

# Chest X-ray Abnormalities Detection

**Heyang Huang**

Department of Computer Science  
Stanford University  
heyangh@stanford.edu

**Yijun Long**

Department of Statistics  
Stanford University  
yjlong33@stanford.edu

**Yi Wei**

Department of Statistics  
Stanford University  
yvettew@stanford.edu

## Abstract

X-rays can be extremely helpful in diagnosing and treating abnormalities in chest images, and an automated AI powered detection tool can help reduce misdiagnosis and lower the pressure on doctors. In this project, we aimed to apply deep learning techniques to the task of detecting and localizing abnormalities in chest X-rays. We incorporated data augmentation, oversampling, and hyperparameter optimization to a YOLO model, yielding improved model performance on the test set significantly. Bounding boxes and the corresponding diagnosis labels for each image in the test dataset are predicted. Our model is able to achieve a 0.34 mAP@0.5 in the test set, which is very competitive in the current leaderboard of Kaggle competition.

## 1. Introduction

Radiologists help save the day when you have a broken arm. They diagnose and treat medical conditions using imaging techniques like CT and PET scans, MRIs, and, of course, X-rays. However, radiologists face many daily challenges, perhaps the most difficult being the chest radiograph. The interpretation of chest X-rays can lead to medical misdiagnosis, even for the best practicing doctor. A method of localization and classification of abnormalities would be highly valuable, and deep learning approaches offer some way to achieve this goal. Using a dataset of 18,000 scans along with the corresponding label annotated by experienced radiologists, we train a neural network based on YOLO (you only look once) framework. The trained network is evaluated on a test set of 3,000 images. With this method, an accurate identification along with precise localization can be generated for a certain chest X-ray image, thus relieving the stress of busy doctors while also providing patients with a more accurate diagnosis.

### 1.1 Research Background

A detailed background of the dataset has been published by Ha Q. Nguyen[2]. It is proven that using high-performance deep learning algorithms to make diagnosis on chest radiographs has achieved great success[3][4]. An earlier study by Tang, Y.-X. et al. applied deep convolutional neural networks to classify abnormality of chest radiographs[5]. Majkowska, A. et al. used the Natural Language Processing model to make predictions on the unlabeled ChestX-ray data[6].

## 2. Data Description

We have worked on the dataset of 18,000 independently-labeled postero-anterior (PA) CXR scan (DICOM) images provided by two hospitals in Vietnam: the Hospital 108 and the Hanoi Medical University Hospital for the Kaggle competition -VinBigData Chest X-ray Abnormalities Detection[1]. The data can be found online at <https://www.kaggle.com/c/vinbigdata-chest-xray-abnormalities-detection/data>.

All images were labeled by a panel of experienced radiologists for the presence of 14 critical radiographic findings, which are aortic enlargement(0), Atelectasis(1), Calcification(2), Cardiomegaly(3), Consolidation(4), ILD(5), Infiltration(6), Lung Opacity(7), Nodule/Mass(8), Other lesion(9), Pleural effusion(10), Pleural thickening(11), Pneumothorax(12), and Pulmonary fibrosis(13). Each number in the parentheses represents the corresponding label number. Note that the "No finding" observation (14) captured the absence of all 14 findings above.

In our train datasets, one row for each object includes a class and a bounding box. Note that some images have multiple bounding boxes. For each test image, we will be predicting bounding boxes and classes for all findings. If no findings are predicted, a

prediction of "14 1 0 0 1" should be created (14 is the class ID for no finding, and this provides a one-pixel bounding box with a confidence of 1.0).

### 2.1 Preprocessing

By investigating the training dataset, we observed that the observations labeled 14 (no finding) largely outnumber those labeled with abnormalities. In order to avoid problems resulting from imbalance data, we removed rows that are labeled 14 (no finding). The goal of this project can be accomplished without using images with no finding. It is sufficient to use images with certain findings and bounding boxes to fit a model and make predictions. We then resized the input images to 512x512 pixels. Then we calculate and normalize x-mid, y-mid, width and height of the bounding boxes, and set up the labels in correspondence to 14 classes. We prepared data this way to fit into the YOLO model.

### 2.2 Exploratory Data Analysis

We investigated the training dataset by plotting the distribution of labels. Figure 2 shows that Aortic enlargement (0) is the most common type of abnormality in our dataset. Other common types are Cardiomegaly (3), Pleural thickening (10), and Pulmonary fibrosis (13). Note that we removed the rows labeled no finding (14), which was initially the most common label in the training dataset.

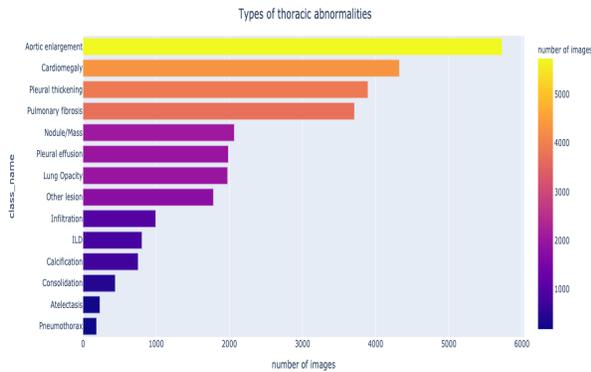


Figure 1: Distribution of Types of Abnormalities

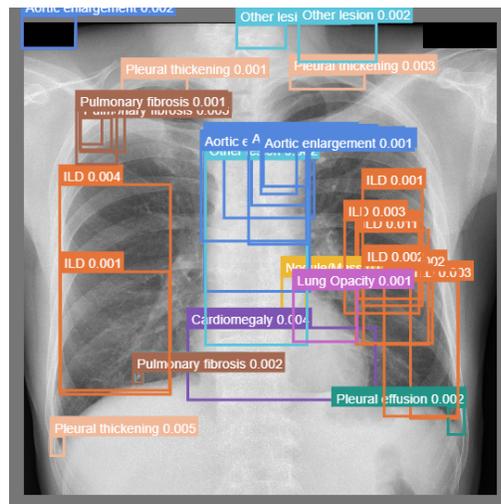


Figure 2: Proposed Bounding Boxes before suppression

### 2.3 Data Augmentation

We augmented our dataset by flipping the images vertically or horizontally, rotating 90 degrees, adding noise, blurring, and changing brightness or contrast respectively. We didn't apply random cropping since it may even hinder experienced doctors to detect abnormalities. The probability for each augmentation step is a hyperparameter that we can tune. We tried to populate our training set with 10%, 20%, 30%, 40%, and 50% augmented data respectively. We finalized on 20% since we didn't observe any further enhancement of mAP beyond 20%. Although there are more complex data augmentation methods applied in the field of study, we did not augment further because we require the augmented samples readable for professional doctors.

## 3. Architect

We mainly used the YOLOv5 [7] framework, which uses bounding boxes for object localization. YOLO ("You Only Look Once") is one of the most popular and most favorite algorithms for AI engineers. It always has been the first preference for real-time object detection. The architecture is shown below in Figure 3.

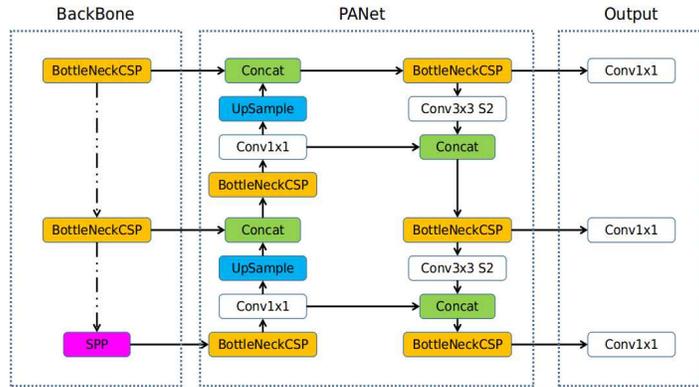


Figure 3: Overview of YOLOv5

## 4. Evaluation

Our evaluation metric will be  $mAP@0.5$  and  $mAP[0.5:0.95]$  for training and validation. Kaggle requires a  $mAP@0.4$  for leaderboard ranking, in other words our test set. We don't think this subtle difference in evaluation metric and actual loss function would affect our model selection, since mAPs at different IoUs are closely correlated to each other.

We achieved a  $mAP@0.5$  of 0.21 in our baseline model and boosted the performance to 0.30 with further preprocessing and data augmentation. We then select models from the Yolo family: YOLOv5s, YOLOv5x, YOLOv5l, etc. and get a final  $mAP@0.5$  of 0.34 with YOLOv5x and further hyperparameter tuning. Worth noticing, this 0.34 is derived from a test set that we split from the original training set rather than the leaderboard test set. We unfortunately cannot provide the latest leaderboard test set result due to the total number of submissions per day restriction on Kaggle.

	Baseline	Data Augmentation	Model Selection and Hyperparameter Tuning
$mAP@0.5$ - local test set	0.21	0.30	0.34
$mAP@[0.5:0.95]$ -local test set	0.08	0.13	0.16
$mAP@0.4$ -leaderboard test set	0.24	NA	NA

Table 1: Performance Improvement Progress

We once also implemented the train-dev set configuration in our model selection process and didn't gather sufficient statistical evidence to support the alternative hypothesis that the distribution discrepancy due to data augmentation impairs prediction performance. Hence, we let the entire validation set follow the distribution of the test set.

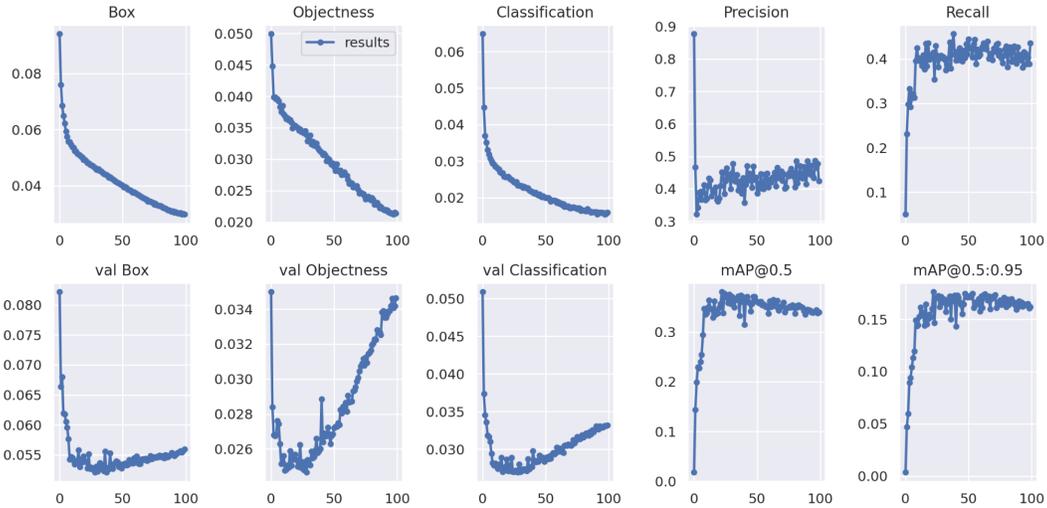


Figure 4: Traces of Metrics in Training Process

#### 4.1 Error Analysis

After thorough analysis of error profile, we believe that there are three types of errors:

1. Imbalance category population

As shown in Figure 5, our model predicts certain abnormalities reasonably well, while fails in others. Aortic Enlargement and Cardiomegaly are the two most well-predicted abnormalities. Other Lesion, Infiltration, and Atelectasis are the worst performers. In fact, as shown in Figure 2, this contrast is extremely likely a result of an imbalanced number of samples in these categories. To solve this issue, we tried to oversample the minority cases. However, those minority cases often coexist with common abnormalities and therefore hinder our oversampling. In future, we may have to crop the minority cases first and then oversample. We may also consider cropping a few common cases to make all categories in the training set have exposure to cropped samples.

2. Indistinguishable background

As shown in the last row of Figure 5, our model fails to spot many abnormalities and treats them as background. We don't have the necessary medical expertise to interpolate such errors, some of which may even be beyond top doctors' understanding. In future, we may ask medical experts for assistance to understand the underlying patterns of these errors.

3. Overlapping Objects

Yolo families cannot handle cases that a large number of objects overlap due to the limitation of anchor boxes. Unfortunately, we are facing exactly this problem as shown in Figure 6. However, Kaggle doesn't allow us to use computationally expensive models, so for this project, we stick with YOLOv5.

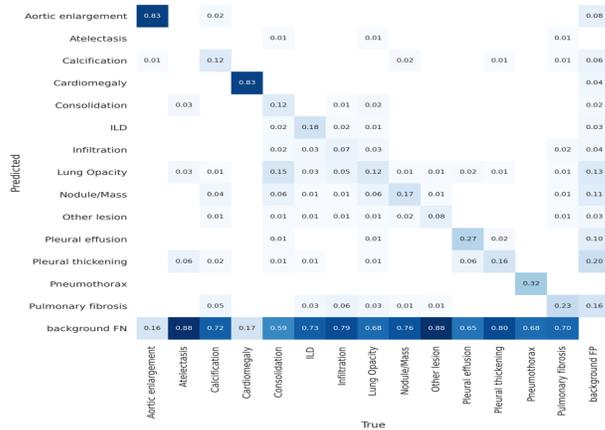


Figure 5: Confusion Matrix

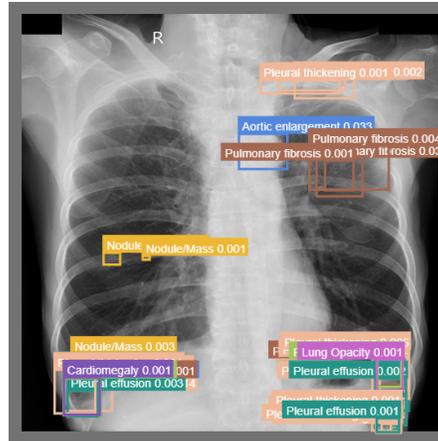


Figure 6: A Training Sample with Extensive Overlaps

## 4.2 Other Metrics

We reconfirm our previous observation that our model is especially predictive in Aortic Enlargement and Cardiomegaly through the lens of precision, recall, f1 score, and confidence.

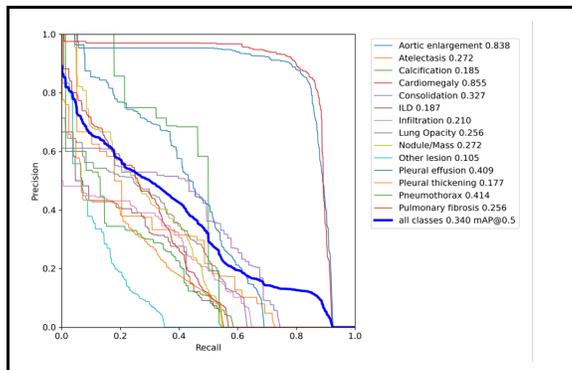


Figure 7: Precision Recall Curve

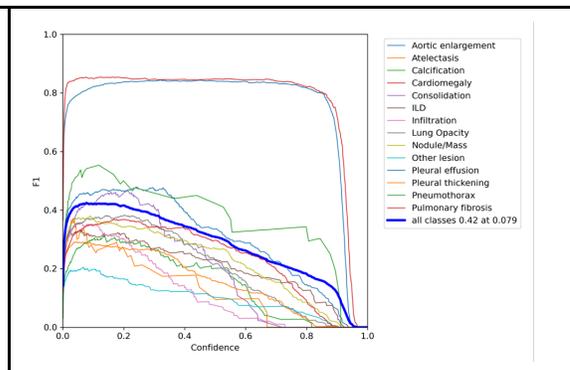


Figure 8: f1 Confidence Curve

## 5. Future work

We would like to conduct follow-up research to enhance performance by oversampling minority categories with cropping, further analyzing background related misclassification, and other models that handle overlapping well. We would also perform a more robust hyperparameter tuning, since the tuning process takes time longer than the scope of this project.

## 6. Contribution

**Heyang Huang:** General Project Setup (basic files, Kaggle environment, etc.), YOLO Implementation, Model Evaluation, Error Analysis, Final Writeup

**Yijun Long:** Preprocessing, Exploratory Data Analysis, YOLO Implementation, Final Writeup

**Yi Wei:** Data Augmentation, YOLO Implementation, Error Analysis, Final Writeup

## References

- [1] *VinBigData Chest X-ray Abnormalities Detection* | Kaggle. (2020). Kaggle. <https://www.kaggle.com/c/vinbigdata-chest-xray-abnormalities-detection/overview>
- [2] *VinDr-CXR: An open dataset of chest X-rays with radiologist's annotations*. (2012). <https://arxiv.org/pdf/2012.15029.pdf>
- [3] *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*. (2017). <https://arxiv.org/abs/1711.05225>
- [4] Rajpurkar, P., Irvin, J., Ball, R. L., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C. P., Patel, B. N., Yeom, K. W., Shpanskaya, K., Blankenberg, F. G., Seekins, J., Amrhein, T. J., Mong, D. A., Halabi, S. S., Zucker, E. J., ... Lungren, M. P. (2018). Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists. *PLOS Medicine*, 15(11), e1002686. <https://doi.org/10.1371/journal.pmed.1002686>
- [5] *Automated abnormality classification of chest radiographs using deep convolutional neural networks*. (2020). <https://www.nature.com/articles/s41746-020-0273-z>
- [6] *Chest Radiograph Interpretation with Deep Learning Models: Assessment with Radiologist-adjudicated Reference Standards and Population-adjusted Evaluation*. (2019). <https://pubs.rsna.org/doi/10.1148/radiol.2019191293>
- [7] *YOLOv5 in PyTorch*. (2020). Github. <https://github.com/ultralytics/yolov5>

## Code:

- [1] <https://www.kaggle.com/awsaf49/vinbigdata-cxr-ad-yolov5-14-class-infer>
- [2] <https://www.kaggle.com/nxhong93/yolov5-chest-512>
- [3] <https://www.kaggle.com/c/vinbigdata-chest-xray-abnormalities-detection>