RSNA Pneumonia Detection Challenge – Winning Model Documentation

Background on Team

Competition Name: RSNA Pneumonia Detection Challenge

Team Name: 16bit.ai / layer6

Private Leaderboard Score: 0.23901

Private Leaderboard Place: 4th

Our team consisted of a unique mix of engineers, computer scientists, and medical doctors, contributing to an exciting team dynamic and a creative mix of approaches.

Team Members:

Name	Location	Email
Daniel Eftekhari	Toronto, Canada	daniel@16bit.ai
Alexander Bilbily	Toronto, Ontario	alexander@16bit.ai
Mark Cicero	Toronto, Ontario	mark@16bit.ai
Lucas Pereira	Goiânia-GO, Brazil	lucaspereira0612@gmail.com
Guang Wei Yu	Toronto, Canada	guang@layer6.ai
Himanshu Rai	Toronto, Canada	himanshu@layer6.ai
Chundi Liu	Toronto, Canada	chundi@layer6.ai
Jason Chang	Toronto, Canada	jason@layer6.ai
Maks Volkovs	Toronto, Ontario	maks@layer6.ai

Summary

We developed an ensemble of three convolutional neural network object detectors (Mask RCNN, YOLOv3, and Faster RCNN architectures), in combination with a classification network (DenseNet-121 architecture) that served to reduce false positives, to detect pneumonia on chest x-rays (see Figure 1). We found that using a relaxed detection threshold for object detection, whilst requiring unanimous agreement among the detectors, effectively consolidated the need to minimize both false positives and false negatives. The classifier's detection threshold was computed by optimizing the area under the curve (AUC). Adaptive histogram equalization was used to improve image contrast as a data preprocessing step. We used age, sex, and view position as inputs into the penultimate layer of the classifier to improve performance.

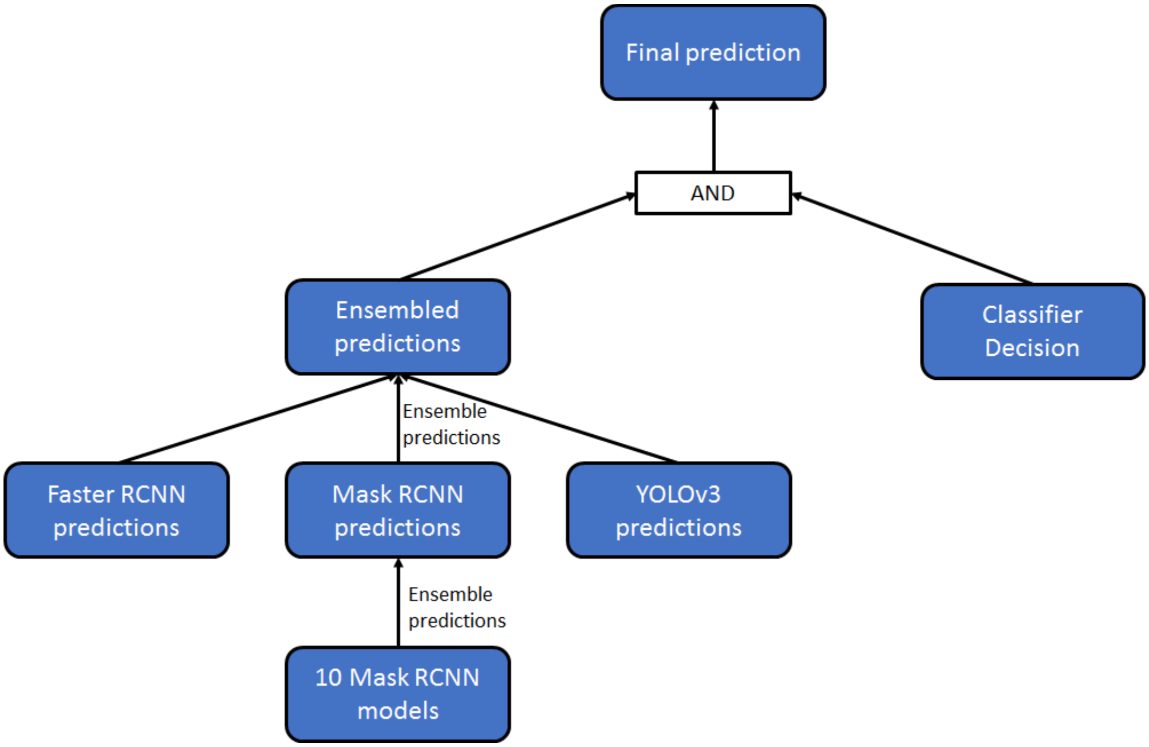

Figure 1 Our methodological pipeline for object detection. Three object detection algorithms were used to propose bounding box predictions, and a classifier served to minimize false positives.

Model Ensemble Approach & Interesting Findings

We ensembled our detection models in two stages. First, we took the intersection of the bounding box predictions of ten Mask RCNN (He et al. 2017) detectors, each trained on a subset of the training data. Concretely, we define an intersection as any set of pixels that are encompassed by the bounding box predictions of all ten models. Requiring agreement between the ten models helped reduce false positives. To reduce false negatives, which might occur if any one of the ten models did not output a bounding box, a relatively low bounding box confidence threshold was used.

Next, we took the ten bounding boxes which contributed to the resulting regions in the intersected image, and calculated a weighted, average bounding box. The weighted bounding box position for a given box side is given by:

$$y_i = \frac{1}{Z}\sum_{i=1}^{M} w_i f_i$$

where i indexes each of the $M = 10$ models, $w_i = \frac{1}{loss_i}$ is the inverse of the validation loss of model i obtained during training, and f_i is the box position prediction for model i. Finally, $Z = \sum_{i=1}^{M} w_i$ normalizes the coordinates.

The two approaches above result in two boxes: a small one (corresponding to the intersection), and a larger one (corresponding to the average). We hypothesized that the optimal bounding box would lie somewhere between the two. Therefore, we re-weighted the coordinates of the intersection and average bounding boxes using a 3:1 ratio. Figure 2 demonstrates this process.

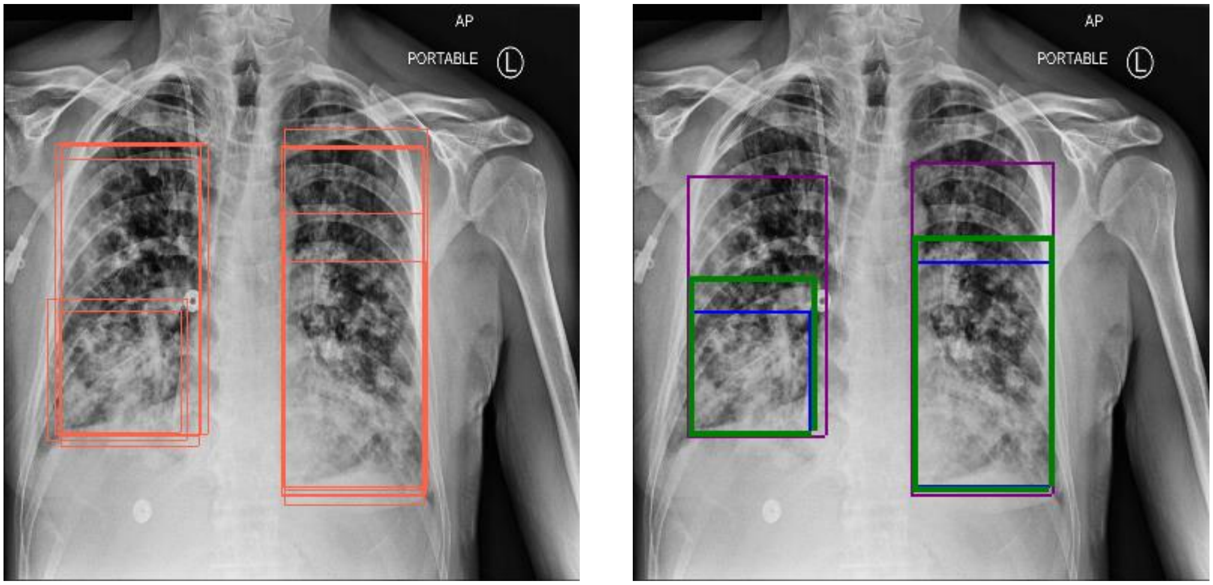

Figure 2 Model ensemble approach for Mask RCNN. Left) First, the ten Mask RCNN models output bounding boxes (in orange), after which their intersections are taken. This leads to the blue boxes in the Right) image. The average of

the connected bounding boxes is also calculated, which is given by the purple boxes. The final predictions, in green, are taken by re-weighting the blue and purple boxes.

This exact approach was repeated to consolidate the predictions of the combined ten Mask RCNNs, YOLOv3 (Redmon et al. 2015), and Faster RCNN (Ren et al. 2015), except without weighing the average bounding boxes using each model's inverse validation loss. We ensembled these models to minimize the possibility that the short-coming of any one model would have a significant impact on performance.

The rationale for intersecting different models was, in a crude sense, to emulate the approach used by the radiologists who labeled the data, as they took the intersection of their bounding box predictions. There is also theoretical justification for shrinking the box sizes: all three detection models use loss functions that are less sensitive to errors for large bounding box predictions than for smaller ones. Finally, note that the intersection is necessarily smaller than any of the contributing boxes. This had a positive effect, since we found that the stage 1 test images had much smaller bounding boxes than those of the stage 1 training set (likely because the two were labeled differently, see Figure 3).

We experimented with alternative definitions of intersection, for example where only a fraction of the detectors need to agree, as a means of reducing false negatives. However, our preliminary results showed this did not improve performance, and introduced other difficulties, such as having non-rectangular intersections. False negatives did affect performance to some extent, as the combined detectors sometimes did not produce a bounding box, despite the classifier labeling the image as a positive case. Similarly, for some positive cases, the classifier may have incorrectly predicted the image as normal/healthy, despite the object detectors outputting a bounding box. Future work would involve better consolidating the decision-making of the classifier and the detectors using a probabilistic approach, as opposed to using the two as hard decision rules. We also found that our bounding box predictions, on average, were well-centered but larger than the ground truth bounding boxes for the stage 1 test data. This suggests simply increasing the weight of the intersection vs. the weighted average bounding box may have further improved results.

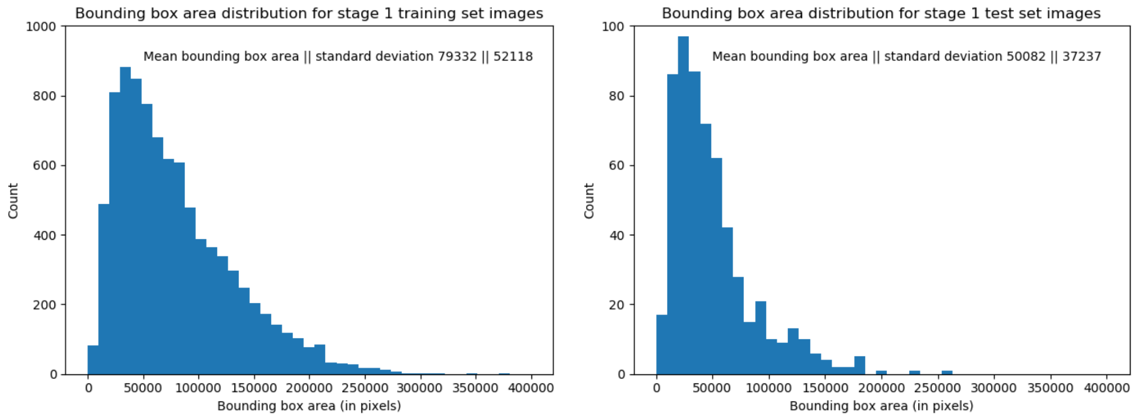

Figure 3 The statistics of the stage 1 training and test sets differed widely, possibly because of the different labeling strategies used. These two plots demonstrate one such difference: the average bounding box sizes are significantly larger in the training set than in the test set. This may explain why our ensemble approach, which intersects boxes, may have led to large improvements in performance.

Finally, we found the statistics of the training and (stage 1) test sets to differ widely, both in the number of pneumonia cases, and in terms of the bounding box sizes (see Figure 3). Ultimately, we decided to use the models that performed best on the stage 1 public leaderboard for our final model submission. Despite the risk of overfitting to the test set, we reasoned that due to differences in the labeling scheme of the training and test sets, this approach made sense. Note that we did not re-train our models when the stage 1 test set labels were released.

Model Execution Time

As is often the case with neural network-based solutions, training time is significantly greater than inference time. The whole training pipeline takes several hours on a cluster of three GPUs, but inference time per image takes no more than a few seconds.

Other Approaches

We experimented with a plethora of approaches before deciding on our final approach. We believe the diversity of approaches we investigated was a strength, and given more time to investigate, could have led to significant improvements in results. Below are some example approaches we investigated:

1) Intelligently reduce the confidence threshold for the detection of second and third bounding boxes in an image. The reason for this is that if one bounding box is

detected when using a high confidence threshold, the network should “search” or “pay closer attention” to possible second findings. We found this approach to be beneficial, but we did not end up pursuing it when we introduced the ensemble approach, as the latter approach allowed us to use a low confidence threshold for each detector. The need for collective agreement naturally led to improved confidence in the predictions.

2) Training on full resolution images. We implemented an approach to use the images at full resolution (1024 x 1024), as we hypothesized that the rich information contained in the full resolution images would be beneficial for accurate classification and detection. However, this idea presented its own challenges, including overfitting to the high dimensional input data, and fitting the full resolution images into GPU memory. In practice, we split the full resolution image into quadrants, made predictions on each quadrant separately, and then combined the predictions of adjacent quadrants by computing a maximum size rectangle within the convex hull of the predictions.
3) Training Mask RCNN on “negative region proposals”. The open source implementation of Mask RCNN we used does not make use of images without objects, thereby reducing our effective dataset size by around two thirds. We experimented with modifying the training procedure so that, for images with no ground truth bounding boxes, we would sample a random number of boxes (between one and three), each randomly sized (but in size similar to the pneumonia bounding boxes). This resulted in a second object category, one which only existed for images with no positive cases. Our initial results showed that this approach worked very well, but as we investigated this approach late in the competition, there was not enough time to integrate it into the full pipeline and test its performance adequately.
4) We investigated test time augmentation, in particular left-right flips, but our preliminary findings suggested this was not beneficial.
5) We concatenated age, sex, and view position channel-wise in the penultimate convolutional layer for the object detectors, then used a 1x1 convolution to resize the channels to the expected size. Our preliminary findings suggested this did not improve performance, but further investigation may be warranted.
6) Use the distribution of bounding box centers in the training set to make more informed decisions during inference. This approach could be of use because the distribution of bounding box centers in the training data may be predictive of the distribution in the validation/test sets. We modelled the prior probability of bounding box locations on the training set using a gaussian mixture model, with the centers of the two Gaussians located in the left and right lungs. For test-time bounding box predictions, we computed to which of the two clusters the predicted bounding box belonged to, then calculated the probability that the bounding box would occur in that position given the distribution of bounding box locations in the training set. This probability could then serve as a prior probability (in the Bayesian sense) when deciding which bounding boxes to keep (see Figure 4 for a heat map of the

prior probabilities). One challenge with this approach is how to meaningfully integrate the prior probability in the object detection decision-making process. We did not have sufficient time to investigate this approach in full.

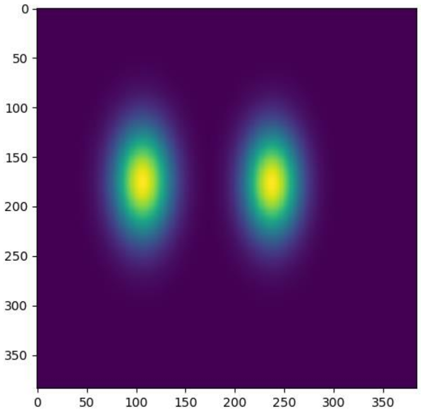

Figure 4 Heap map of the prior probability of bounding box centers in the stage 1 training set, assuming a mixture of two Gaussians is used to model the density.

Data Preprocessing, Model Architecture, and Training

We applied adaptive histogram equalization to improve local image contrast as a pre-processing step. Figure 5 shows the effect of adaptive histogram equalization on a sample image.

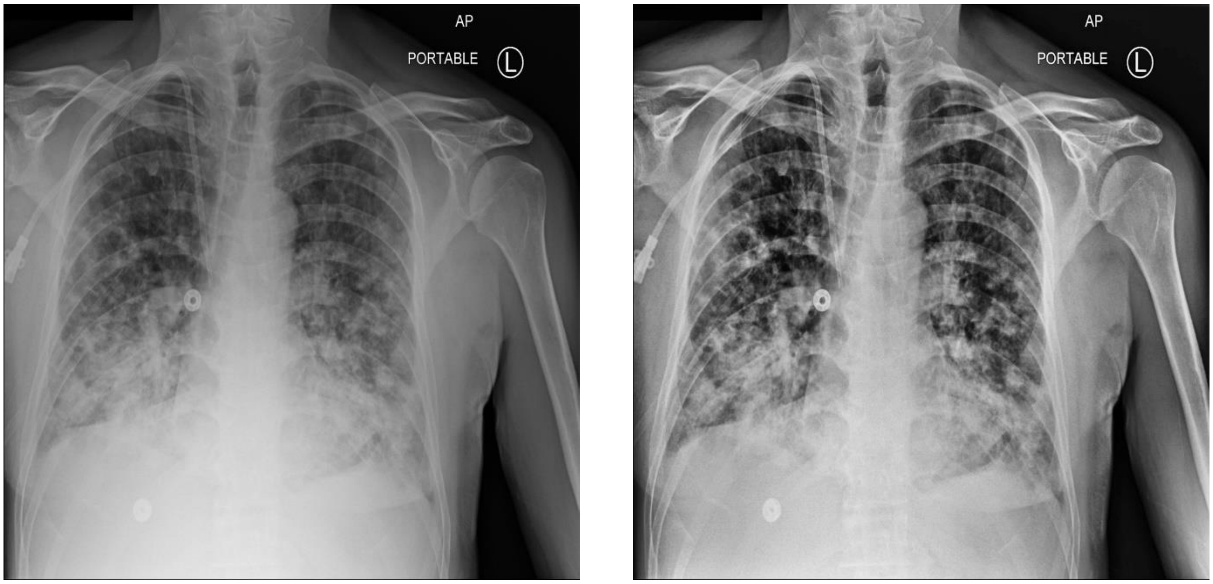

Figure 5 The effect of adaptive histogram equalization. Left: Original image, Right: Adaptive histogram equalized image.

We used Resnet50, with 384x384 image sizes, and the stochastic gradient descent (SGD) with momentum optimization algorithm when training Mask RCNN. The learning rate was reduced multiplicatively during validation loss plateaus. We found that larger architectures did not improve performance, possibly due to the relatively small dataset size. We used Inception Resnet V2 with atrous convolutions, 256x256 image sizes, and the SGD with momentum optimization algorithm when training Faster RCNN. We used NASNet with 256x256 image sizes and the Adam optimization algorithm when training YOLOv3. For the classifier, we used DensetNet-121 (Huang, Liu, and Weinberger 2016), with 256x256 image sizes, and the Adam optimization algorithm. Data augmentation during training consisted of horizontal flips, affine transformations, and pixel-wise intensity multiplications. The classifier was pre-trained on the NIH open source dataset.

Acknowledgements

We would like to acknowledge the competition organizers, and our technical sponsor NVIDIA for the donation of the GPUs we used.

References

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross B Girshick. 2017. “Mask {R-CNN}.” *CoRR* abs/1703.06870. http://arxiv.org/abs/1703.06870.

Huang, Gao, Zhuang Liu, and Kilian Q Weinberger. 2016. “Densely Connected Convolutional Networks.” *CoRR* abs/1608.06993. http://arxiv.org/abs/1608.06993.

Redmon, Joseph, Santosh Kumar Divvala, Ross B Girshick, and Ali Farhadi. 2015. “You Only Look Once: Unified, Real-Time Object Detection.” *CoRR* abs/1506.02640. http://arxiv.org/abs/1506.02640.

Ren, Shaoqing, Kaiming He, Ross B Girshick, and Jian Sun. 2015. “Faster {R-CNN:} Towards Real-Time Object Detection with Region Proposal Networks.” *CoRR* abs/1506.01497. http://arxiv.org/abs/1506.01497.

Open Source Software and External Datasets

Open source software and pretrained weights:

https://github.com/matterport/Mask_RCNN

https://github.com/qqwweee/keras-yolo3

https://github.com/tensorflow/models/tree/master/research/object_detection

https://github.com/arnoweng/CheXNet

External Datasets:

https://www.kaggle.com/nih-chest-xrays/data