Multi-Label Classification and Visual Explanation of Chest X-ray Images using Neural Networks with Attention Mechanism and Grad-CAM

Marcus Tan, Xiaohan Tian, Wing Chan, Joshua Ceaser

University of Illinois, Urbana-Champaign

ABSTRACT

We develop a method to select a subset of images from the full Chest-Xray14 data set to allow for the training of deep learning models with limited computing resources. The resulting subset contains approximately 20% of the original images. To determine if the subset of images is sufficiently large to provide meaningful results, we compute AUROC scores and compare the performance of the DenseNet-121 and ResNet-50 models used in the CheXNet study. Average validation and test AUROC scores as high as 0.7 or greater can be achieved with those models with pre-trained weights. The superior performance of DenseNet-121 over ResNet-50 shown in the CheXNet study is reproduced with the smaller data set. Two variants of an attention mechanism (DenseNet-121-attA and DenseNet-121-attB) are added to the DenseNet-121 model and shown to improve the test AUROC between 0.02 and 0.03 when all models were trained with a learning rate of 0.01 for 8 epochs. Visual explanation of model prediction is provided with heat maps generated using the Grad-CAM method.

1. INTRODUCTION AND BACKGROUND

Chest X-rays are widely used as a diagnostic method to examine a patient's current respiratory system for illnesses and prioritize patient care [3]. Due to the large number of images and limited access to experienced radiologists who can interpret the images accurately, deep learning models have been used on the images to assist in determining the presence of a disease, and if a disease is present, to identify the disease and highlight the diseased region [12].

1.1 Literature Review

Existing literature predominantly uses three datasets when classifying chest X-rays: ChestX-ray14 [18], CheXpert [9] and JSRT [16]. ChestX-ray14 was originally called ChestX-ray8 as there were 8 disease labels, but was renamed when an additional 6 disease classes were added. The dataset was created by Wang et al. using radiology reports and images from the Picture Archiving and Communication systems of NIH [18]. Reports with keywords corresponding to 8 common thoracic pathologies and the corresponding images were extracted from the system. Natural Language Processing (NLP) techniques were applied on the reports to obtain the disease classifications for the images. AlexNet, GoogLeNet, VGGNet-16 and ResNet-50 models pre-trained on ImageNet images were used. All models had AUROCs between 0.51 to 0.81 for all disease classes and ResNet-50 had the best performance for all classes except one.

Yao et al. proposed an encoder model similar to DenseNet [8] with LSTM decoders exploiting the dependencies between labels. Trained on the ChestX-ray14 dataset, the encoder-only model had higher AUROCs in 13 out of 14 pathologies than the pre-trained models used by Wang et al. [18]. Rajpurka et al. subsequently trained a 121-layer DenseNet to obtain a model called CheXNet that had higher AUROCs than those obtained by Yao et al. [19] and Wang et al. [18]. Liu et al. proposed a segmentation-based model that first trained a model (called Lung Region Generator) on the JSRT dataset to extract the lung regions of the ChestX-ray14 images [11]. Two CNNs were then applied to the entire image and the lung region, which were then combined with a fusion model to predict the pathology labels. Models using the entire image and the lung regions separately outperformed the models by Wang et al. [18] and Yao et al. [19] in all classes except one for the lung-region-only model. Neither model (entire image or lung region only) is consistently better than the other but the fusion model is consistently better than both.

CheXpert is a relatively newer and larger dataset created by Irvin et al. using chest radiographic data from Stanford Hospital [9]. The disease classifications were extracted using a labeler designed by the authors and were claimed to be more accurate than those used to build the ChestX-ray14 dataset. An additional uncertainty label was also introduced. The study trained several CNN models such as ResNet-152, DenseNet-121, Inception-v4 etc. and found DenseNet-121 to have the best performance. The best model outperforms 2 or 3 radiologists in 4 of the disease classes on a test set.

1.2 Data Exploration

The image dataset chosen for this project is the ChestX-ray14 dataset consisting of 112120 frontal view images, where each image is associated with one of fourteen disease classes or a "No Finding" class. This dataset is an augmented version of the original ChestX-ray8, which has a slightly smaller number of images and eight disease classes [18]. Each image of the dataset has a resolution of 1024×1024 pixels and a size of several hundred KB. Details can be found in the file

"Data_Entry_2017_v2020.csv." The original label (which is referred to as a string label in this document) for each image is in the form "Disease 1 | Disease 2 | ... " or "No Finding", i.e., multiple diseases can be associated with each image. To facilitate data exploration and model training/evaluation, a 15-dimensional one-hot vector including "No Finding" as a class is used to represent each image label. Other details such as "Patient Gender", "Patient Age" and "View Position" are also included. The list of training and test sets are provided by the authors in the files "train_list.txt" and "test_list.txt".

Table 1 shows the number of images for the full (original) training-validation and test sets. The corresponding image frequency per class is highly non-uniform with 60361 images without a finding and only 227 images with Hernia as shown in Fig. 1. Some basic exploration of the frequency of diseases vs. non-image features are shown in Fig. 2. One can see similar label distribution for both genders, but higher prevalence for some diseases for the AP view position. The latter makes sense since AP X-rays are typically used for patients who are unable to stand [1]. The probability density plot also shows higher mean patient age for images with disease findings.

To overcome the limitation in time and available computing resources, we sampled a subset of the training validation and testing data, which together represent about 20% of the original images. The percentage of training, validation and test images in the subset is 66%, 7% and 27%, respectively. Our sampling method is based on the targeted number of images for each disease class and that of the "No Finding" class, which for this particular subset are respectively 2000 and 10000. We first sort the pathologies from the least to most frequent and perform one of two things on each class starting from the least frequent: (1) select all images if there is insufficient number of images and (2) randomly sample the targeted number of images if there are enough images. We then intersect the resulting subset with the full training-validation and test sets to obtain the corresponding subsets. We achieve numbers that are quite close to the targets when there is a sufficient number of images for that disease class as observed in Fig. 1. We obtain the training-validation splits using `StratifiedShuffleSplit` from `scikit-learn` with the string labels. Since stratified splitting requires at least two samples per label, images with string labels that occur just once have to be rejected.

Table 1: Number of images in datasets

Dataset	No. of images	% of all images
Full training-validation	86524	77.1 %
Full test	25596	22.8 %
Training-validation subset	16944	15.1 %
Training subset	15249	13.6 %
Validation subset	1695	1.5 %
Test subset	6112	5.45 %

1.3 Problem and Objectives

The deep learning problem we are trying to solve is the classification of an X-ray image into one or more disease or no-finding classes (a multi-label binary classification problem). Let the number of classes be C and the number of images be N. We define the probability vector predicted by a model for the j-th images as $\hat{\mathbf{y}}^{(j)} = \{\hat{y}_1^{(j)}, ..., \hat{y}_C^{(j)}\}$, where $\hat{y}_i^{(j)} \in [0, 1]$, $i = 1, ..., C$. The classification of the image can be obtained by setting a probability threshold for each $\hat{y}_i^{(j)}$ above which the image is labeled with class i. Let $\mathbf{X}^{(\mathbf{j})}$ be a flattened vector of the image pixel values and $\mathbf{y}^{(j)} = \{y_1^{(j)}, ..., y_C^{(j)}\}$ be the multi-hot encoded vector of the true label of that image. Then, the deep learning problem can be formulated as

$$\text{argmin}_\theta \frac{1}{N} \sum_{j=1}^{N} L(\hat{\mathbf{y}}^{(\mathbf{j})}, \mathbf{y}^{(\mathbf{j})}) = \text{argmin}_\theta \frac{1}{N} \sum_{j=1}^{N} L(\mathbf{f}(\mathbf{X}^{(\mathbf{j})}); \theta), \mathbf{y}^{(\mathbf{j})}), \tag{1}$$

where $L(.,.)$ is the loss function, $\mathbf{f}$ represent the neural network model and θ are the weights of the model.

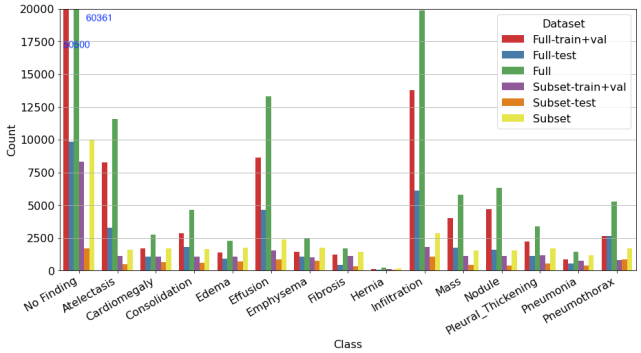


Figure 1: Number of images for each class. The number of images with "No Finding" in the full training+validation and full sets are 50500 and 60361, respectively.

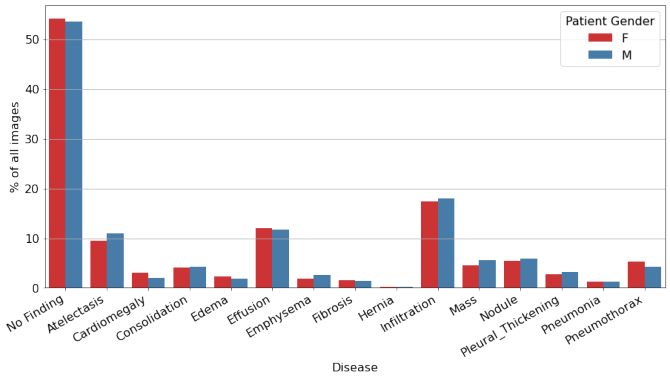
This project has two objectives. First, it aims to reproduce the relative comparison of the DenseNet-121 [8] and ResNet-50 [7] models used in the CheXNet paper [14], where it was shown that the CheXNet DenseNet-121 model has higher AUROC scores than the ResNet-50 model. The ResNet-50 model has the best performance in the study by Wang et al. [18].

The second objective is to use feature engineering and/or modified models to improve the AUROC scores of the DenseNet-121 model. To that end, we have tried the following approaches:

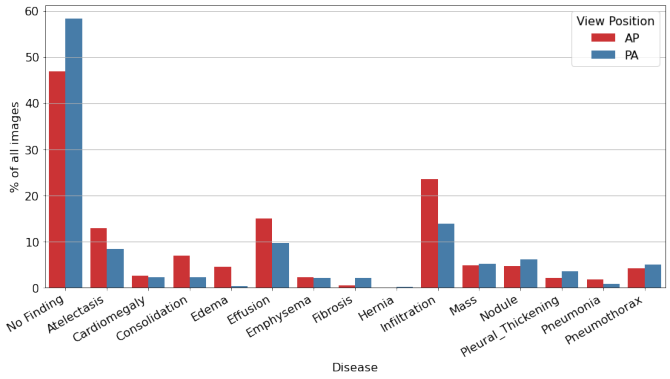
1. Adding an attention mechanism to the DenseNet-121 model in a manner similar to that proposed by Jetley et al. [10] for VGGNet [17] and ResNet and applied to CIFAR-10 and CIFAR-100 datasets.

2. Combining the DenseNet-121 model with a smaller network using the non-image features view position, patient age and patient gender similar to another study [4], but with a slightly different network for the non-image features.

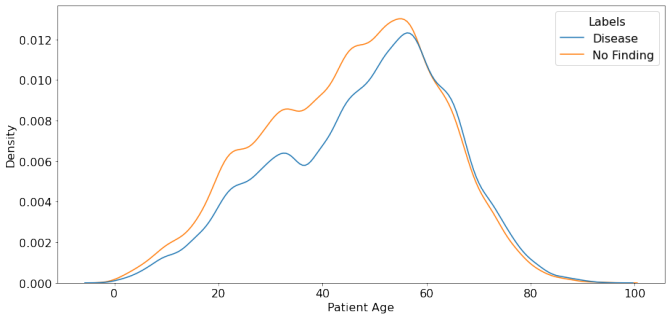
2. APPROACH AND IMPLEMENTATION



(a)



(b)



(c)

Figure 2: Percentage of images for each disease class for (a) different genders, and (b) different view positions. (c) Probability density vs. patient age for images with no finding and images with disease labels

2.1 Baseline Models

Two state-of-the-art models — ResNet-50 [7] and DenseNet-121 [8] — serve as our baselines. With the invention of VGGNet [17], it was realized that deeper networks can help improve prediction accuracy. However, deeper networks suffer from the vanishing gradient problem. ResNet contains skip connections (identity shortcut connections) to alleviate the problem. Each skip connection adds the input from the previous layer to the output of the current layer. As a result, a very deep ResNet network can be used without performance degradation. ResNet-50 is a ResNet network consisting of 50 layers.

However, like many deep networks, ResNet suffers from an extremely large number of weights. DenseNet was proposed to not only maintain the ability to alleviate the vanishing gradient problem, but also to reduce the number of weights by encouraging feature-reuse. DenseNet starts with a convolution, batch normalization, ReLU, and pooling layers followed by one or more dense blocks connected by transition layers. Batch normalization, ReLU, pooling, and fully connected layers are introduced in the final layers. Unlike traditional CNN, each layer in the dense block is connected to every other layer to allow for feature-reuse. DenseNet-121 is a 121-layer network consisting of 4 dense blocks. Table 2 demonstrates the effectiveness of feature-reuse in drastically reducing the number of weights — even with 121 layers, the number of weights for DenseNet-121 is only 30% of ResNet-50.

Table 2: Number of weights for the models of interest.

Model	No. of weights
ResNet-50	23538767
DenseNet-121	6969231
DenseNet-121-attA	7920527
DenseNet-121-attB	6971279

2.2 DenseNet-121 with Attention Mechanism

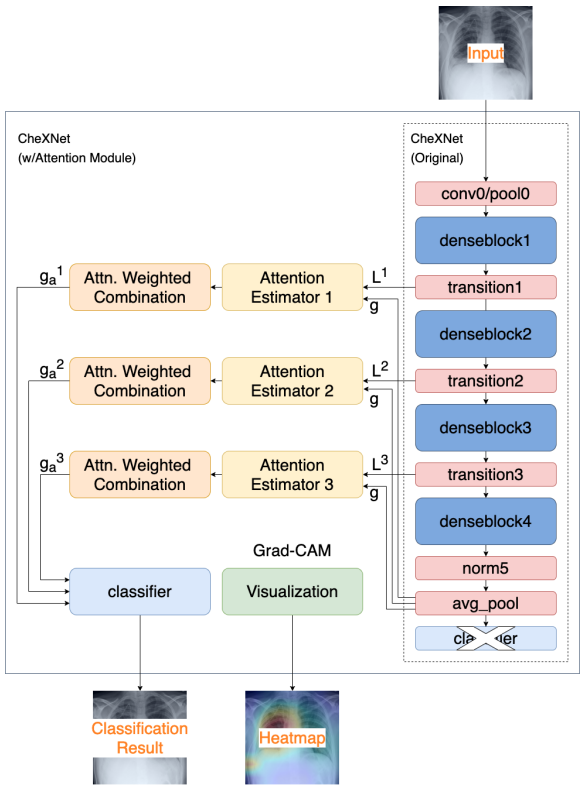


Figure 3: Proposed DenseNet-121 Model with Attention Module

We adapt the attention mechanism described in Jetley et al. [10] to the DenseNet-121 model as shown in Fig. 3. The key

idea of the method is to amplify local features using weights measuring the compatibility between local and global features. The amplified local features, instead of global features, are then used directly for classification. In the original study [10], the mechanism was applied to VGGNet and ResNet. To the best of our knowledge, we have not seen it applied to DenseNet models for X-ray image classification.

To apply the attention mechanism, we use the outputs containing local features immediately after the transition1, transition2 and transition3 layers, which are denoted respectively by $\mathbf{L}^1$, $\mathbf{L}^2$, $\mathbf{L}^3$. We also use the global feature vector $\mathbf{g}$ immediately before the classifier layer of the baseline DenseNet-121 model. Let $\mathbf{L}^s = \{\mathbf{l}_1^s, \mathbf{l}_2^s, ...\mathbf{l}_n^s\}$, where $\mathbf{l}_i^s$ is the multi-channel response at pixel location i for the s-th transition layer output, $s = 1, 2, 3$. We compute the compatibility score in two ways. In the first case, we define the compatibility score as

$$c_i^s = \langle \mathbf{u}^s, \mathbf{l}_i^s + \mathbf{g} \rangle, \quad i \in \{1...n\}, \tag{2}$$

where $\mathbf{u}$ is an unknown vector to be learned. For the DenseNet-121 network, the dimensions of $\mathbf{L}^1$, $\mathbf{L}^2$, $\mathbf{L}^3$ and $\mathbf{g}$ are 128, 256, 512 and 1024, respectively. Since $\mathbf{l}_i^s$ does not have the same dimension as $\mathbf{g}$, we first pass $\mathbf{l}_i^s$ through a convolution layer with kernel size = 1 and number of output channels equal the dimension of $\mathbf{g}$ in the same manner described in [5]. We refer to the model with compatibility score defined by Eq. 3 as **DenseNet-121-attA**.

In the second variant, we concatenate $\mathbf{l}_i^s$ and $\mathbf{g}$ and define the compatibility score as

$$c_i^s = \langle \mathbf{u}^s, \{\mathbf{l}_i^s, \mathbf{g}\} \rangle, \quad i \in \{1...n\}. \tag{3}$$

The resulting model, called **DenseNet-121-attB** in this report, has much lower number of additional weights than the first attention model as evident in Table 2. Note that concatenation was not implemented in the attention mechanism proposed by Jetley et al. but was suggested in a peer review of that paper.

Next, we normalize the compatibility score with softmax as follows:

$$a_i^s = \frac{\exp(c_i^s)}{\sum_j^n \exp(c_j^s)}, \quad i \in \{1...n\}, j \in \{1...n\} \tag{4}$$

The transition layer outputs are then weighted with the normalized compatibility score to produce the attention outputs $\mathbf{g}_a^s$, $s = 1, 2, 3$ in the following manner:

$$\mathbf{g}_a^s = \sum_{i=1}^{n} a_i^s \mathbf{l}_i^s \tag{5}$$

Finally, we concatenate the vectors $\mathbf{g}_a^1$, $\mathbf{g}_a^2$ and $\mathbf{g}_a^3$, and pass them through a classifier layer that consists of a fully connected layer followed by a sigmoid function. The main advantage of using sigmoid is to allow for direct interpretation of the classifier layer output as a probability vector when multiple labels for an image is possible. We have compared the performances of applying softmax and sigmoid on the classifier's fully connected layer output and observed that sigmoid performs slightly better.

2.3 Models with Image and Non-Image Feature Inputs

Based on previous studies such as [13], using patient history and past examinations helps radiologist interpret chest X-ray images more accurately. To generalize our models, we have combined DenseNet-121/DenseNet-121-attA with a model that receives the non-image features (patient age, patient gender and view position) accompanying the Chest X-ray14 dataset as shown in Fig. 4, similar to the study by Baltruschat et al. [4]. Instead of directly feeding the non-image features to the final fully connected layer as was done in the aforementioned study, we add a linear layer that produces a vector with 3 components followed by a ReLU layer before the classifier layer. The patient age was normalized with the mean age and standard deviation of 46.63 and 16.6, respectively. The patient gender and view position are converted to one-hot encoded inputs.

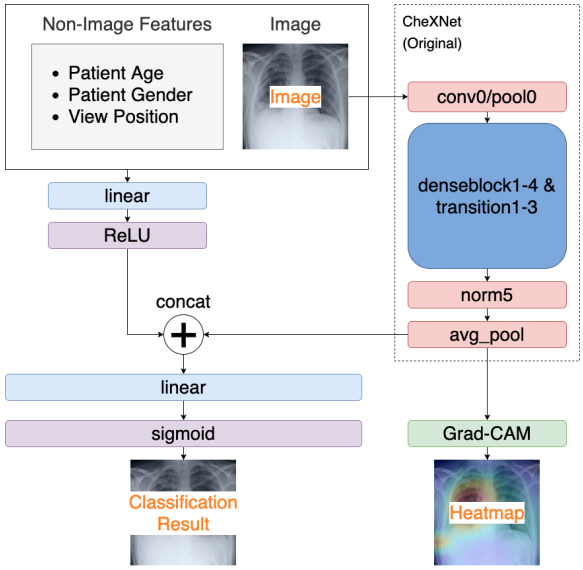


Figure 4: DenseNet-121 with non-image features

2.4 Image Processing

Unless specified otherwise, we adopt the image transformations used to train the DenseNet/ResNet models on ImageNet images [2]. The gray-scale X-ray images are first converted into RGB images, down-sampled to a size of 256x256 and center-cropped to obtain an image of size 224x224. Next, normalization is applied to the 3 image channels with the means = $[0.485, 0.456, 0.406]$ and standard deviations = $[0.229, 0.224, 0.225]$.

2.5 Heatmap Generation with Grad-CAM

We apply the Grad-CAM method developed by Selvaraju et al. [15] to generate heatmaps from DenseNet-121 and DenseNet-121-attA. First, we perform a forward operation to obtain the output after the norm5 layer, which we write as $\mathbf{A} = \{\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_n\}$, where n is the number of pixels. A backward operation follows to obtain the gradient of the classifier layer output for class c (before sigmoid) wrt $\mathbf{A}$, which is denoted by $\mathbf{G}^c = \{\mathbf{G}_1^c, \mathbf{G}_2^c, ..., \mathbf{G}_n^c\}$. We next calculate the average gradient over all pixels for each channel,

i.e.,

$$\overline{G_j^c} = \frac{1}{n}\sum_{k=1}^{n} G_{k,j}^c, \quad j = 1, ..., m, \tag{6}$$

where $G_{k,j}^c$ is j-th channel gradient at location k.
We then average $\mathbf{A}$ over all channels weighted by $\overline{G_j^c}$ and set negative values to 0:

$$h_i^c = \text{ReLU}\left[\sum_{j=1}^{m} \overline{G_j^c} A_{i,j}\right], \quad i = 1, ..., n, \tag{7}$$

where $A_{i,j}$ is j-th channel pixel value at location i.

The gray-scale heatmap ($\mathbf{H}^c = \{h_1^c, h_2^c, ..., h_n^c\}$) has a lower resolution compared with the original image. However, as shown in Fig. 5, we can upsize the heatmap to the same resolution as the input image, colorize the gray-scale heatmap and superimpose the colored heatmap onto the original image.

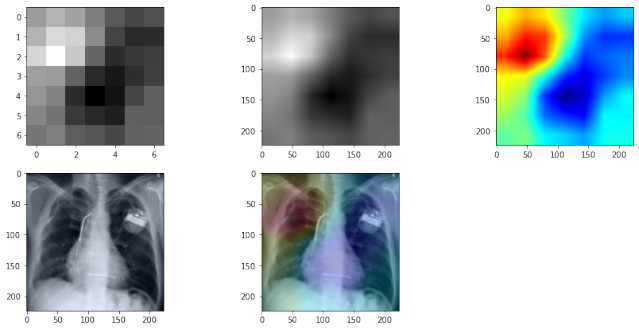


Figure 5: Process of generating heat map. Top left image corresponds to the heat map generated from Eqn. (6) and (7). The heat map is upsampled to the same resolution as the original X-ray image, i.e., 224x244, converted to a colored image and superimposed on the X-ray image to highlight the suspected diseased region.

2.6 Model Setup

Pytorch is the deep-learning library used in this project. For image processing, we apply the Pytorch Vision and Pillow Image modules. Other standard libraries in use are Numpy, Pandas, Scikit-Learn etc.

For model training, we select the binary cross entropy loss function (`BCEloss`), which for a single image is given by

$$L(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{j=1}^{C} \left[y_j \log \hat{y}_j + (1 - y_j) \log(1 - \hat{y}_j)\right], \tag{8}$$

where $\hat{y}_j$ and y_j are respectively the predicted probability and true binary label associated with class j.

We utilize the built-in Pytorch Stochastic Gradient Descent (SGD) optimizer with momentum = 0.9 and learning rates = 0.001 and 0.01. The default learning rate is 0.01 unless specified otherwise. By default, we stop training the model after 8 epochs unless stated otherwise.

We use pre-trained weights whenever possible and initialize other weights with a Xavier method that samples from a normal distribution [6]. For the baseline models, weights except classifier weights are pre-trained on ImageNet images. They can be downloaded by specifying `pretrained=True` when initializing the models. The classifier weights are initialized with the Xavier method. For the DenseNet-121 attention model, pre-trained weights can be used on layers in common with the DenseNet-121 model. Other weights, i.e., those associated with the attention mechanism and classifier, are initialized with the Xavier method and trained from scratch.

The models were trained on a single NVIDIA K80 GPU of the EC2 p2.Xlarge instance or a NVIDIA GeForce RTX 2070 Max-Q GPU on a personal workstation. For EC2, the computation time for 8 epochs was around 1.5hrs. On the personal workstation, it was 2.4 hrs.

2.7 Performance Metric

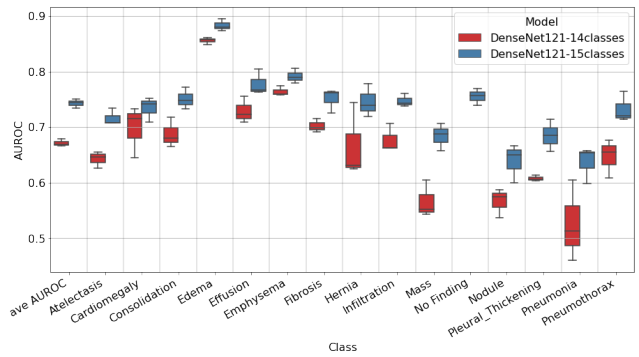
We apply the `roc_auc_score` function from the Scikit-Learn library to calculate the Area Under the Receiver Operating Characteristic Curve (AUROC) score of each pathology and the "No Finding" class. The weighted average of the class AUROC, where the weight of a class corresponds to the fractions of samples with the class label, is also reported here. We generate 3 training-validation splits, train the model on the training set and calculate the AUROCs on the validation set of each split. We also train the model on the combined training and validation set, and evaluate on the test set.

3. EXPERIMENTAL EVALUATION

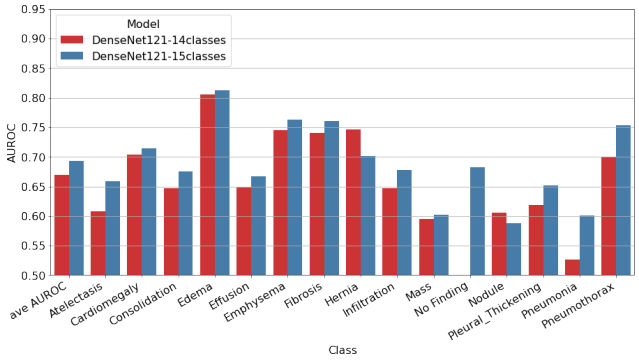
3.1 Performance

CheXNet outputs the classifier responses for all 14 pathologies but ignored the "no finding" class [14] . Other studies such as [4] include "No Finding" as an additional class. In Fig. 6, we compare the performances of the 14-class and 15-class multi-label classification and find the latter to be significantly better. Hence for the remainder of this paper, we use a 15-class multi-label classifier.

We next study the effect of learning rates on the model. CheXNet uses a single initial learning rate of 0.001 that decays by a factor of 10 when the validation loss plateaus. As shown in Fig. 7a for this study, the higher learning rate 0.01 reduces the training mean epoch loss substantially faster than 0.001. Further, ResNet-50 training loss decreases at a faster rate than the DenseNet-121 models due to the use of roughly 3 times more weights. The corresponding validation AUROCs in Fig. 7b shows that the DenseNet-121 has higher scores than ResNet-121 on average and for most classes consistent with the CheXNet study. It also demonstrates that lower loss for one model does not translate into better performance than another model with higher loss. For the baseline models, the impact of learning rate on the average AUROCs appear to be small. The AUROCs of some classes like Hernia increase while those of other classes like pleural thickening decrease. However, as shown in Fig. 7c, the gain in AUROC for the attention models with higher learning rate is significant and consistent for all classes. This is consistent with the fact that more weights of the attention models have to



(a)



(b)

Figure 6: (a) Validation AUROC and (b) test AUROC. Number of epochs = 3, learning rate = 0.001 and weights are pre-trained.

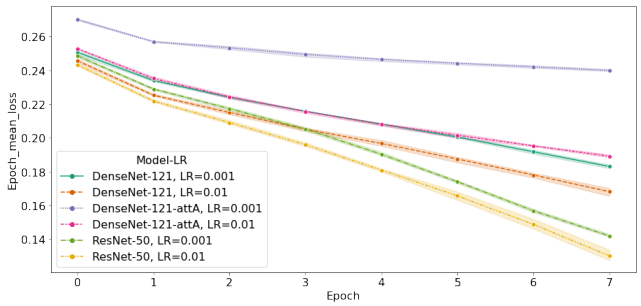
be learned from scratch.

For completeness, we compare the test AUROCs from our DenseNet-121 and ResNet-121 models with those from the other two cited studies in Fig. 8. It can be seen that CheXNet test AUROCs are significantly higher than those of our models as it was trained on a dataset roughly 5 times larger and to the point of convergence in validation score. Despite training for only 8 epochs, our model test AUROCs are rather close to that of Wang's and are higher for some of the disease classes like Cardiomegaly and Hernia.

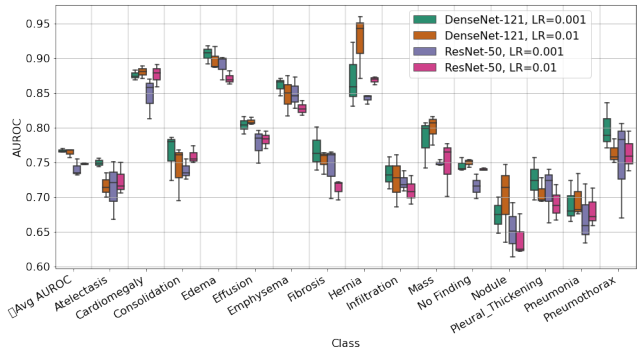
With attention mechanism, we observe that the DenseNet-121-attA and DenseNet-121-attB models have higher average scores than the baseline models as shown in Fig. 9. DenseNet-121-attA improves more over the baseline models than DenseNet-121-attB. Further, DenseNet-121-attA has higher AUROCs than other models for all classes except hernia. For that attention model, the diseases with notable AUROC improvement are fibrosis, infiltration, pneumonia and pneumothorax.

3.2 Heatmaps

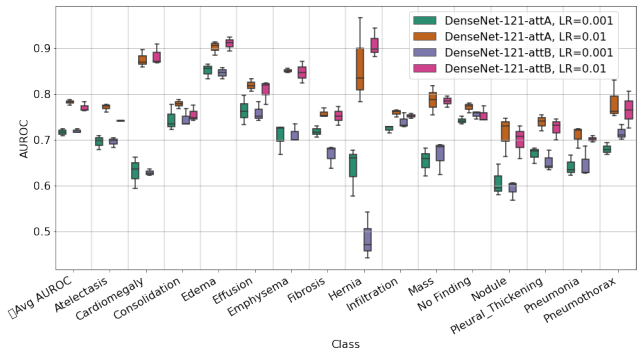
To verify that the models are correctly identifying the diseased regions, we use the Grad-CAM method described earlier to generate the heatmaps for DenseNet-121 in Fig. 10. For the image with hernia in Fig. 10a, the network highlights the lung protrusion into the vertebra region. For Fig. 10b, where the image is associated with emphysema,



(a)



(b)



(c)

Figure 7: (a) Effect of learning rate (LR=0.001 and 0.01) on the mean training loss at each epoch. Corresponding validation AUROCs for the baseline models (b) and attention models (c) with different learning rates 0.001and 0.01.

a darker-than-normal region of the lung is highlighted. The dark region indicates larger air spaces consistent with weakened and ruptured air sacs characteristic of the disease. The most-probable-class heatmaps for DenseNet-121-attA shown in Fig. 10c and d (emphysema and cardiomegaly, respectively) appear to focus on regions that are associated with those diseases.

3.3 Effect of Non-Image features

The effect of adding 3 non-image features (patient age, patient gender and view position) on model performance is shown in Fig. 11. It is observed that the non-image features do not help improve the model performance significantly on average. One reason is that those features are not strongly correlated to the disease classes as alluded to in the Data Exploration section. We expect the model to benefit

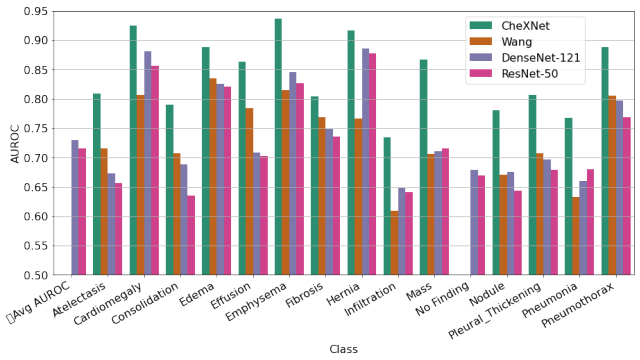
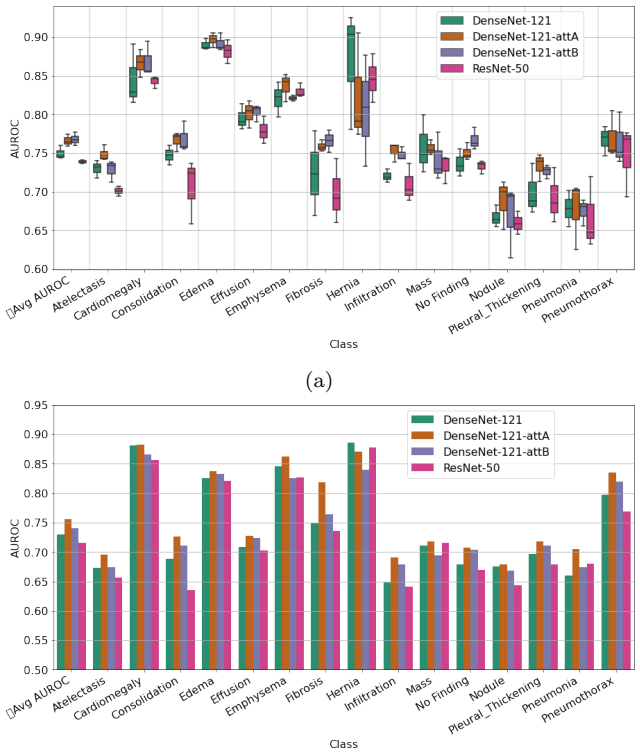


Figure 8: Test AUROCs of our baseline models with learning rate=0.01 and test AUROCs of existing studies Wang et al. [18] and CheXNet [14].



(a)

(b)

Figure 9: Comparison of baseline and attention models in terms of (a) validation AUROCs and (b) test AUROCs.

substantially with the inclusion of non-image diagnosis and history.

4. CONCLUSION

We have developed a method to choose a meaningful subset of images from the full ChestX-ray14 dataset that allowed us to train deep learning models on a single GPU within a reasonable amount of time. With this limited data set, we reproduced the superior performance of DenseNet-121 over ResNet-50 consistent with the CheXNet study and explored different ways of tuning or improving the models. We have shown that explicitly including the "No Finding" class

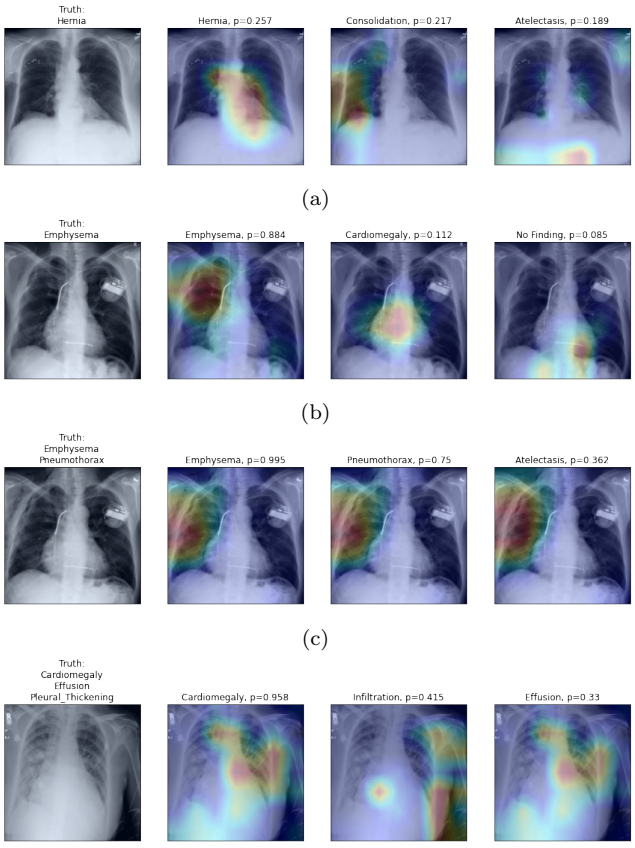


Figure 10: (a) and (b): Heat maps for the top-3 most probable classes (2nd to 4th images from left) extracted from a trained DenseNet121 model. (c) and (d): Heat maps obtained from a trained DenseNet121-att-A model. Corresponding original images are shown on the left. The predicted class and corresponding probability are shown at the top of each heat map.

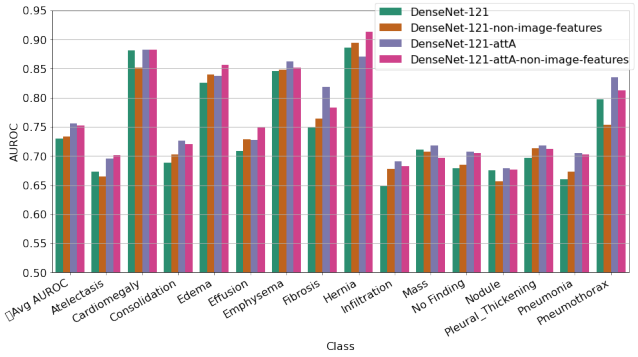


Figure 11: Performance of models with only image inputs (DenseNet-121 and DenseNet-121-attA) and models with image and non-image inputs.

produced more accurate predictions and that the attention models benefited more from the higher learning rate of 0.01 than 0.001. We have also adopted an attention mechanism first proposed for VGGNet and ResNet for the DenseNet-121

model. Two variants of the attention mechanism (DenseNet-121-attA and DenseNet-121-attB) were implemented and shown to perform significantly better than DenseNet-121. Using the non-image inputs included in the ChestX-ray14 dataset did not enhance model performance substantially. We have also applied Grad-CAM to obtain heat maps that visually explain the predictions of the models.

5. CHALLENGES AND LEARNINGS

One of the main challenges of this project was training the models on a large dataset without free access to high-performance computing resources. Our initial workflow was uploading the data into Google Drive and training the model using Google Colab's GPU. However, the latency for data transfer between Google Drive and the local drive of Google Colab greatly slowed down model training and we abandoned that workflow eventually. We overcame the problem with the EC2 p2.xlarge Instance, which allowed us to utilize GPU and high-speed data transfer for model training and testing. We also chose a subset of images for training and testing to further speed up model training. By overcoming the challenge, we have learned how to set up a cloud environment for deep learning on large datasets.

Another challenge was the modification of the DenseNet model in Pytorch to add attention modules and non-image features. Going through that process has given us the knowledge to customize and extract more information from existing Pytorch models such as freezing certain weights, initializing weights with saved models, adding and removing modules from complex models, extracting gradients etc.

Reading the literature also gave us a broader and deeper understanding of CNN models applied to image processing and the current advancements in the field. We have become more comfortable in adopting and implementing newly developed models. We also gained exposure to the exciting field of medical image analysis with neural networks.

6. CONTRIBUTIONS

M. T. contributions:

- Wrote the following scripts for training, validating and testing of the multi-label classification models: t01-multilabel-main-test.ipynb, t01-multilabel-main-val.ipynb, t01-multilabel-non_image_features-main-test.ipynb, t01-multilabel-non_image_features-main-val.ipynb
- Implemented the DenseNet-121-attA model with attention mechanism based on X. T.'s initial code. Script: densenet121attA.py
- Proposed and implemented DenseNet-121-attB and DenseNet-121 with non-image features. Scripts: densenet121attB.py, densenet_models_w_non_image.py, densenet121attA_w_non_image.py
- Devised and implemented the method for selecting a subset of images for multi-label classification. Script: p02-dataset-selection-multilabel.ipynb
- Designed the evaluation plan, wrote the code that outputs statistics of interest, collected evaluation statistics and wrote scripts to plot collected data. Script: pp01-postprocess-performance.ipynb
- Trained multi-label classification models on EC2
- Wrote all sections of the draft and report from scratch except the Background and two subsections DenseNet with Attention Mechanism and Heatmap Generation with Grad-CAM. Revised substantially and added contents to the two subsections
- Prepared the data exploration and result figures in the reports
- Improved presentation slides

X. T. contributions:

- Chest X-Ray Dataset initial analysis, implemented a selection method to generate a subset from Chest X-Ray Dataset to train binary pathology classification models. Script: p01-multi-hot-encoding.ipynb, p02-binary-dataset-selection.ipynb p03-prepare-data.ipynb
- Implemented DenseNet-121 and ResNet-18/50 based binary pathology classification models. Script: t01-densenet121-pneumonia.ipynb
- Designed and implemented Grad-CAM based heatmap visualization method based on Selvaraju et al. [15] for DenseNet-121 with attention module support. Scripts: densenet_models.py and densenet121attA.py
- Proposed and implemented Attention mechanism based on Jetley et al. [10] and integrated it to DenseNet-121. Script: densenet121attA.py
- Set up EC2 p2.xlarge environment
- Trained models in a local environment
- Wrote or partially contributed following sections in the report: DenseNet with Attention Mechanism, Heatmap Generation with Grad-CAM, Heatmaps
- Prepared the model and heatmap generation illustrations in the reports
- Improved presentation slides and created presentation video

W. C. contributions:

- Acted as the project coordinator to develop the project timeline and facilitate team meetings and communication
- Explored different training environment and experimental setup on both Google Colab and AWS EC2
- Trained models on EC2
- Created presentation slides
- Wrote or partially contributed the following sections in the report: Abstract and Introduction

J. C. contributions:

- Reviewed all work papers(proposal, draft, final) for consistency, clarity, and grammar
- Assisted with training and running initial baseline models along with basic model tuning
- Ran train, validation, and test validation on personal workstation
- Reviewed and edited presentation slides
- Wrote draft of presentation script
- Wrote or partially contributed the following sections in the report: Abstract and Introduction

7. REFERENCES

[1] Chest radiograph. https://radiopaedia.org/articles/chest-radiograph?lang=us.

[2] Pytorch densenet documentation. https://pytorch.org/hub/pytorch_vision_densenet/.

[3] X-ray (radiography) - chest. https://www.radiologyinfo.org/en/info/chestrad.

[4] I. M. Baltruschat, H. Nickisch, M. Grass, T. Knopp, and A. Saalbach. Comparison of deep learning approaches for multi-label chest X-ray classification. 2019.

[5] R. L. B. Draelos. Towards Data Science — learn to pay attention. https://towardsdatascience.com/learn-to-pay-attention-trainable-visual-attention-in-cnns-87e2869f89f1, 2019.

[6] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (PMLR)*, volume 9, pages 249–256, 2010.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. https://arxiv.org/abs/1512.03385, 2015.

[8] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. https://arxiv.org/abs/1608.06993, 2017.

[9] J. Irvin et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, 2019.

[10] S. Jetley, N. A. Lord, N. Lee, and P. H. S. Torr. Learn to pay attention. In *International Conference on Learning Representations*, 2018.

[11] H. Liu, L. Wang, Y. Nan, F. Jin, Q. Wang, and J. Pu. Sdfn: Segmentation-based deep fusion network for thoracic disease classification in chest x-ray images. *Computerized Medical Imaging and Graphics*, 2019.

[12] A. Majkowska et al. Chest radiograph interpretation with deep learning models: Assessment with radiologist-adjudicated reference standards and population-adjusted evaluation. *Radiology*, 294(2), 2019.

[13] Potchen et al. Effect of clinical history data on chest film interpretation-direction or distraction. *Investigative radiology*, 14:404, 1979.

[14] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, R. L. Ball, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng. CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning. arXiv:1711.05225, 2017.

[15] R. R. Selvaraju et al. Grad-cam: Visual explanations from deep networks via gradient-based localization, 2016.

[16] J. Shiraishi et al. Development of a digital image database for chest radiographs with and without a lung nodule receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. https://www.ajronline.org/doi/full/10.2214/ajr.174.1.1740071, 2000.

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[18] 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. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] L. Yao, E. Poblenz, D. Dagunts, B. Covington, D. Bernard, and K. Lyman. Learning to diagnose from scratch by exploiting dependencies among labels. arXiv preprint arXiv:1710.10501.