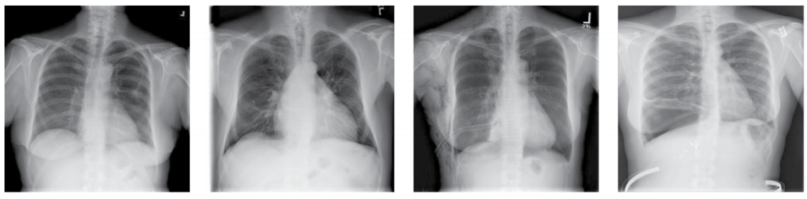
Classification and Localization of Disease with Bounding Boxes from Chest X-Ray Images

Hugo Kitano
Department of Computer Science
Stanford University
hkitano@stanford.edu

1 Introduction

The use of deep learning has become increasingly popular in current biomedical science circles with the recent surge in availability of many types of medical data. One of the most popular machine learning fields within healthcare is computer vision, an area that has achieved success across a variety of datasets and usages. Chest x-rays are the most common type of radiology exam in the world[1], but diagnosing one of the possible chest afflictions to the many organs and systems in the chest is a difficult task. The NIH Chest X-ray Dataset[2], released in 2017, is one of the largest publicly available X-ray datasets, and has spawned a number of models to predict disease from the x-rays. Most of these models, including the famous CheXNet[3], are implemented to predict the binary classification of the presence of each disease.



(a) "No Finding" **(b)** "Cardiomegaly" **(c)** "Pneumothorax" **(d)** "Pneumothorax"

Figure 1: Four examples from the dataset, from "Comparison of Deep Learning Approaches for Multi-Label Chest X-Ray Classification"

However, this project is focused not ultimately on the classification, but also on the localization of the disease. Approximately 1% of images in the dataset have their disease localization described by bounding boxes. The approach of this project was to train a multi-class classification model on the training and validation set, optimizing for the mean AUC over all classes. Then, I obtained the images' class activation maps using Grad-CAM++, an extension of Grad-CAM, and converted the maps to bounding boxes. Lastly, I evaluated the results with intersection-over-union and other metrics. Overall, I believe the model is useful as a diagnostic tool for doctors and other medical professionals who could diagnose and locate diseases of the chest from a single chest x-ray.

2 Data

The NIH Chest X-ray Dataset is comprised of 112,120 X-ray images with disease labels from 30,805 unique patients; each of the images has a label describing one of 14 diseases (atelectasis, consolidation, infiltrate, pneumothorax, edema, emphysema, fibrosis, effusion, pneumonia, pleural thickening, cardiomegaly, nodule, mass, and hernia), or the absence of disease. Multiple diseases can be present in one x-ray. Of the images with listed bounding boxes, only eight of the diseases

(atelectasis, cardiomegaly, effusion, infiltrate, mass, nodule, pneumonia, and pneumothorax) are represented.

One large problem is that the dataset is quite imbalanced, with most images containing no disease and far less than a thousand examples for the least frequent classes. Because the images that have bounding boxes in the dataset represent only eight classes, this project only deals with these eight classes (atelectasis, cardiomegaly, effusion, infiltrate, mass, nodule, pneumonia, and pneumothorax). These eight categories are actually the original eight represented in the dataset, before examples featuring the last six categories were added to the dataset later. This also means that a weighted loss by class is likely optimal to try to mitigate the effects of a class imbalance.

The chief caveat of the dataset is that the class labels are not recorded by a medical professional; rather, they are gleaned using NLP techniques from their corresponding radiology reports, which are not released due to privacy concerns. Thus, the dataset creators believe their labels to be around 90% accurate.

However, the labels might be more problematic than that. According to Luke Oakden-Rayner, a radiologist, the labels are highly inconsistent[4], with more inaccuracy than the dataset authors hypothesize. Furthermore, there are structured inconsistencies throughout the dataset that produce structured noise, which means that obtaining good performance on the test sight might be not clinically useful at all. However, Oakden-Rayner praises the CheX-Net model[5], stating it probably does predict at expert-level, despite the data's flaws. Thus, clinically useful results are possible despite the frequent mislabeling of data.

At least the bounding box data seems legitimate. The 983 images with bounding boxes were labeled by an expert; although these boxes could be inaccurate, they're much more likely to be correct than the NLP-dependent labeling. All of the bounding-box images are part of the test set. Each image is labeled with a disease and a (x, y, w, h) description, where x and y denote the upper-left corner of the box, and w and h denote the width and height of the box.

3 Approach

The approach can be divided into four steps. The first step is to pre-process the data. Second, a classifier was trained on the eight classes using the training and validation dataset. Next, Gradient-weighted Class Activation Mapping ++ (Grad-CAM++)[6] was used to obtain activation maps on the test images with bounding boxes. Lastly, the best bounding boxes given the class activation maps were found, and their success was evaluated by comparing them to the ground truth.

The first step is data pre-processing. I borrowed heavily from T.H. Tang's public code on processing the data, though I additionally processed test data as well. As mentioned earlier, only the eight categories represented among the bounding-box images were used. The original dataset had a train-validation and test split, so I split the train and validation set via a 90-10 ratio. There is no patient overlap between the sets. I resized every image from a 1024 by 1024 image to a 256 by 256 image for efficiency, and stored them in numpy arrays to be reloaded later. This step is crucial to reduce run-time while training.

The second step was training a classifier. This is also based on T.H. Tang's code, but with some major changes. I first handled data augmentation, a common technique for increasing the training set size; a random crop and random horizontal flip were implemented on every training image. With these augmentations, I decided to increase the training set size by a factor of four. For the model, I used transfer learning like many previous projects in the domain, using ResNet101 and DenseNet121 pre-trained on ImageNet. The only change to these pre-trained models is an added dense layer that outputs eight logits, one for each class, and a final sigmoid function. Note that I use a sigmoid function rather than a softmax, since multiple classes can be represented by one image. I also compared a vanilla loss function with CheXNet's weighted loss function[3], which alters the loss depending on the prevalence of the class.

The training used Adam optimizer over binary cross-entropy loss for eight epochs, with a learning rate of 0.0005. Every epoch, I would calculate the AUC scores for each class between the ground truth and the predicted class of the validation set images. As described in the Experiments section, the weighted DenseNet121 performed the best, obtaining the best mean AUC over all classes for the validation set.

I made crucial improvements on T.H. Tang's code that make gigantic memory savings and speeds up training. Tang's code requires 500GB of RAM, which is simply way too expensive and slow. This problem was circumvented by saving and loading intermediate memory-intensive arrays and reworking the data augmentation steps into an infinitely looping dataloader. These optimizations cut the time it took to complete the data-loading task in more than half, and ensured the training only uses about 15 GB of memory maximum, which is more efficient by a factor of more than ten.

For the third step, I experimented with both Grad-CAM and Grad-CAM++ as methods for obtaining the activation map. Though at this step I used both, by the final step, I realized Grad-CAM++ was clearly superior. Class activation maps are always taken with respect to a class, but selecting only the class with highest probability for a given image would limit the total information provided, and is also dangerous with an unbalanced dataset with severe questions about its accuracy. For this project, every image that has a bounding box is run through the trained model, and if the class probability is greater than the class's maximum Youden's index on the validation set, the class activation map is produced and saved for the last step. Youden's index optimizes for sensitivity and specificity, so any probability higher than the Youden's index should be a likely class. This way, there can be multiple activation maps for the bounding box step, even if some of them are not corresponding to the same class as the image. Both the Grad-CAM and Grad-CAM++ activation maps were saved. Adapting the Grad-CAM++ implementation for this project was done entirely by me.

Lastly, I needed to derive one bounding box per image, as per the bounding box data in the dataset. I realized that since the goal of the project is to assist medical professionals with reading x-rays, larger bounding boxes should be more acceptable than small ones: as long as the bounding box contains the disease, doctors should be able to find the disease within it. This would lead to lower intersection-over-union scores, but in practice just as useful. To obtain the bounding box, I used a threshold of the mean of the activations multiplied by a hyperparameter t. The class activation map's pixels that are above the threshold are divided into various rectangles using the scipy package. Lastly, the largest connected rectangle over all the class activation maps assigned to an image is found and converted to bounding box coordinates. I implemented this all myself, while adding extra methods to visualize the class activation map, the class activation map over the original image, the bounding boxes over the original image, and more visualization techniques.

I based the first two steps of my code on T.H. Tang's repository on Github[7], which implements the CheX-Net model. I used WonKwang Lee's repository on Github[8] for his implementation of Grad-CAM++.

4 Experiments

The immediate results of training were promising. I found that DenseNet121 did marginally better than ResNet101, and was much faster as well. The best ResNet model had an average AUROC across eight classes in the validation set of 0.766, and each individual AUROC was above 0.60. Over the test set, the average AUROC across the eight classes was 0.739, and all individual AUROC's were over 0.57. However, the best DenseNet model had an average AUROC of 0.779 on validation, and 0.758 on test. When limiting the test set to the images with bounding-boxes, the average AUROC is 0.784, which is quite high. I expect the images given bounding boxes are intentionally easier to classify and localize.

One problem seen during training is with each epoch, the binary cross-entropy loss would decrease, but the AUC would sometimes increase, thus making the best-performing model occur within the first few epochs. My hypothesis was that the unbalanced nature of the dataset causes this standard loss function to not do as well. So, I used the weighting function for the loss from the CheX-Net paper[3], and found some marginal improvement. Thus, the best model was weighted-loss version of DenseNet121.

The next step was to get class activation maps with Grad-CAM++. Figure 2 shows five activation maps: *a* is the only activation map for image 250, an instance of cardiomegaly, a class our model is good at recognizing. The class activation map does in fact correspond with cardiomegaly. *B* and *c* correspond to image 607, which is an instance of mass, with *b* corresponding to the nodule class activation map and *c* with the mass class activation map. *D* and *e* correspond to image 932, which is an instance of pneumothorax, with *d* corresponding to the atelectasis class activation map and *e* with the pneumonia class activation map.

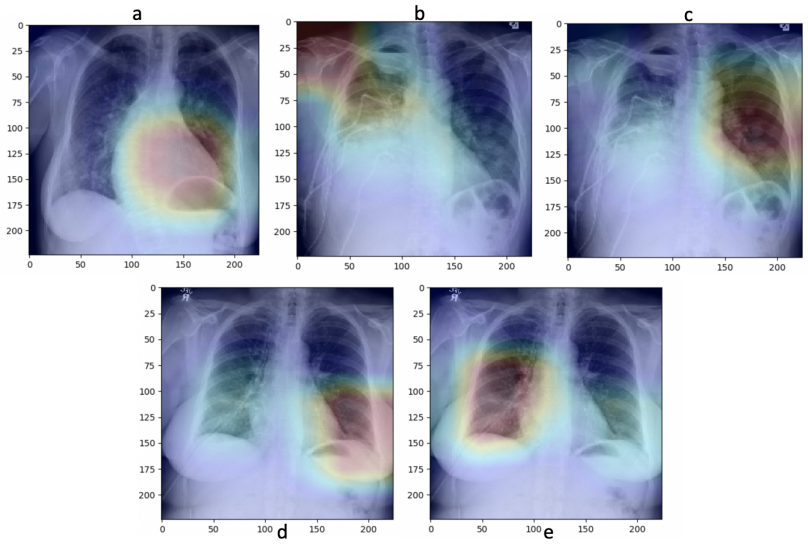


Figure 2: Class activation maps overlaid over original image. *a* is of an instance of cardiomegaly, *b* and *c* are of an instance of mass, and *d* and *e* are of an instance of pneumothorax

Then, I found the largest bounding box among all the class activation maps associated with the image above a certain threshold t. The three metrics of success used are the following:

- intersection-over-union (IoU), the most common bounding-box metric, measures the intersection of two bounding boxes and divides that by the union of the two bounding boxes.
- containment measures whether one of the bounding boxes completely contains the other. Since the predicted bounding boxes will likely be large, this is an important metric.
- non-overlap describes when neither boxes overlap with each other (and have an IoU of zero). These are clear failures.

Varying t had great effects on these three metrics. Simply, as t increases, the size of the bounding boxes decreased, so the average IoU increases, but containment decreases and non-overlap increases. Because I would rather use a larger bounding-box, I chose a higher than average threshold, and set it at 1.95.

The final Grad-CAM++ results state an average IoU of 0.201, with a 19.3% non-overlap rate and a 35.4% containment rate. It clearly outperforms a Grad-CAM implementation, which has an average IoU of 0.186, a 21.4% non-overlap rate and a 32.8% containment rate.

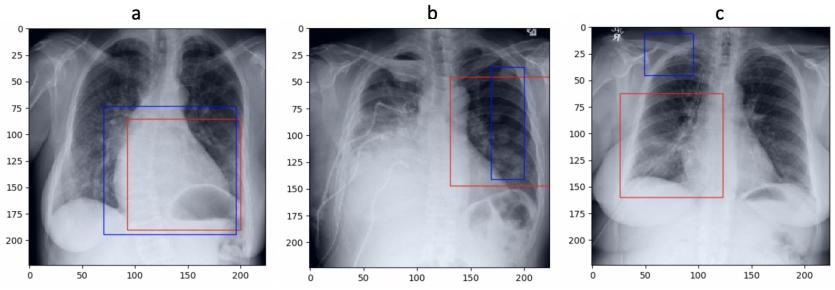


Figure 3: Ground truth (blue) and prediction (red) bounding boxes overlaid over original image. *a* is of an instance of cardiomegaly, *b* is of an instance of mass, and *c* is of an instance of pneumothorax

Looking at the heatmaps shown in Figure 2, *a*, the instance of cardiomegaly, has bounding boxes predicted almost perfectly correctly by the model in Figure 3. There was only one class activation

map to choose from, and it was the one corresponding to the cardiomegaly class. For *b* and *c* in Figure 2, which correspond to an instance of mass, the model chose the bigger heatmap, which ended up corresponding to the correct class as well. Even though the predicted bounding box is acceptable, the strange shape of the ground truth makes it difficult to predict well. For *d* and *e*, an instance of pneumothorax, neither of the class activation maps correspond to the correct class (they correspond to atelectasis and pneumonia, respectively), so the prediction is very wrong. This would be a case of non-overlap.

The success of the model is very dependent on the class of the image. Figure 4 shows the total number of images in the bounding-box test set per class, as well as the IoU, non-overlap, and containment values.

	Atelectasis	Cardiomegaly	Effusion	Infiltrate	Mass	Nodule	Pneumonia	Pneumothorax
Test Images	180	146	153	123	85	79	120	98
IoU	0.102	0.534	0.174	0.227	0.124	0.013	0.232	0.083
Non-overlap	0.194	0.007	0.222	0.114	0.165	0.443	0.100	0.459
Containment	0.538	0.055	0.261	0.398	0.612	0.443	0.375	0.224

Figure 4: Number of images and average IoU, non-overlap, and containment values per class

There are many interesting phenomena to note here. There seems to be no relationship between IoU, non-overlap, and containment between classes; each class can be quite unique in its properties. The model is strong at classifying and localizing cardiomegaly: it's the class with the highest AUC for classification, and it completely misses only one of 146 cases, with a strong IoU. This is likely because cardiomegaly is a case of a large heart, which is easy to locate and is likely to be fit with a large bounding box. Regardless, these results are quite amazing.

However, the class with the worst classification AUC, infiltrate, does a little above average here. This means that success with bounding boxes is dependent not only on model's strength, but the size, shape, and location of the bounding box. Luckily enough, pulmonary infiltrate is a simple buildup in the lungs, so bounding boxes are quite large, and always central to the lungs.

Meanwhile, the two classes that perform the worst, nodule and pneumothorax, have extremely small bounding boxes with great variety of placements in the image. Nodules are usually less than three centimeters in width, so their ground-truth bounding boxes are tiny in comparison with the other classes. They also can appear anywhere in the lungs. Pneumothorax is a collapsed lung, so their bounding boxes appear anywhere on the circumference of the lung, which is a wide range. Its bounding boxes can also be very small.

The variety in size, shape, and location of the bounding boxes makes localization difficult. My bounding box algorithm currently suggests larger bounding boxes that are more central (boxes towards the edges simply have a smaller chance of being large). This works well for some classes, but not others.

5 Conclusion

For this project, I was able to train a model for both classification and localization, dealing with an unbalanced dataset, clear data inaccuracies, and no bounding boxes to train with. The model classified decently with a strong AUC, and was able to localize many of the classes, such as cardiomegaly and infiltrate, well. For most classes, my choice of larger bounding box seems to work well to indicate the region of disease.

Some adjustments to improve the model would be redoing it with a different train-validation split: because of the data imbalance, different splits can affect the model greatly. We also could have trained with a smaller training rate (or with learning rate decay) to try to pinpoint the model with the best validation, since the best model usually appeared in the first couple epochs.

For stronger classification, a more accurate dataset is a must; as it is currently, there's only so much that can be learned when we do not trust the labels. If our model was very accurate with respect to classification, then we could use prior knowledge about the classes to improve our bounding boxes; for example, we'd know if we are using a nodule class activation map to use a very small bounding box. This would fix the problem of weirdly-shaped bounding boxes, and would be a logical next step.

6 Contributions

Hugo Kitano is responsible for the entire paper and code represented here. His code can be found at `https://github.com/hugokitano/cs230_cs271project`. This project is shared between CS230 and CS271; both project reports can be found there.

Notes

[1]Yao, Li and Poblenz, Eric and Dagunts, Dmitry and Covington, Ben and Bernard, Devon and Lyman, Kevin. (2017). Learning to diagnose from scratch by exploiting dependencies among labels.

[2]Wang, Xiaosong and Peng, Yifan and Lu, Le and Lu, Zhiyong and Bagheri, Mohammadhadi and Summers, Ronald. (2017). ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. arXiv:1705.02315.

[3]Rajpurkar, Pranav and Irvin, Jeremy and Zhu, Kaylie and Yang, Brandon and Mehta, Hershel and Duan, Tony and Ding, Daisy and Bagul, Aarti and Langlotz, Curtis and Shpanskaya, Katie and Lungren, Matthew and Ng, Andrew. (2017). CheX-Net: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning.

[4]Oakden-Rayner, Luke. "Exploring the ChestXray14 Dataset: Problems." Luke Oakden-Rayner, 18 Dec. 2017, lukeoakdenrayner.wordpress.com/2017/12/18/the-chestxray14-dataset-problems/.

[5]Oakden-Rayner, Luke. "CheXNet: an in-Depth Review." Luke Oakden-Rayner, 24 Jan. 2018, lukeoakdenrayner.wordpress.com/2018/01/24/chexnet-an-in-depth-review/.

[6]Chattopadhyay, Aditya and Sarkar, Anirban and Howlader, Prantik and Balasubramanian, Vineeth. (2017). Grad-CAM++: Generalized Gradient-based Visual Explanations for Deep Convolutional Networks.

[7]Tang, T.H. "Weakly Supervised Learning for Findings Detection in Medical Images." GitHub, 7 Aug. 2019, github.com/thtang/CheXNet-with-localization.

[8]Lee, WonKwang. A Simple Pytorch implementation of Grad-CAM, and Grad-CAM++. GitHub, 3 Aug. 2018, https://github.com/1Konny/gradcam$_plus_plus - pytorch$.