Weakly Supervised Learning for Findings Detection in Medical Images

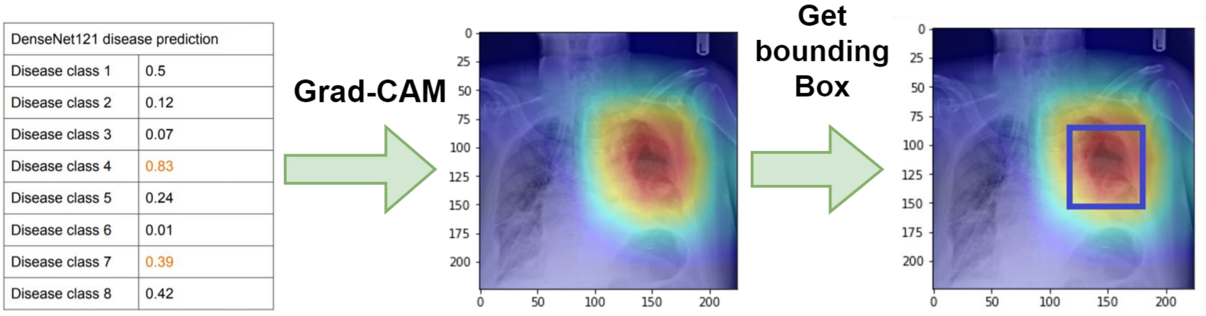
Introduction

Deep convolutional neural networks have been widely adopted in various applications. Applications on medical researches attracted attention and demonstrated remarkable progress. In this task, we applied convolutional neural networks to fulfill the requirements. Furthermore, we experimented a variety of models and mechanisms.

Data Preprocessing

We extracted data with the eight classes {Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, Nodule, Pneumonia, Pneumothorax} out from the whole dataset. We applied data augmentation on the training data, through the use of image processing techniques including horizontal flipping, shifting the image, rotating the image and center cropping. Furthermore, before sending images to the network, we divided the pixel value in the image array by 255.

Methodology



DenseNet121 disease prediction	
Disease class 1	0.5
Disease class 2	0.12
Disease class 3	0.07
Disease class 4	0.83
Disease class 5	0.24
Disease class 6	0.01
Disease class 7	0.39
Disease class 8	0.42

Figure1. Process flow of this project.

Part 1: Disease Prediction

We implemented several models for this task:

(1) **Simple CNN networks**

• Network Structure: 4 layers of CNN + 3 layers of fully connected layers. For CNN layers, the number of filters are 32, 32, 64, 128, and the kernel sizes are (2, 2), (2, 2), (4, 4), (8, 8) respectively. The activation functions are mostly ‘relu’, only ‘sigmoid’ for the last dense layer. We applied binary cross entropy loss along with adam optimizer.

Hyperparameters: batch_size=100, epochs = 20, learning rate = 1e-3, loss = binary cross entropy

(2) **DenseNet121**

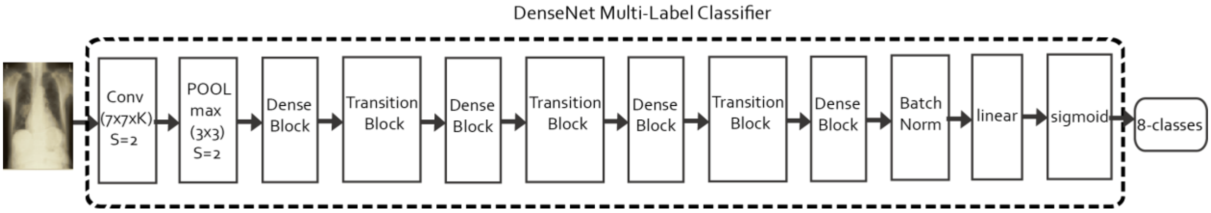


Image is modified from Ref [1].

• Training: Here we used DenseNet without pre-trained weights. The input pictures were tiled to 3 channel, resized to 224 and normalized with mean [0.485, 0.456, 0.406], standard deviation [0.229, 0.224, 0.225] on three channels. The network was trained end-to-end using Adam with standard parameters (β1 = 0.9 and β2 = 0.999) starting from 0.0002 learning rate. Loss function is binary cross entropy. After 5 epochs, the model would achieve the highest AUC score on validation set.

(3) **Resnet50**

• Training: With pre-trained ImageNet weights, we applied a two-stage training procedure. Starting with the model pre-trained on ImageNet (with include_top = False), a Global Averaging Layer, a Dense layer, and a Sigmoid activation are appended to the model. Then, with every layer but the last Dense layer locked, the model is trained with RMSProp(lr = 1e-3) for 30 epochs. Apply fine tuning based on the previous model consists of unfreezing more layers (101, 131, 153 out of 174 layers) and training with SGD(1e-4) for 50 epochs.

(4) **InceptionV3**

Model structure is described in Ref [2].

• Training: With pre-trained ImageNet weights, we applied a two-stage training procedure. Starting with the model pre-trained on ImageNet (with include_top = False), a Global Averaging Layer, a Dense layer, and a Sigmoid activation are appended to the model. Then, with every layer but the last Dense layer locked, the model is trained with RMSProp(lr = 1e-3) for 30 epochs. Apply fine tuning based on the previous model consists of unfreezing more layers (165, 197, 249 out of 311 layers) and training with SGD(1e-4) for 50 epochs.

(5) **Ensemble predictions**

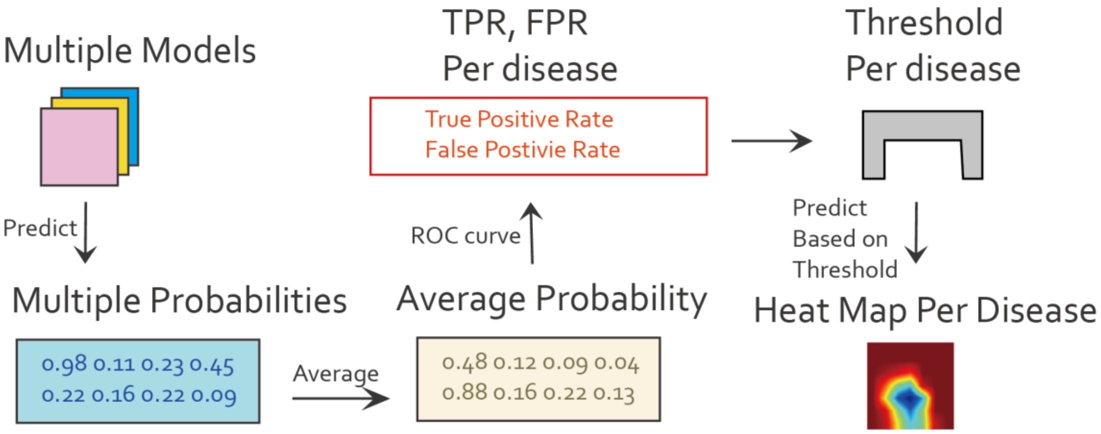


Figure 2. Ensemble Work flow

With the predictions of multiple models, one can perform averaging on the probabilities of every disease predicted by each model. Then by applying the function *roc_curve*, we can obtain the true positive rates and false positive rates of the ensembled-probabilities. By maximizing Youden's index, we can then decide an optimal threshold of each disease. The threshold is applied to the raw predictions output to indicate whether or not the disease is present. Lastly, we have our best model (DenseNet121-Pytorch) to produce the heat map of the diseases according to the thresholded predictions.

Part 2: Heat Map

We applied class-activation maps (CAM, grad-CAM) on models to assist predicting the bounding boxes.

a) CAM (Class Activation Map) (Image and paper are in Ref [3].)
Following the method proposed by Zhou et al. (Learning Deep Features for Discriminative Localization), localization can be easily achieved with the global average layer and the dense layer in the very last part of the model if they exist. By performing a linear combination on the maps before entering GAP, heat-maps are obtained and localization predictions are made according to it.

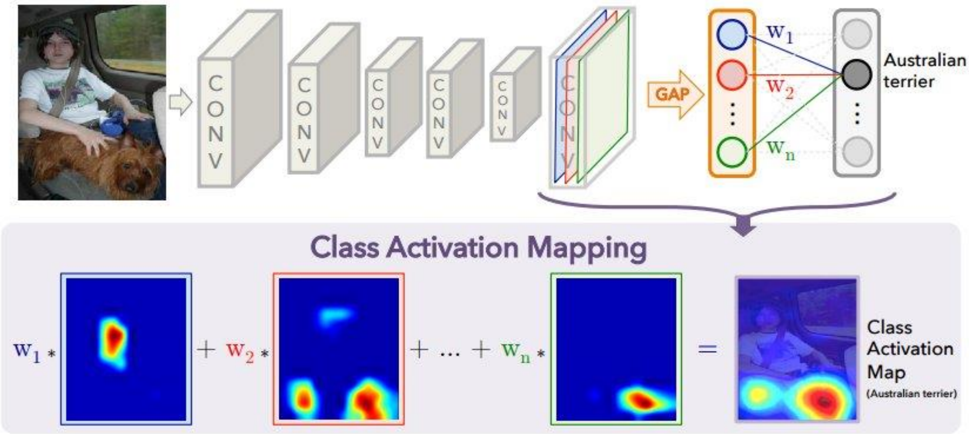


Figure 3. Class Activation Mapping

b) Grad-CAM (Gradient-weighted Class Activation Mapping) (Image and paper are in Ref [4].)
This approach uses the gradients of the target disease, flowing into the final convolutional layer to produce a coarse localization map highlighting the critical regions in the image for predicting the disease. Unlike CAM, Grad-CAM requires no re-training and is broadly applicable to any CNN-based architecture. In our application, by using grad-CAM, we don't have to modify our premodel like adding Global average pooling layer.

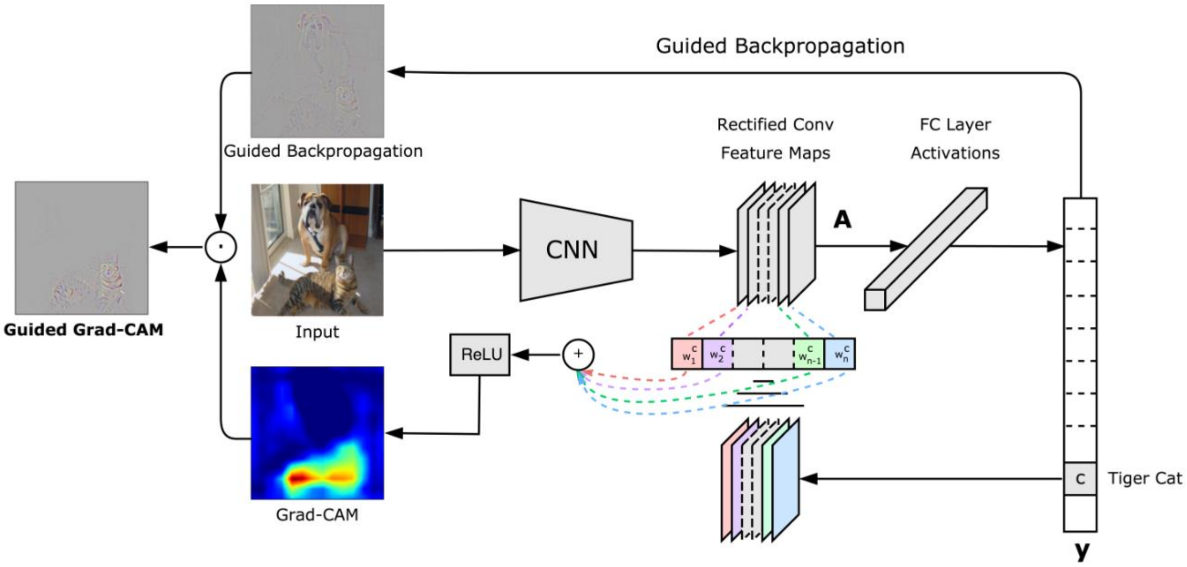


Figure 4. Gradient-weighted Class Activation Mapping

Part 3: Bounding Box

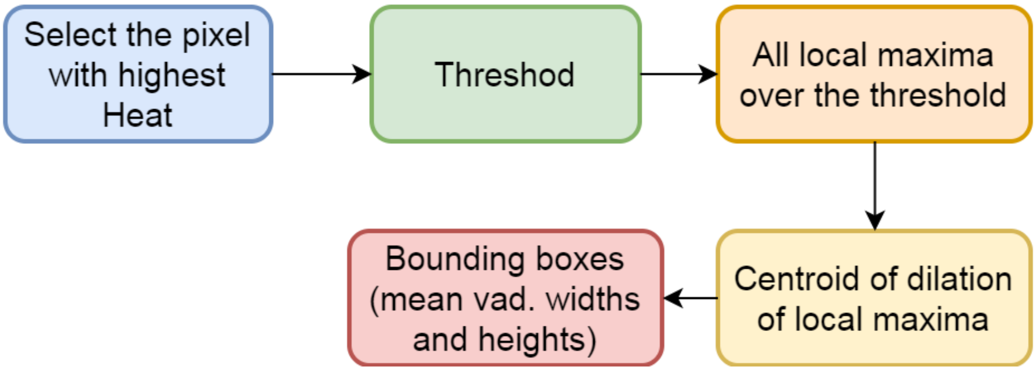

Select the pixel with highest Heat
Threshod
All local maxima over the threshold
Centroid of dilation of local maxima
Bounding boxes (mean vad. widths and heights)

Figure 5. Bounding Box decision work flow

After calculating the heat map, we select global peak value, scaled by a factor of 0.9, as the threshold to select local maximum points with intensity high enough. We apply a maximum filter and a minimum filter to each heat map, and calculate the difference between resulting heat maps to get probable regions with local maximal centroids. Then, for all the local maxima greater than the threshold in the image, we applied dilation on them to accumulate multiple candidate points, and choose the centroid of the accumulated components as the center of a predicted bounding box.

We utilized fixed size bounding box for each disease, since we have observed that same disease tends to have similar box size. For each image with certain prediction and its corresponding heat map, we first construct a box covering every local maximal centroid whose boundaries are not lower than a given threshold. The box size is then calculated as the average of all these boxes, for each distinct disease. Thresholds for each class is initially set as 0.9, and decreases by 0.05 on every experiment to get the threshold producing boxes with a size resembling ones in the validation set the most. And then we applied the mean width and height to each predicted centroid for each distinct disease.

Experiment

1) Data augmentation techniques
   a) ResNet50: controlling the percentage of image shifting

Image Shifting Percentage	Mean AUROC score
0.05 (5% of width and height)	**0.809581**
0.15 (15% of width and height)	0.796238

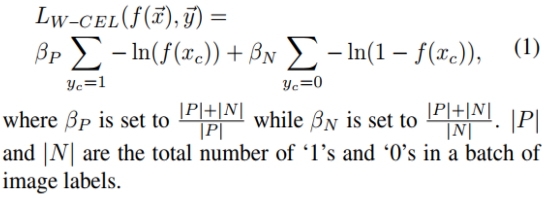
2) Using pre-trained weights (some layers frozen) / train from scratch
   a) ResNet50 : pre-trained weights from ImageNet, total layers = 174
   fine tune top N residual blocks -> mean AUROC score

N = 2 (layers[153:])	0.733198
N = 4 (layers[131:])	0.758562
N = 7 (layers[101:])	0.763800
N = 11 (layers[59:])	**0.767270**

   b) InceptionV3 : pretrained weights from ImageNet, total layers = 311
   fine tune top N inception blocks -> mean roc_auc score

N = 2 (layers[249:])	0.743673
N = 4 (layers[197:])	0.752644
N = 5 (layers[165:])	**0.768054** (18 epochs)

   c) DenseNet121 :
   Both of the following models use weight balance on loss function described in Ref [5], without data augmentation.



$$L_{W-CEL}(f(\vec{x}),\vec{y}) = \beta_P \sum_{y_c=1} -\ln(f(x_c)) + \beta_N \sum_{y_c=0} -\ln(1-f(x_c)), \qquad (1)$$

where β_P is set to $\frac{|P|+|N|}{|P|}$ while β_N is set to $\frac{|P|+|N|}{|N|}$. $|P|$ and $|N|$ are the total number of '1's and '0's in a batch of image labels.

With ImageNet pre-trained weight	**0.8126**
Without pre-train weight	0.7794

3) loss (weight or not)
   |P| and |N| denote the number of positive cases and negative cases of each disease in the training set respectively.

	positive balancing factor	negative balancing factor

Wang et al.	(|P|+|N|)/|P|	(|P|+|N|)/|N|
Pranav Rajpurkar et al.	|N|/(|P|+|N|)	|P|/(|P|+|N|)

The above table shows two different ways for weight on the loss function. We therefore did some experiments on those settings. **Here we augmented the data to four times of the original image counts with random cropping(224) and flipping.**

	DenseNet121 AUC
weight method of Wang et al.	0.8232
weight method of Pranav Rajpurkar et al.	**0.8239**
without weight	0.8195

From the above experiments, we found out that the model was still trainable even if we did not assign weight on loss. The best score was reached by using weights proposed in ChexNet paper.

Results

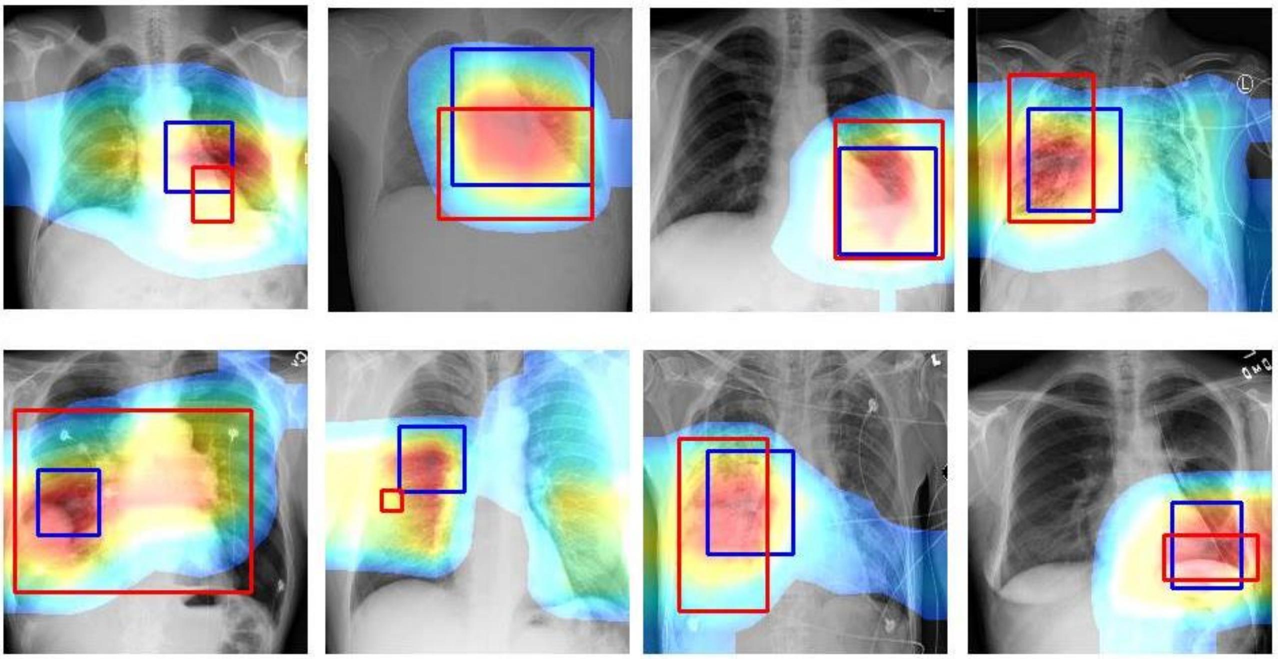


Figure 6. Visualization of some heat maps with its ground-truth label (red) and its prediction (blue) selected from each disease class. (From top-left to bottom: Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, Nodule, Pneumonia and Pneumothorax)

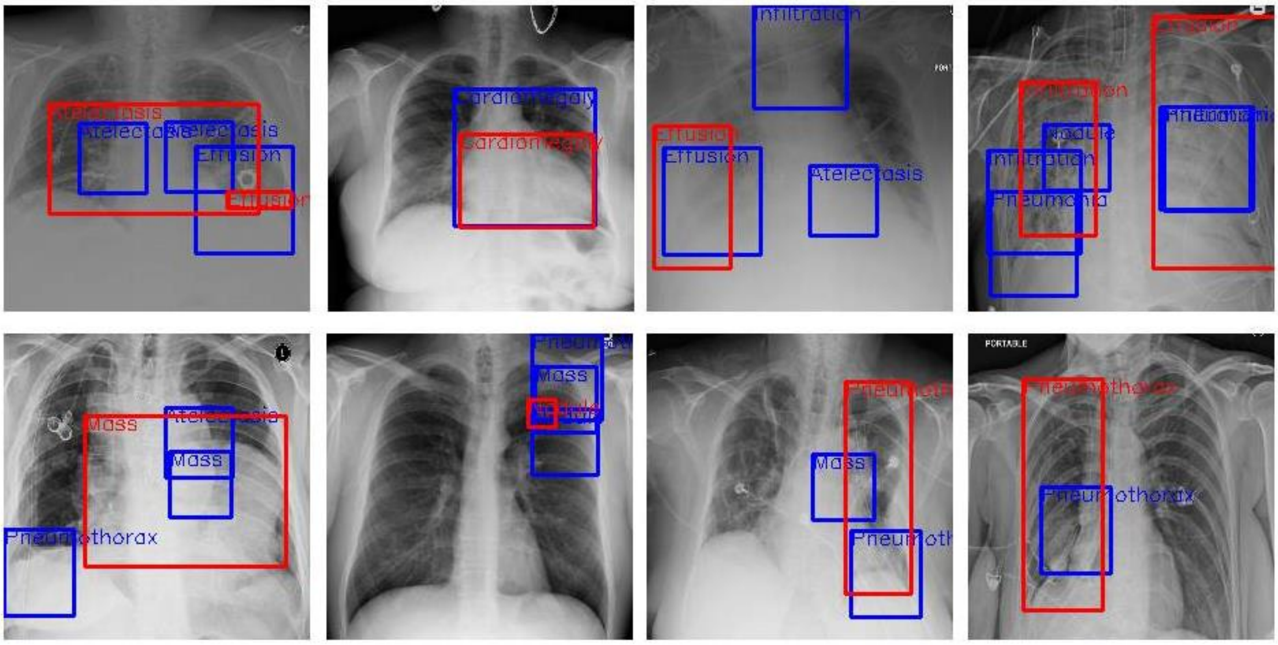


Figure 7. Visualization of some images with its ground-truth label (red) and its prediction (blue) selected from each disease class.

	AUROC
Atelectasis	0.8486
Cardiomegaly	0.9479
Effusion	0.8884
Infiltration	0.6630
Mass	0.8147
Nodule	0.8066
Pneumonia	0.7228
Pneumothorax	0.8996
Total average	**0.824**
Testing IOU	**0.25303**

Table 1. Final Disease Prediction AUROC

Local Maxima	0.21780
Local Maxima + Average Box[1]	0.25075
Local Maxima + Average Box[2]	**0.25303**

Table 2. Result of different Bonding Box Algorithms

[1] Add an average box for each prediction. The average coordinate is calculated by the same method for the average box size, described in previous section

[2] Revised version to reduce predictions with relatively lower likelihoods

Conclusion

We completed the task by predicting the disease using a multi-label classifier. The best model, DenseNet121 was used to produce heat map, utilizing the gradients of the last convolution layer. After receiving the heat map, we transformed it to boxes according to local maxima detection and thresholding.

Reference

[1] Learning to diagnose from scratch by exploiting dependencies among labels :
https://arxiv.org/pdf/1710.10501.pdf
[2] Inception-v3 Model : http://josephpcohen.com/w/wp-content/uploads/inception-v3.pdf
[3] Learning Deep Features for Discriminative Localization:
https://arxiv.org/pdf/1512.04150.pdf
[4] Grad-CAM: Visual Explanations fro
m Deep Networks via Gradient-based Localization: https://arxiv.org/pdf/1610.02391.pdf
[5] ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases:
https://arxiv.org/pdf/1705.02315.pdf