pathology labels from the 'Finding Labels' column. Figure 25 shows the extracted labels.

Figure 25. Extracted labels.

```
['Atelectasis', 'Cardiomegaly', 'Consolidation', 'Edema', 'Effusion', 'Emphysema', 'Fibrosis', 'H
ernia', 'Infiltration', 'Mass', 'Nodule', 'Pleural_Thickening', 'Pneumonia', 'Pneumothorax']
```

Since the task is to classify pathologies and more than half (about 60 thousand) of the instances are normal, i.e., has 'No Finding' value in the 'Finding Labels' column, these instances were removed. Figure 26 shows the shape of the metadata dataframe after the removal of normal instances, more than half of the total instances were removed (about 60 thousand out of 112 thousand).

Figure 26. Shape of the metadata dataframe after the removal of normal instances..

```
# Drops rows where the Finding Label is empty
df = df[df['Finding Labels'] != '']
# Sets Image Index column as index
df.set_index('Image Index', inplace=True)
df.shape
```

```
(51759, 10)
```

Figure 27 shows that the reduced dataset is comprised of x-ray images from 14402 unique patients compared to the 30805 unique patients in the original, full dataset.

Figure 27. Number of unique patients in the reduced dataset.

```
# Number of unique patients in the dataset after the removal of 'No Finding' rows.
unique_patients = df['Patient ID'].unique()
print(len(unique_patients))
```

```
14402
```

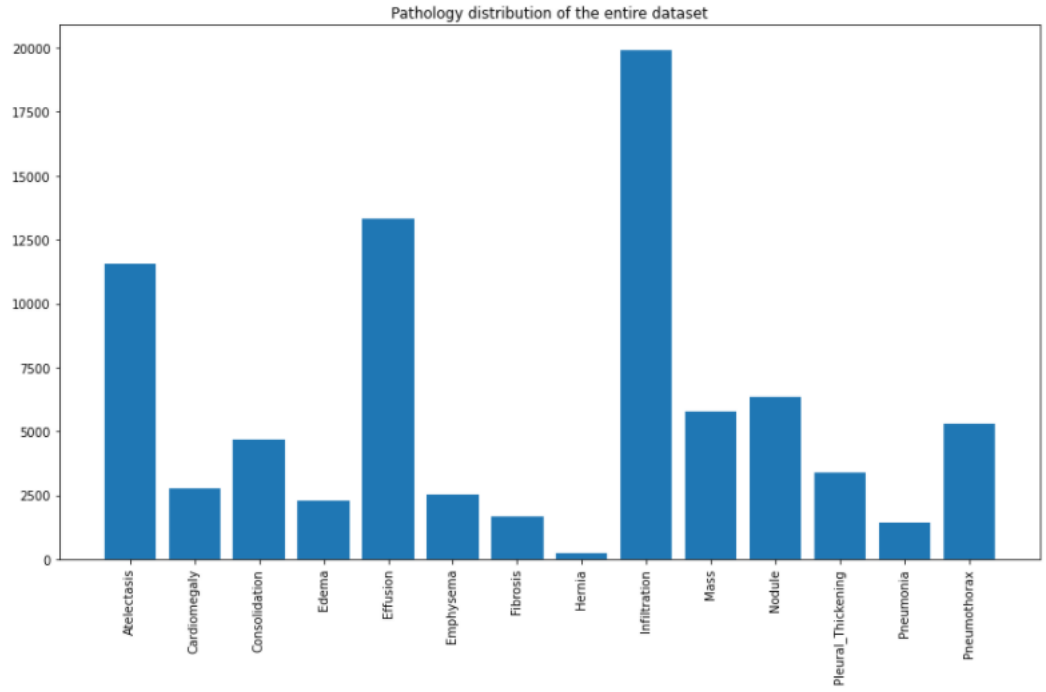
Figure 28 shows the number of instances for each pathology/label in the dataset.

Figure 28. Number of instances for each pathology.

```
{'Atelectasis': 11559, 'Cardiomegaly': 2776, 'Consolidation': 4667, 'Edema': 2303, 'Effusion': 13
317, 'Emphysema': 2516, 'Fibrosis': 1686, 'Hernia': 227, 'Infiltration': 19894, 'Mass': 5782, 'No
dule': 6331, 'Pleural_Thickening': 3385, 'Pneumonia': 1431, 'Pneumothorax': 5302}
```

Figure 29 Illustrates the pathology/label distribution in the remaining 51759 instances shown in Figure 28

Figure 29. Label distribution of the remaining 51759 instances.



Above image shows a high class-imbalance in the dataset and this fact should be considered when training a model.

According to the authors of the ChestX-ray14 dataset, the official train_val and test splits are patient-wise, Figure 30 shows that there is indeed no patient overlap between the two splits.

Figure 30. Confirming that there is no patient overlap between the train_val and test splits.

```
# Checks for a patient overlap between 2 subsets
def check_overlap(df1, df2):
    patients1 = df1.index.map(lambda x: x.split('_')[0])
    patients2 = df2.index.map(lambda x: x.split('_')[0])
    return list(set(patients1) & set(patients2))
```

```
# Checking whether the official split: train_val and test are indeed patient-wise
overlap = check_overlap(train_val_df, test_df)
print('Number of patients in both subsets', len(overlap))
```

```
Number of patients in both subsets 0
```

Figure 31 shows the shapes of train_val and test split dataframes after the exclusion of normal ('No Finding') instances.

Figure 31. Number of instances in the train_val and test splits after the removal of normal instances.

```
# Drops instances from the df1 set that are not in the df2 set
def drop_differences(df1, df2):
    df2.drop(df2.iloc[:, 6:11], inplace = True, axis = 1)
    df3 = df1[df1.index.isin(df2.index)]
    return df3

# Drops instances from the train_val and test sets that has 'No Finding' labels.
train_val_df = drop_differences(train_val_df, df)
test_df = drop_differences(test_df, df)
print('train_val_df:',train_val_df.shape, 'test_df:', test_df.shape)
```

```
train_val_df: (36024, 1) test_df: (15735, 1)
```

Figure 32 shows the label distribution in the train_val and test splits.

Figure 32. Label distribution in the train_val and test splits.

```
# Dictionary for pathology: occurence from the train_val set
train_val_label_counts = dict()
train_val_label_counts = count_occurences(labels, train_val_df)
print('Training and validation set label counts:', train_val_label_counts)
```

```
Training and validation set label counts: {'Atelectasis': 8280, 'Cardiomegaly': 1707, 'Consolidat
ion': 2852, 'Edema': 1378, 'Effusion': 8659, 'Emphysema': 1423, 'Fibrosis': 1251, 'Hernia': 141,
'Infiltration': 13782, 'Mass': 4034, 'Nodule': 4708, 'Pleural_Thickening': 2242, 'Pneumonia': 87
6, 'Pneumothorax': 2637}
```

```
# Dictionary for pathology: occurence from the test set
test_label_counts = dict()
test_label_counts = count_occurences(labels, test_df)
print('Test set label counts:', test_label_counts)
```

```
Test set label counts: {'Atelectasis': 3279, 'Cardiomegaly': 1069, 'Consolidation': 1815, 'Edem
a': 925, 'Effusion': 4658, 'Emphysema': 1093, 'Fibrosis': 435, 'Hernia': 86, 'Infiltration': 611
2, 'Mass': 1748, 'Nodule': 1623, 'Pleural_Thickening': 1143, 'Pneumonia': 555, 'Pneumothorax': 26
65}
```

Although the train_val and test subsets were split by patient, the label distributions within these splits are not too different. Figure 33 shows the comparison of label distribution within the train_val and test splits.

Figure 33. Label distribution comparison in the train_val and test splits.

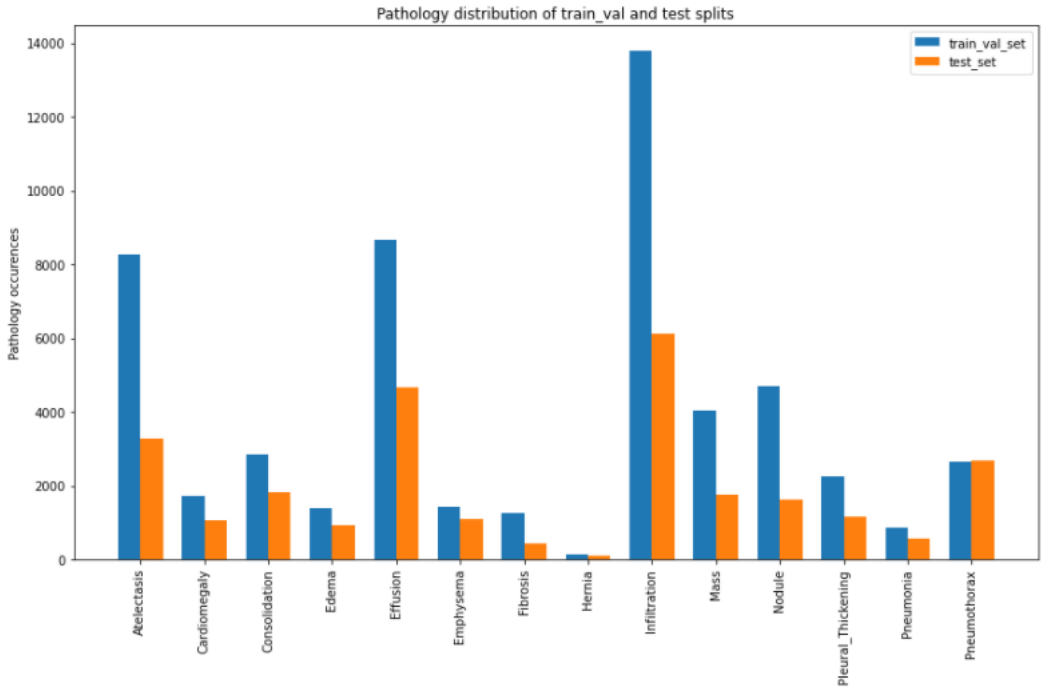


Figure 34 shows the comparison of label proportions in the train_val and test splits by each label and the error percentages are quite high because the split was done patient-wise. Especially, Pneumothorax has very high error percentage which can also be seen in Figure 33

Figure 34. Label-wise comparison of train_val and test splits.

	Pathology	Entire dataset	Training and validation split	Test split	Training and validation split error %	Test split error %
0	Atelectasis	0.142394	0.153419	0.120525	7.742067	-15.358353
1	Cardiomegaly	0.034197	0.031629	0.039293	-7.511169	14.900309
2	Consolidation	0.057492	0.052844	0.066713	-8.084882	16.038413
3	Edema	0.028370	0.025533	0.034000	-10.002487	19.842469
4	Effusion	0.164051	0.160441	0.171212	-2.200515	4.365279
5	Emphysema	0.030994	0.026366	0.040175	-14.931359	29.620138
6	Fibrosis	0.020770	0.023180	0.015989	11.602769	-23.017034
7	Hernia	0.002796	0.002613	0.003161	-6.573843	13.040885
8	Infiltration	0.245072	0.255364	0.224656	4.199434	-8.330643
9	Mass	0.071228	0.074745	0.064251	4.938061	-9.795896
10	Nodule	0.077991	0.087234	0.059656	11.850871	-23.509208
11	Pleural_Thickening	0.041700	0.041542	0.042013	-0.378709	0.751266
12	Pneumonia	0.017628	0.016231	0.020400	-7.925429	15.722099
13	Pneumothorax	0.065315	0.048860	0.097956	-25.192415	49.975543

Since the task is a multi-label classification, meaning each instance can belong to one or more categories, I have added 14 new columns to the dataframes to indicate the presence (1) or absence (0) of each pathology. Figure 35 illustrates the first five rows of the train_val dataframe, showing ones or zeros indicating presence or absence of the corresponding pathologies.

Figure 35. Dataframe of train_val split after adding one column for each label.

```
train_val_df.head()
```

	Finding Labels	Follow-up #	Patient ID	Patient Age	Patient Gender	View Position	Atelectasis	Cardiomegaly	Consolidation	...	Effusion	Emp
	Cardiomegaly	0	1	57	M	PA	0	1	0	...	0	
	Cardiomegaly|Emphysema	1	1	58	M	PA	0	1	0	...	0	
	Cardiomegaly|Effusion	2	1	58	M	PA	0	1	0	...	1	
	Mass|Nodule	0	4	82	M	AP	0	0	0	...	0	
	Infiltration	6	5	70	F	PA	0	0	0	...	0	

The train_val subset was further split into training and validation, but by label. The split was done in a stratified manner to keep the label proportions as identical as possible with the train_val set. Figure 36 shows the result of this split, the training set has 28819 instances and the validation split has 7205 instances.

Figure 36. Training and validation splits.

```
from sklearn.model_selection import train_test_split

# Stratified label-wise split of the train_val set with a 4:1 ratio, using the first 4 characters
train_df, valid_df = train_test_split(train_val_df, test_size = 0.2, random_state = 42,
                                      stratify = train_val_df['Finding Labels'].map(lambda x: x.sp

print('Training split by label', train_df.shape, 'Validation split by label', valid_df.shape)
```

```
Training split by label (28819, 21) Validation split by label (7205, 21)
```

Figure 37 shows the compared proportion of label distributions in the training and validation splits. The label distribution proportions are much more similar compared to Figure 33

Figure 37. Comparison of the label distributions in the traning and validation splits.

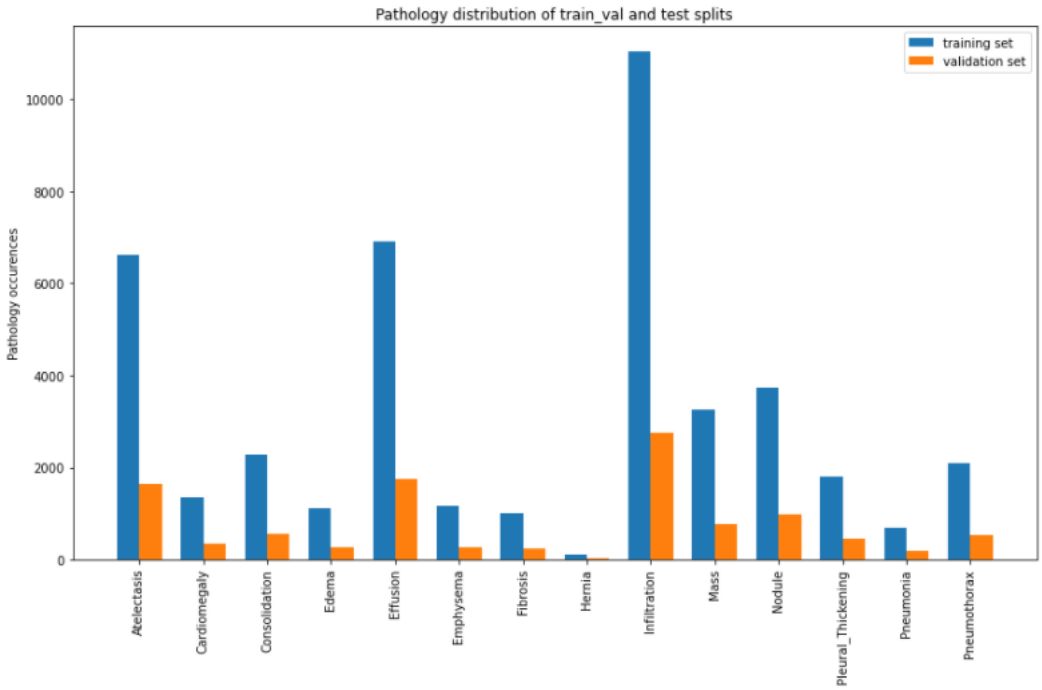


Figure 38 shows the comparison of the proportions of label distribution in the training and validation splits. The error percentages in training and validation splits are much lower compared to those in train_val and test splits shown in Figure 34

Figure 38. Comparison of the proportions of label distribution in the training and validation splits.

	Pathology	Training and validation split	Training split by label	Validation split by label	Training split by label-error %	Validation split by label-error %
0	Atelectasis	0.153419	0.153287	0.153946	-0.085622	0.343962
1	Cardiomegaly	0.031629	0.031518	0.032072	-0.349017	1.402071
2	Consolidation	0.052844	0.052924	0.052524	0.150851	-0.605998
3	Edema	0.025533	0.025548	0.025472	0.059391	-0.238586
4	Effusion	0.160441	0.160068	0.161941	-0.232742	0.934971
5	Emphysema	0.026366	0.026705	0.025007	1.283549	-5.156269
6	Fibrosis	0.023180	0.023442	0.022125	1.132359	-4.548909
7	Hernia	0.002613	0.002731	0.002138	4.520360	-18.159180
8	Infiltration	0.255364	0.255409	0.255183	0.017685	-0.071043
9	Mass	0.074745	0.075325	0.072418	0.775068	-3.113601
10	Nodule	0.087234	0.086479	0.090267	-0.865540	3.477047
11	Pleural_Thickening	0.041542	0.041585	0.041368	0.103778	-0.416898
12	Pneumonia	0.016231	0.016245	0.016176	0.085464	-0.343325
13	Pneumothorax	0.048860	0.048735	0.049363	-0.256125	1.028904

4.2 Data Augmentation

Data Augmentation is a technique used very often in computer vision tasks to artificially increase the number of training instances. But the increase of training instances is not the only purpose of Data Augmentation as it also adds random augmentations/imperfections to the training images that can probably be encountered in real-life datasets. Learning from such imperfect images prepares models if and when it encounters similar images.

I have used Keras API's ImageDataGenerator class to create augmented images, below are the augmentations and their values I have chosen:

- horizontal_flip = True

  Flips the image along the horizontal axis, this parameter was set True because there are two types of X-ray images: AP and PA, one taken from the front of a patient's and other from the back.
- vertical_flip = False

  Flips the image along the vertical axis, it is set to False because the model is very unlikely to encounter an upside down X-ray image.
- height_shift_range = 0.05

  Shifts the image vertically either up or down by a random amount between 0 and 5 percent of the image's height, creating an empty region above or below the image.
- width_shift_range = 0.01

  Works the same way as the height_shift_range, except horizontally.
- rotation_range = 10

  Rotates the image along the vertical axis by a random amount between 0 and 10 degree angle, this augmentation was chosen because patient's chest might not be perfectly parallel to the X-ray machine.
- fill_mode = 'constant'

  Fills up any empty region, for example caused by width_shift_range, height_shift_range. I have chosen the 'constant' value along with cval = 0 to fill the empty regions by solid black pixels. The default value is 'nearest',

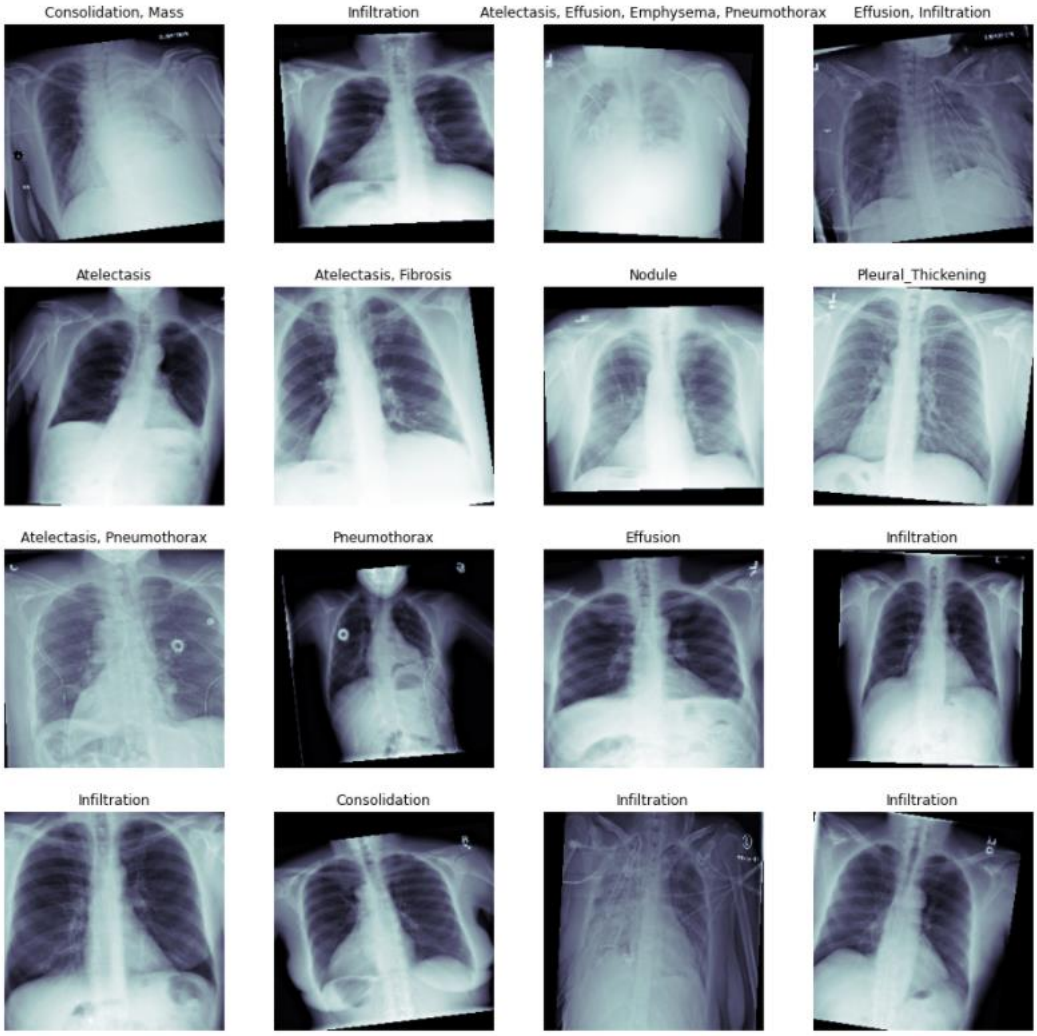
which fills empty regions by copying its nearest non-empty pixels, but this was not suitable for X-ray images.

- zoom_range = 0.2

  Zooms in and out of the target image, the 0.2 value means 20% percent zoom.

Figure 39 shows examples of randomly augmented images with their labels.

Figure 39. Examples of augmented images.



4.3 Models

I have used models based on two (ResNet50 and MobileNet) out-of-the-box models from the keras.applications package for classification. And I used the base models without pre-trained weights on the ImageNet dataset because X-ray images are not part of ImageNet's category list. The models were trained with Adam as optimizer, binary cross-entropy function as the loss function and binary accuracy, mean absolute error as metrics.
The models were first trained, validated and tested on the original images (without augmentation) and then further trained, validated, tested on augmented images.
Figure 40 and Figure 41 shows the summaries of the models based on ResNet50 and MobileNet respectively.

Figure 40. Summary of the ResNet50-based model.

```
resnet50 = create_resnet50()
resnet50.compile(optimizer='adam', loss = 'binary_crossentropy',
                 metrics = ['binary_accuracy', 'mae'])

resnet50.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Functional)        (None, 7, 7, 2048)        23581440
_________________________________________________________________
global_average_pooling2d (Gl (None, 2048)              0
_________________________________________________________________
dropout (Dropout)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 224)               458976
_________________________________________________________________
dropout_1 (Dropout)          (None, 224)               0
_________________________________________________________________
dense_1 (Dense)              (None, 14)                3150
=================================================================
Total params: 24,043,566
Trainable params: 23,990,446
Non-trainable params: 53,120
_________________________________________________________________
```

Figure 41. Summary of the MobileNet-based model.

```
mobilenet.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
mobilenet_1.00_224 (Function (None, 7, 7, 1024)        3228288
_________________________________________________________________
global_average_pooling2d_2 ( (None, 1024)              0
_________________________________________________________________
dropout_4 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_4 (Dense)              (None, 224)               229600
_________________________________________________________________
dropout_5 (Dropout)          (None, 224)               0
_________________________________________________________________
dense_5 (Dense)              (None, 14)                3150
=================================================================
Total params: 3,461,038
Trainable params: 3,439,150
Non-trainable params: 21,888
_________________________________________________________________
```

5 Results and Discussion

Both models were trained for 50 epochs with early stopping, model checkpoint and the patience parameter set to 5.

5.1 ResNet50-based model

Figure 42 and Figure 43 shows the learning and ROC curves of the ResNet50-based model trained on non-augmented images respectively.
Figure 44 and Figure 45 shows the learning and ROC curves of the ResNet50-based model trained on augmented images respectively.
Comparison of the two ROC curves shows that the model accuracy was improved after the training on augmented images on almost all 14 categories, except Pneumonia.
From the learning curve of the second training (on augmented images), one can see that both training loss and training accuracy have not plateaued, training loss is decreasing, and accuracy is increasing, which indicates the model probably could have been improved with further training on more data or by increasing the number of epochs and the patience parameter.

Figure 42. Learning curves of the ResNet50-based model on non-augmented images.

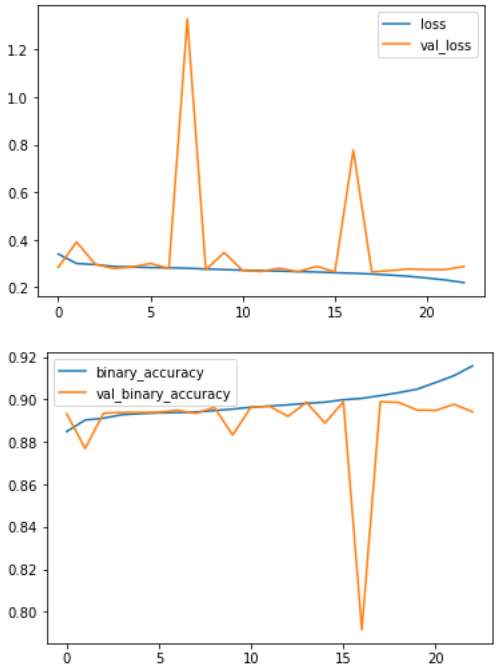


Figure 43. ROC curve of the ResNet50-based model trained on non-augmented images.

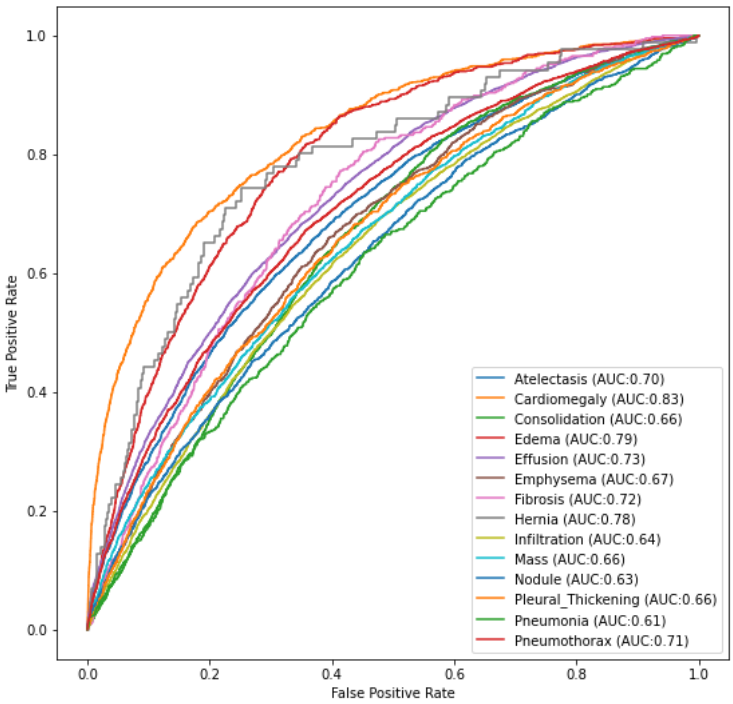


Figure 44. Learning curves of the ResNet50-based model trained on augmented images.

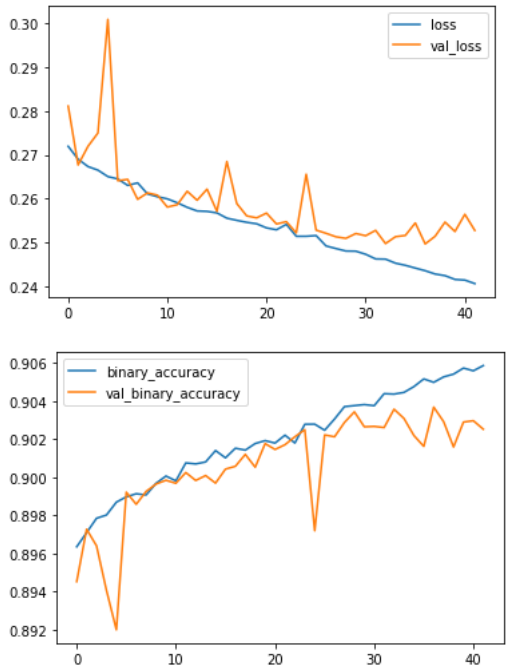
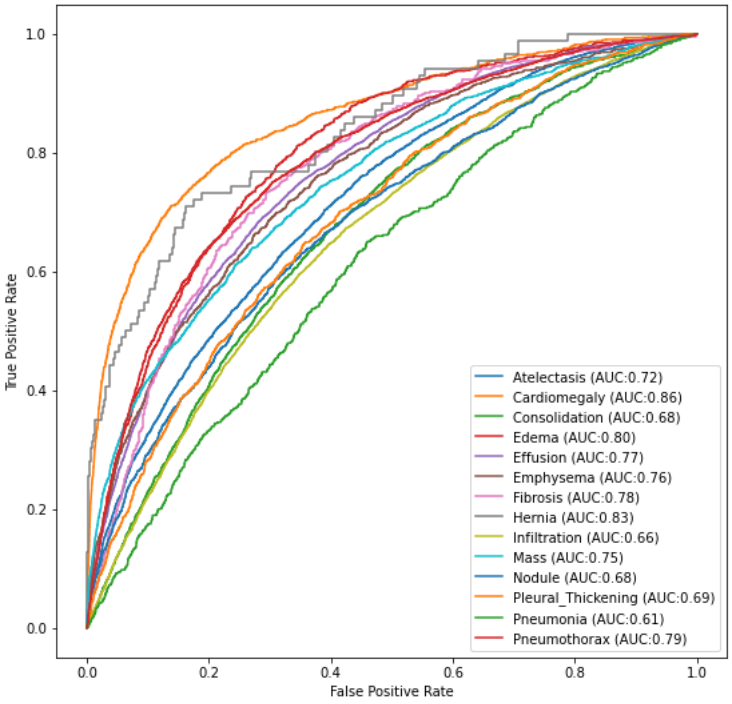


Figure 45. ROC curve of the ResNet50-based model trained on augmented images.



5.2 MobileNet-based model

Figure 46 and Figure 47 shows the learning and ROC curves of the ResNet50-based model trained on non-augmented images respectively.

Figure 48 and Figure 49 shows the learning and ROC curves of the ResNet50-based model trained on augmented images respectively.

Comparison of the two ROC curves shows that the model accuracy was improved after the training on augmented images on all 14 categories.

From the learning curve of the second training (on augmented images), one can see that both training loss and training accuracy have not plateaued, training loss is decreasing, and accuracy is increasing, which indicates the model probably could have been improved with further training on more data or by increasing the number of epochs and the patience parameter.

Figure 46. Learning curves of the MobileNet-based model trained on non-augmented images.

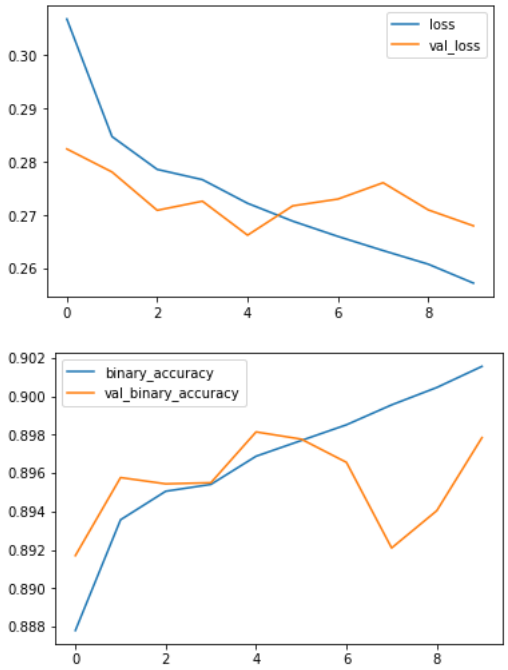


Figure 47. ROC curve of the MobileNet-based model trained on non-augmented images.

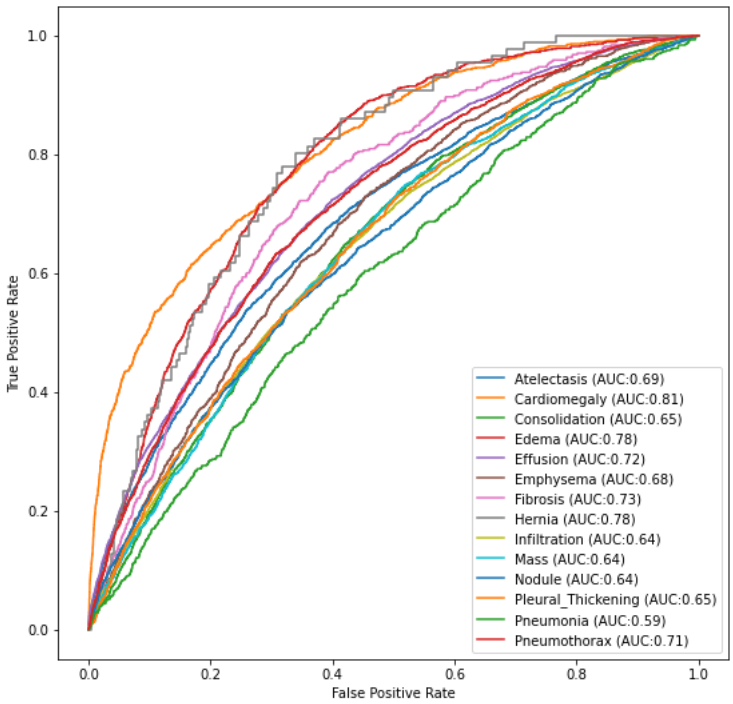


Figure 48. Learning curves of the MobileNet-based model trained on augmented images.

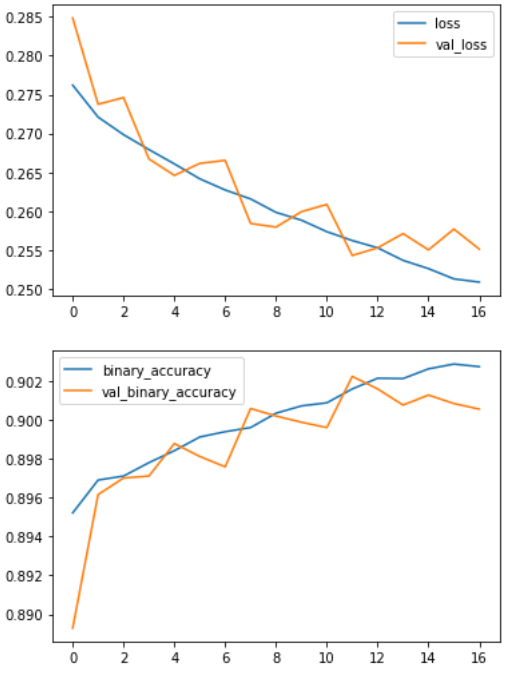
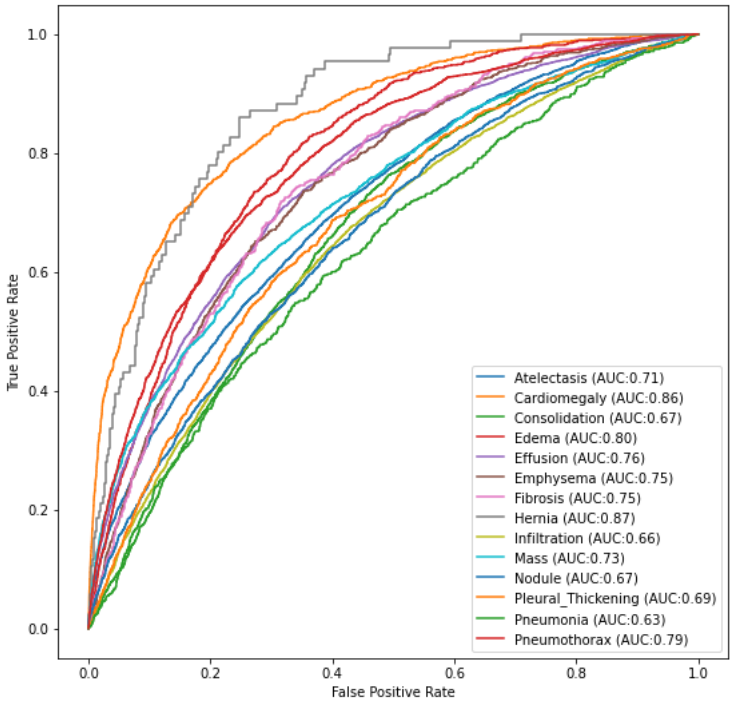


Figure 49. ROC curve of the MobileNet-based model trained on augmented images.



5.3 Voting Classifiers

Three voting classifiers: Max Vote, Hard Vote and Simple Average has been used on the AUC scores of both models. Surprisingly, all three classifier AUC scores were lower compared to individual model scores. Figure 50 shows the comparison table.

Figure 50. AUC score comparison of both models and three voting classifiers.

	Class	Resnet50	Mobilenet	Max Vote Ensemble	Hard Vote Ensemble	Avg Ensemble
0	Atelectasis	0.719	0.710	0.647	0.555	0.651
1	Cardiomegaly	0.857	0.859	0.803	0.595	0.802
2	Consolidation	0.678	0.669	0.622	0.500	0.631
3	Edema	0.802	0.799	0.727	0.498	0.728
4	Effusion	0.769	0.757	0.679	0.633	0.691
5	Emphysema	0.760	0.748	0.684	0.500	0.679
6	Fibrosis	0.775	0.752	0.706	0.500	0.704
7	Hernia	0.834	0.871	0.829	0.500	0.829
8	Infiltration	0.662	0.659	0.567	0.569	0.583
9	Mass	0.751	0.725	0.674	0.546	0.670
10	Nodule	0.683	0.665	0.633	0.500	0.619
11	Pleural_Thickening	0.693	0.685	0.626	0.500	0.626
12	Pneumonia	0.614	0.634	0.617	0.500	0.604
13	Pneumothorax	0.789	0.789	0.735	0.530	0.731

6 Conclusion

Thorax diseases account for a significant proportion of global deaths every year, pneumonia alone kills millions of people annually. People in developing countries are especially vulnerable compared to the people in developed countries due to variety of factors such as: poor healthcare system, air pollution and lack of medical professionals. Developing countries tend to have a more severe lack of medical professionals because doctors and nurses often leave their home countries to seek higher paying jobs in developed countries. X-ray imaging is the most common method used for diagnosing thorax diseases due to its low cost compared to other methods such as Computed Axial Tomography (CAT) and Magnetic Resonance Imaging (MRI). But radiologists are in short supply, same as other medical professionals. Thus, a deep learning-based diagnostic tool can be used make up for the lack of radiologists if the diagnostic accuracy of such a tool is comparable to that of a certified and experienced radiologist.

7 References

Baltruschat, I. M. et al., 2019. *Comparison of Deep Learning Approaches for Multi-Label Chest X-Ray Classification.* [Online]
Available at: https://www.nature.com/articles/s41598-019-42294-8#Tab4
European Commission, 2012. *COMMISSION STAFF WORKING DOCUMENT on an Action Plan for the EU Health Workforce,* Strasbourg: European Commission.
Garyfallos, S., Biseda, B. & Khan, M., 2019. *NIH-Chest-X-rays-Classification,* Berkeley: s.n.
Gohagan, J. K., Prorok, P. C. H. R. B. & Kramer, B. S., 2000. *The Prostate, Lung, Colorectal and Ovarian (PLCO) Cancer Screening Trial of the National Cancer Institute: history, organization, and status.* [Online]
Available at: https://pubmed.ncbi.nlm.nih.gov/11189683/
Guendel, S. et al., 2018. *Learning to recognize Abnormalities in Chest X-Rays with Location-Aware Dense Networks.* [Online]
Available at: https://arxiv.org/abs/1803.04565
Lumen Learning, n.d. *The Lungs | Anatomy and Physiology ||.* [Online]
Available at: https://courses.lumenlearning.com/suny-ap2/chapter/the-lungs/
Rajpurkar, P. et al., 2017. *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays.* [Online]
Available at: https://arxiv.org/abs/1711.05225
Szegedy, C. et al., 2015. *Going deeper with convolutions.* [Online]
Available at: https://ieeexplore.ieee.org/document/7298594
telemedicineclinic, 2016. [Online]
Available at: https://www.telemedicineclinic.com/wp-content/uploads/2016/11/Europes_looming_radiology_capacity_challenge-A_comparitive_study.pdf
The Wikimedia Foundation, 2020. *Multiple instance learning - Wikipedia.* [Online]
Available at: https://en.wikipedia.org/wiki/Multiple_instance_learning
The Wikimedida Foundation, 2020. *Saliency map - Wikipedia.* [Online]
Available at: https://en.wikipedia.org/wiki/Saliency_map
Wang, W. et al., 2020. *A Novel Image Classification Approach via Dense-MobileNet Models.* [Online]
Available at: https://www.hindawi.com/journals/misy/2020/7602384/
Wang, X. et al., 2017. *ARXIV_V5_CHESTXRAY.pdf.* [Online]
Available at: https://nihcc.app.box.com/v/ChestXray-NIHCC/file/256057377774
Wang, X. et al., 2017. *ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Thorax Diseases.* [Online]
Available at: https://arxiv.org/abs/1705.02315
World Health Organization, 2014. *7 million premature deaths annually linked to air pollution,* Geneva: World Health Organization.
Yao, L. et al., 2018. *Weakly Supervised Medical Diagnosis and Localization from Multiple Resolutions.* [Online]
Available at: https://arxiv.org/abs/1803.07703

8 Bibliography

Brownlee, J., 2019. *A Gentle Introduction to Channels-First and Channels-Last Image Formats.* [Online]
Available at: https://machinelearningmastery.com/a-gentle-introduction-to-channels-first-and-channels-last-image-formats-for-deep-learning/

Brownlee, J., 2019. *Best Practices for Preparing and Augmenting Image Data for CNNs.* [Online]
Available at: https://machinelearningmastery.com/best-practices-for-preparing-and-augmenting-image-data-for-convolutional-neural-networks/

Brownlee, J., 2019. *Dropout Regularization in Deep Learning Models With Keras.* [Online]
Available at: https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/

Brownlee, J., 2019. *How to Configure Image Data Augmentation in Keras.* [Online]
Available at: https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/

Brownlee, J., 2019. *How to Load, Convert, and Save Images With the Keras API.* [Online]
Available at: https://machinelearningmastery.com/how-to-load-convert-and-save-images-with-the-keras-api/

Brownlee, J., 2019. *How to Normalize, Center, and Standardize Image Pixels in Keras.* [Online]
Available at: https://machinelearningmastery.com/how-to-normalize-center-and-standardize-images-with-the-imagedatagenerator-in-keras/

Brownlee, J., 2019. *How to use Learning Curves to Diagnose Machine Learning Model Performance.* [Online]
Available at: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/

Brownlee, J., 2020. *Multi-Label Classification with Deep Learning.* [Online]
Available at: https://machinelearningmastery.com/multi-label-classification-with-deep-learning/#:~:text=Each%20node%20in%20the%20output,value%20between%200%20and%201.

Brownlee, J., 2021. *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning.* [Online]
Available at: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/#:~:text=Adam%20is%20different%20to%20classical,does%20not%20change%20during%20training.

Géron, A., 2019. *Hands-on Machine Learning with Scikit-Learn, Keras and Tensorflow.* 2nd ed. Sebastopol: O'Reilly Media, Inc..

List of Supplements…

1. Thesis.ipynb