Project: Pneumonia Detection

Yucong Zhang, Steven Guo
{yz646, steven.guo}@duke.edu

Abstract

In this project, we used RetinaNet to detect pneumonia on chest X-ray images. We used focal loss, compound loss, and pre-trained models to improve the performance of the network. Images were downsampled to meet the hardware constraint. The experimental results show that we can have at most 15% improvement with regard to our baseline model. In the end, we also discussed how the performance changes with the training epochs and the confidence levels of the predicted bounding boxes.

1. Overview

Pneumonia is an infection of the lung. This disease makes the lungs fill with fluid and makes it difficult for patients to breathe. 16% deaths of children under 5 years old were caused by it in 2015 [1]. This is approximately 2400 deaths per day. The conventional chest X-ray images lack contrast for soft tissue like lungs [2]. That makes standard diagnosis slow and inefficient. However, with modern computing technology this process can be more efficient and help doctors to provide timely treatments.

The position, shape, and size of the infection area can vary a lot [3]. This makes its image very vague, hard to be detected, and hard to locate the infections. These challenges are major research focuses. Nowadays, the object detection algorithms fall into 2 categories: one-stage object detection like You Only Look Once (YOLO) and Single Shot Detector (SSD), and two-stage object detection like region-based CNN (R-CNN) and Feature Pyramid Network (FPN) [4].

In this project, RetinaNet, which was proposed by Lin et al. in 2018 [5] is used to detect pneumonia. It uses ResNet and Feature Pyramid Network as backbones for feature extraction and another 2 subnetworks for classification and bounding box regression. Some compromises were made to meet the hardware and time constraints. Also, some improvements are made to increase the performance of the network, including using pre-trained models, compound loss, different ResNet models, etc.

2. Methods

In this project, we choose RetinaNet as our basic model due to its fast inference and good performance. RetinaNet is a one-stage object detection model that skips the region proposal stage of common two-stage models. Compared to the traditional one-stage models, RetinaNet contains an extra component called Feature Pyramid Network (FPN). The whole architecture of RetinaNet is shown in Figure 1.

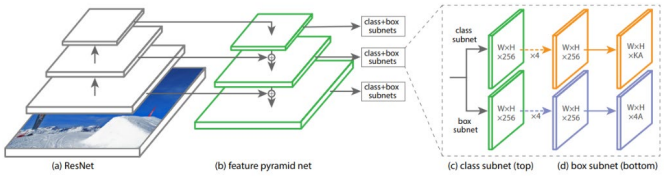


Figure 1 RetinaNet architecture

Generally speaking, it consists of four key components, the backbone model, FPN, the classification subnet, and the regression subnet. First, the image will go through a backbone model, which can generate feature maps of various sizes. Then, all the feature maps will be passed into FPN to further explore the features for both classification and bounding box prediction tasks. Finally, the output of the key features by FPN will be sent to two nets, one for predicting the probability of the presence of the bounding box as well as the object classes (classification subnet), and the other one for predicting the offset of the bounding boxes (regression subnet).

2.1 Backbone Model

The backbone model is crucial for object detection tasks since it has the responsibility to select the key regions that contain much information. Convolutional Neural Network (CNN) naturally has a pyramid-like structure that can compute the feature maps as the model goes deeper, and those features can be used to take the place of the traditional hand-crafted ones. In this sense, CNNs are one of the commonly picked models to be the backbone of the RetinaNet. In order to achieve high computation efficiency, we adopted the residual structure to build the CNN, namely ResNet.

Since the backbone model is used for extracting useful feature maps, it can be pre-trained solely on some other dataset before training the RetinaNet. Later in Section 2.3 and Section 3, we will show how we design the experiments and how the attribute of 'pre-trained' can affect the overall RetinaNet's performance.

2.2 Feature Pyramid Network

Feature Pyramid Network (FPN) is designed to strengthen the features extracted by the backbone model. In the RetinaNet framework, since the backbone model has the pyramid-like structure, the features output by the backbone model are also processed in a pyramid-like structure. Figure 2 has shown four different types of pyramid-like structure. (a) is computational intensive, which needs lots of human effort. (b) and (c) are fast and robust, but they do not utilize the multi-scale feature between different layers. (d) is FPN, which has a top-down pathway that combines the low- and high-resolution features, and thus enables the multi-scale feature generation.

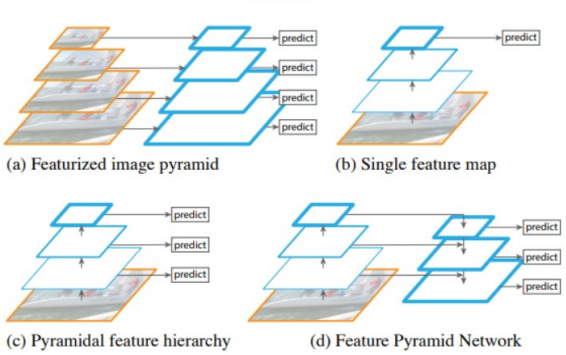


Figure 2 Four types of pyramid-like structures

2.3 Classification Subnet

This is a simple CNN that is used to predict the probability of an object being present at each spatial location for each anchor box and object class. In our project, there are only two classes, pneumonia for positive samples and non-pneumonia for negative samples. After the output layer, the probability of each object class for each anchor box will be derived.

2.4 Regression Subnet

Like the classification subnet, it is also a simple CNN. The only difference lies in the output layer. Since this subnet is used for predicting the offset of the bounding boxes, the number of channels, i.e. number of features per spatial location, used in the output layer equals to 4 times the number of anchors used.

3. Improvements

In total, we have made three improvements. In Section 2.5.1, we show how we have modified the binary cross entropy loss into an enhanced version, namely focal loss. In Section 2.5.2, we further add a penalty to the total loss, so that the bounding boxes can be better predicted and generated. Finally, Section 2.5.3 describes the pre-trained backbone model used in our experiment.

3.1 Focal Loss

Focal Loss is an enhancement over cross-entropy loss. It is used to handle the class imbalance in one-stage models. In our project, focal loss is very important due to the dense sampling of anchor boxes with only few target boxes. In general, focal loss has the ability to reduce the loss contribution from easy samples and increases the importance of correcting misclassified samples. We change the binary cross entropy loss after the classification subnet with the focal loss by multiplying the coefficients to the negative samples. It can be calculated as follows:

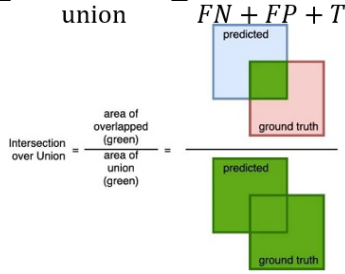
$$FL(p_t) = -\alpha \cdot {p_t}^{\gamma} \cdot log(1 - p_t)$$

where p_t is the probability of the sample t being positive, α and γ are parameters. In our experiment setting, we use $\alpha = 0.25$, and $\gamma = 2$.

3.2 Compound Loss

We experimented with a compound loss, a combination of different losses [6], to improve the model's performance. The focal loss is combined with intersection over union (IoU) loss [7]. IoU loss is a direct optimization of the overlapping area of the predicted bounding box and the ground truth. It is a simple and effective loss since the goal of the detection is to maximize the IoU of the prediction box and the ground truth box. The IoU and IoU loss can be formulated as the following:

$$\text{IoU} = \frac{\text{intersection}}{\text{union}} = \frac{TP}{FN + FP + TN}$$



$$\mathcal{L}_{IoU} = 1 - IoU$$

where TP is the true positive, FN is false negative, and FP is false positive. Then we create a compound loss from focal loss and the IoU loss as follows:

$$\mathcal{L}_{FLIoU} = \mathcal{L}_{FL} - IoU$$

The -IoU term can be seen as a penalty term that we want to minimize.

3.3 Pre-trained Backbone Model

As previously mentioned in Section 2.1, the backbone model is crucial to RetinaNet, since it can extract the feature maps in advance. If the backbone model is pre-trained, the feature maps can seize more valuable information. In the project, we experimented on two settings, the ResNet model pre-trained on ImageNet dataset and the ResNet model pre-trained on the RSNA kaggle dataset.

4. Experiment

4.1 Dataset and Settings

In the project, we used the kaggle dataset provided by Radiological Society of North America (RSNA). It has 6012 positive CT images and 20672 negative CT images, where positive images show the pneumonia and negative images show the normal lungs.

The image is downsampled from 1024 by 1024 to 128 by 128 and the bounding box is also resized accordingly. This was done due to (1) the hardware constraint: the graphics we were provided only came with 6000MB of memory, only 2 images could fit into one batch and (2) the time limitation: the large image dataset and small number of batch size force us to get one result after days.

Since we do not have a test dataset, thus we manually split the dataset into a train dataset and a test dataset, where the train dataset contains 21454 images and test dataset contains 5230

Table 1: The average precision (AP) of all models regarding different IoU values. The Baseline model uses the ResNet34 model as the backbone without pre-trained on any dataset, and it is trained by binary cross entropy loss.

Model	Pretrained (CT)	Pretrained (Imagenet)	Focal Loss	Compound Loss	AP@0.3	AP@0.4	AP@0.5	AP@0.6	AP@0.7	mAP
Baseline	-	-	-	-	0.3300	0.2478	0.1441	0.0699	0.0201	0.1624
ResNet34	-	-	√	-	0.3481	0.2728	0.1801	0.0803	0.0207	0.1804
	-	√	√	-	0.3651	0.2851	0.1897	0.0992	0.0266	0.1931
	-	√	-	√	0.4108	0.3146	0.2011	0.0945	0.0284	0.2099
	√	-	√	-	0.5652	0.4406	0.2981	**0.1508**	**0.0439**	0.2973
	√	-	-	√	**0.5727**	**0.4662**	**0.3144**	0.1383	0.0377	**0.3058**
ResNet50	-	-	√	-	0.2840	0.2274	0.1535	0.0811	0.0246	0.1542
	-	√	√	-	0.4079	0.3207	0.2177	0.1011	0.0289	0.2153
	-	√	-	√	0.4142	0.3374	0.2280	0.1258	0.0424	0.2296
	√	-	√	-	0.5274	0.4260	**0.2883**	**0.1437**	**0.0415**	0.2854
	√	-	-	√	**0.5612**	**0.4452**	0.2861	0.1416	0.0399	**0.2948**

images. Moreover, to be fair, we force the ratio of positive and negative samples to be the same with both dataset.

Regarding the model training settings, we used two versions of the ResNet models, ResNet34 and ResNet50 as the backbone models. ResNet50 is almost the same with ResNet34, except for the bottleneck blocks. We train the RetinaNet for 15 epochs, with a learning rate equal to 0.0001. When training the bounding boxes, we set the number of the anchors to be 9. During the inference stage, we use the bounding boxes with the confidence score > 0.05.

4.2 Metric and Results

We adopt the commonly used metric for the object detection task. We evaluate the model using the average precision (AP) at different intersection-over-union (IoU) levels. The AP is calculated by the area under the precision and recall curve. And the mean of AP (mAP) is calculated by the sum of all APs divided by the total number of different IoU levels. In this project, we have calculated the AP with IoU level = 0.3, 0.4, 0.5, 0.6, and 0.7.

The overall results are shown in Table 1. From Table 1, we can see that for both ResNet34 and ResNet50, the AP is getting better if the backbone model is pre-trained. Moreover, it is shown in the table that the model pre-trained on the CT images provided by the RSNA kaggle dataset can have a better performance than the model pre-trained on the ImageNet dataset. It is reasonable since the ImageNet dataset does not contain any CT images and it only contains images with RGB format. However, the object detection task requires to deal with gray-scale images, i.e. CT image.

It is clear to see that if we change the loss function with the compound loss, the overall model performance can be slightly improved. This is because we add some penalty into the loss, thus forcing the IoU between the predicted bounding boxes and the target boxes to be greater. In this case, the predicted bounding boxes will move towards the target boxes, and thus increase the AP and mAP.

5. Ablation Study and Discussion

In this project, there are lots of parameters to be tuned, and we have studied some of them to see how they can affect the AP, precision, recall and accuracy. Here, we only demonstrate one or two figures. For more figures, please check Appendix A and Appendix B.

The following sections are arranged as follows. Section 5.1 shows how the performance might change with the training epoch. We want to know that during the training of the model, how the performance will be affected. In Section 5.2, we study the effect of the box confidence scores. Although we set the threshold of the confidence score for the valid boxes to be 0.05, we would still like to know how the different threshold on box confidence scores might affect the performance.

5.1 Epoch tuning with box confidence > 0.5

AP vs. Epochs

As shown in Figure 3, the AP under all the IoU circumstances will first grow then drop, which means that the model might suffer from the overfitting problem. And it is clear to see that when the restriction on the IoU value is getting greater, the AP is getting smaller.

Precision vs. Epochs

Interestingly, we see that the precision keeps dropping regarding the number of epochs when we set the confidence threshold to be 0.5. It is reasonable since 0.5 is a rather big value for confidence score, and the bigger it is, the more accurate the predicted bounding boxes will be. Hence, in Figure 3, the precision is high at first with few pure high-score bounding boxes, but with more epochs, more bounding boxes with high confidence scores will be output, thus causing the precision to drop.

In fact, in other cases with different box confidence thresholds, the precision is getting bigger as the epoch increases (see Appendix A.1 and Appendix B.1). However, as the confidence threshold increases, the increasing trend will be smoother.

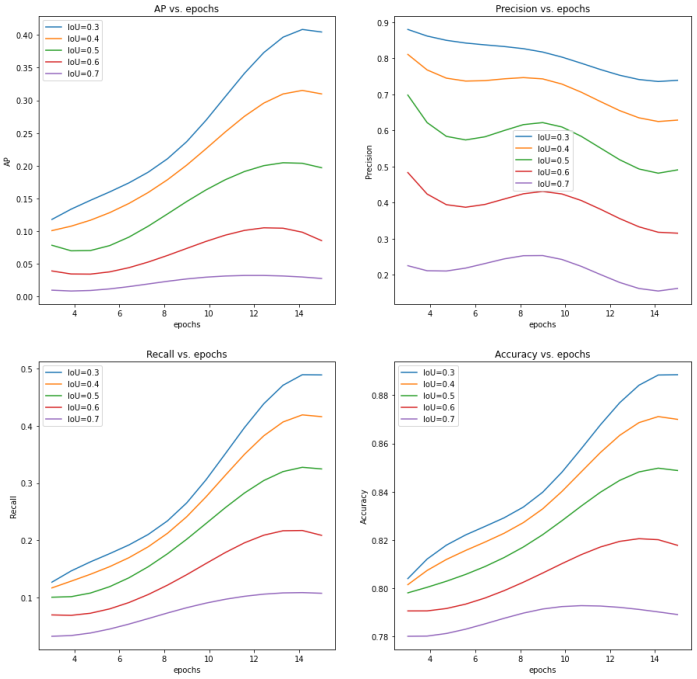


Figure 3 Epoch tuning

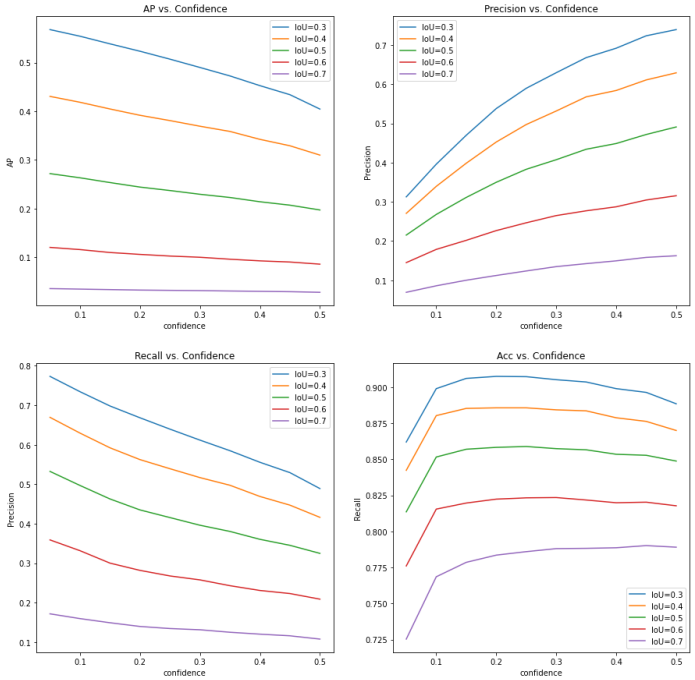


Figure 4 Box confidence tuning

Recall vs. Epochs

In Figure 3, since the confidence threshold is set to 0.5, the number of predicted boxes produced by the model is very small. Hence, at first, the recall is very small. With the epoch number getting larger, more and more boxes will have confidence threshold greater than 0.5, and the recall is getting larger accordingly.

However, this is not the case for all the confidence thresholds. In Appendix A.1 and Appendix B, the recall will first grow as the epoch increases, then it will drop. This is also reasonable, since when the confidence threshold is not so strict, more bounding boxes will be output, and must include the correct boxes. But as the number of epochs increases, the model is getting strict on output bounding boxes, thus causing the recall to drop.

Accuracy vs. Epochs

The accuracy is getting larger with the increasing epochs. This means that the model is getting a better and better classification ability on the pneumonia symptom.

5.2 Box Confidence tuning at epoch=15

AP vs. Confidence

AP is getting smaller as the box confidence threshold gets bigger. This is because when the threshold is larger, few predicted boxes will be viewed as valid, which might affect both the precision and the recall. As a result, AP might be affected.

Precision vs. Confidence

With the confidence threshold getting larger, the predicted boxes are more accurate. Hence, it is reasonable that the precision is getting larger with the increasing confidence score.

Recall vs. Confidence

Recall is getting lower when the confidence threshold increases. This result follows the natural trade-off between precision and recall. Since the precision grows as shown in Figure 4, the recall drops accordingly. Intuitively, this is because when the confidence threshold increases, the number of valid predicted boxes drops, thus making the recall low.

Accuracy vs. Confidence

The accuracy is getting bigger as the confidence level grows in the beginning, but it drops as the level keeps growing. Intuitively speaking, at first, if the confidence level is high, then the valid predicted boxes are accurate with high probability. In other words, the image is regarded as a pneumonia image if and only if the output probability is high enough. Hence, as the confidence threshold increases, the accuracy will grow. However when the confidence level is too high, then some of the correctly predicted boxes that have low probability will not be regarded as valid, which will harm the accuracy.

6. Conclusion

In this project, we have built a RetinaNet for pneumonia detection. We have made several adjustments to the original RetinaNet. Images are downsampled to meet the hardware constraints and

improve the overall training efficiency. A compound loss function composed with focal loss and IoU was used. The backbone ResNet is replaced with a pre-trained classification model. The 2 improvements we made significantly increased the performance of the model. The mAP increased by 15%. Our approach is simple and effective. The performance of the proposed model on downsampled images is also very close to some state-of-the-art models [8].

References

[1] American Thoracic Society. (2019). Top 20 pneumonia facts—2019. Retrieved April 27, 2022, from https://www.thoracic.org/patients/patient-resources/resources/top-pneumonia-facts.pdf

[2] Zhou, S. A., & Brahme, A. (2008). Development of phase-contrast X-ray imaging techniques and potential medical applications. *Physica Medica*, *24*(3), 129-148.

[3] Kallet, R. H. (2015). The vexing problem of ventilator-associated pneumonia: observations on pathophysiology, public policy, and clinical science. *Respiratory Care*, *60*(10), 1495-1508.

[4] Lohia, A., Kadam, K. D., Joshi, R. R., & Bongale, A. M. (2021). Bibliometric Analysis of One-stage and Two-stage Object Detection. *Library Philosophy and Practice*, 1-32.

[5] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).

[6] Ma, J. (2020). Segmentation loss odyssey. *arXiv preprint arXiv:2005.13449*.

[7] Rahman, M. A., & Wang, Y. (2016, December). Optimizing intersection-over-union in deep neural networks for image segmentation. In *International symposium on visual computing* (pp. 234-244). Springer, Cham.

[8] Yao, S., Chen, Y., Tian, X., & Jiang, R. (2021). Pneumonia Detection Using an Improved Algorithm Based on Faster R-CNN. *Computational and Mathematical Methods in Medicine*, *2021*.

Appendix A.1 Using Pre-trained model on ImageNet

Figure of AP, Precision, Recall and Accuracy vs. Epochs (box confidence>0.05)

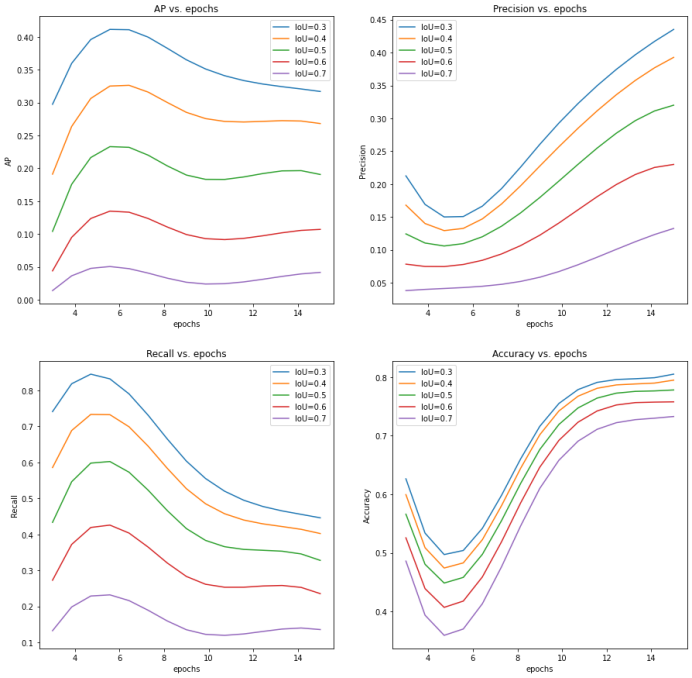


Figure of AP, Precision, Recall and Accuracy vs. Epochs (box confidence>0.25)

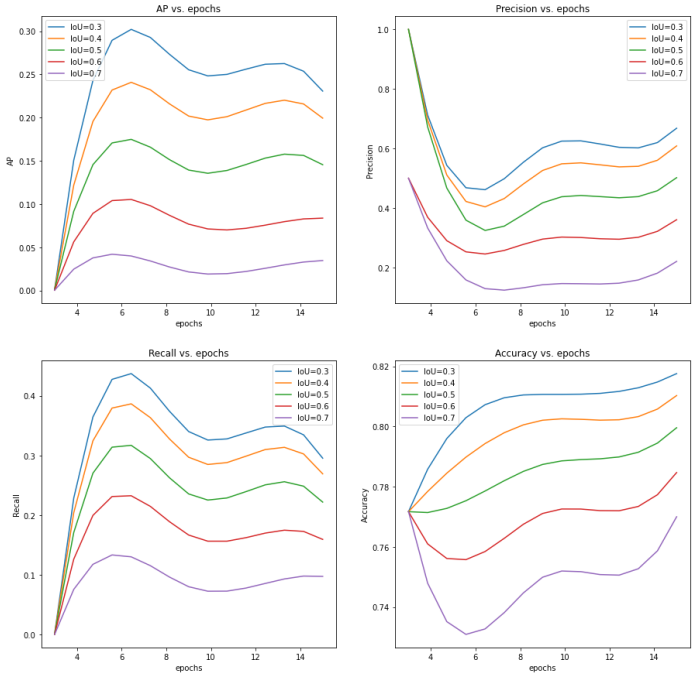


Figure of AP, Precision, Recall and Accuracy vs. Epochs (box confidence>0.15)

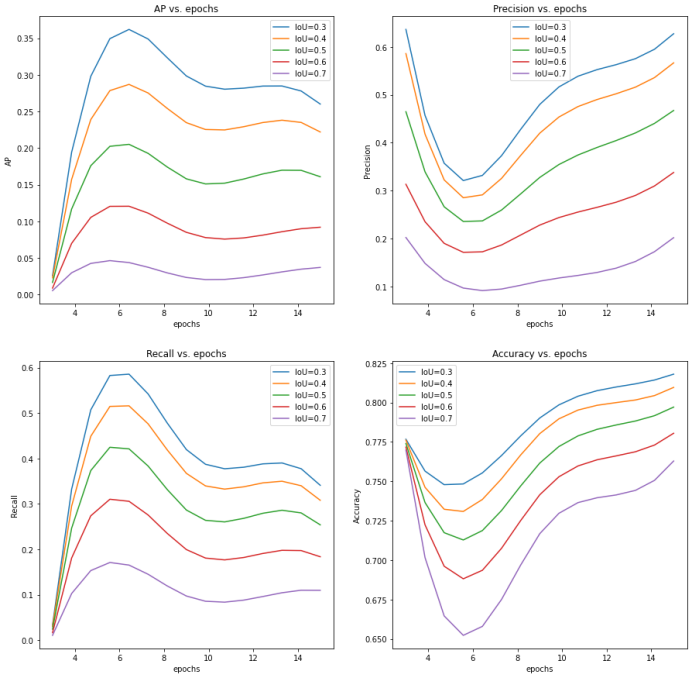
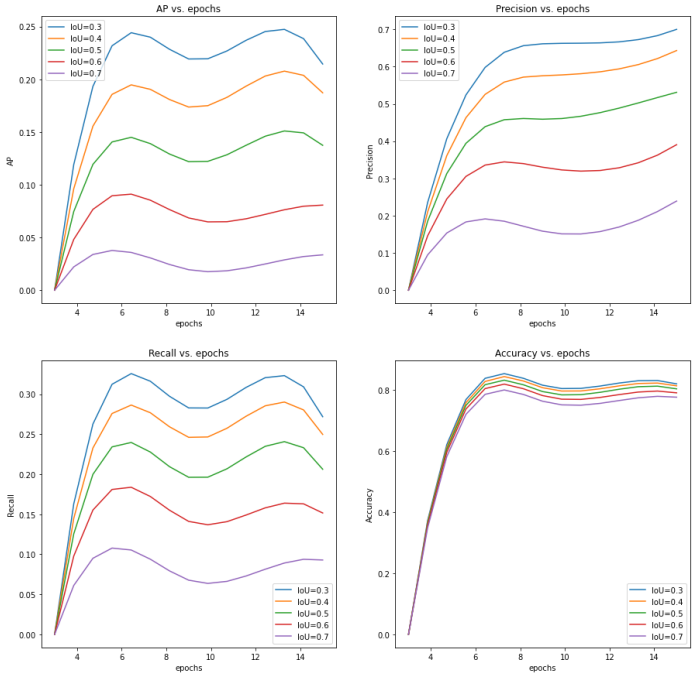


Figure of AP, Precision, Recall and Accuracy vs. Epochs (box confidence>0.35)



Appendix A.2. Using Pre-trained model on ImageNet

Figure of AP, Precision, Recall and Accuracy vs. Confidence (Epoch=3)

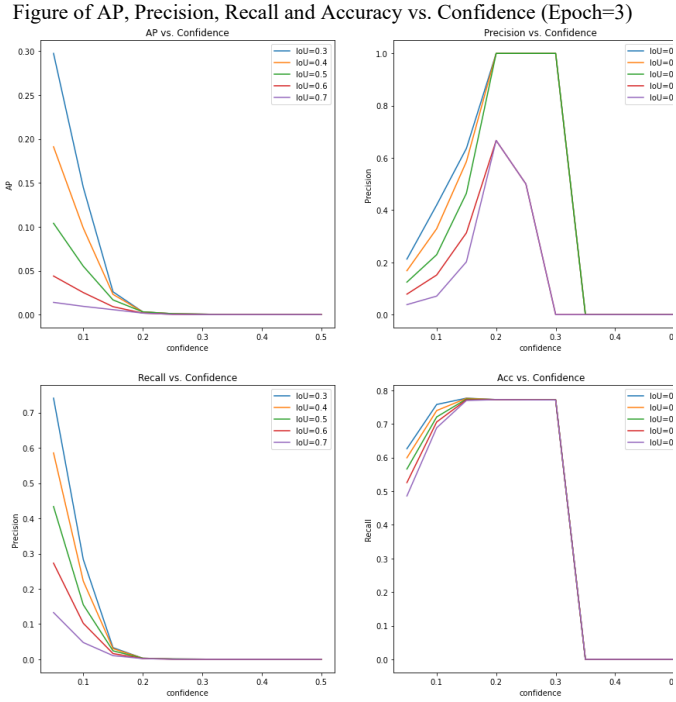


Figure of AP, Precision, Recall and Accuracy vs. Confidence (Epoch=6)

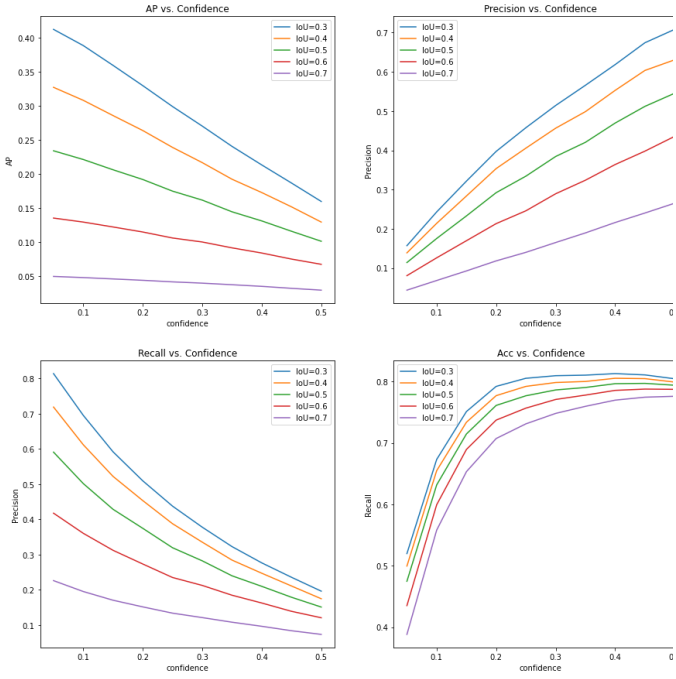


Figure of AP, Precision, Recall and Accuracy vs. Confidence (Epoch=9)

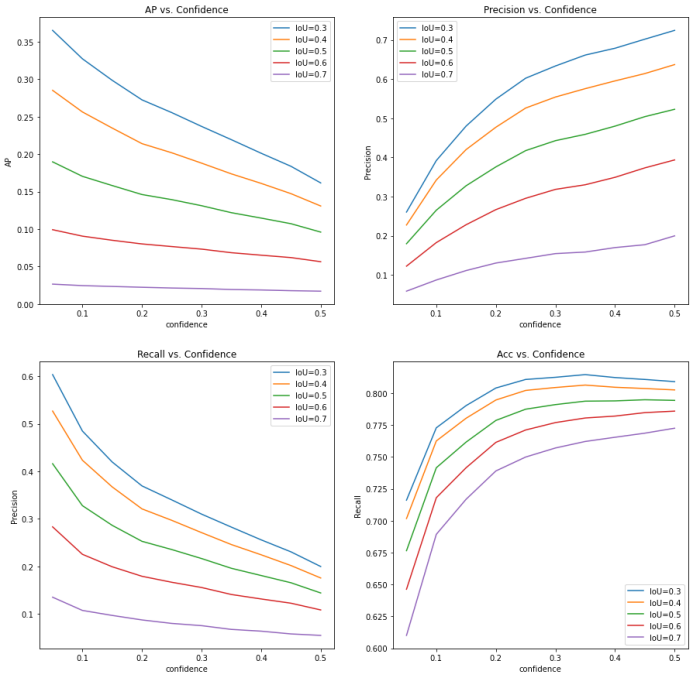
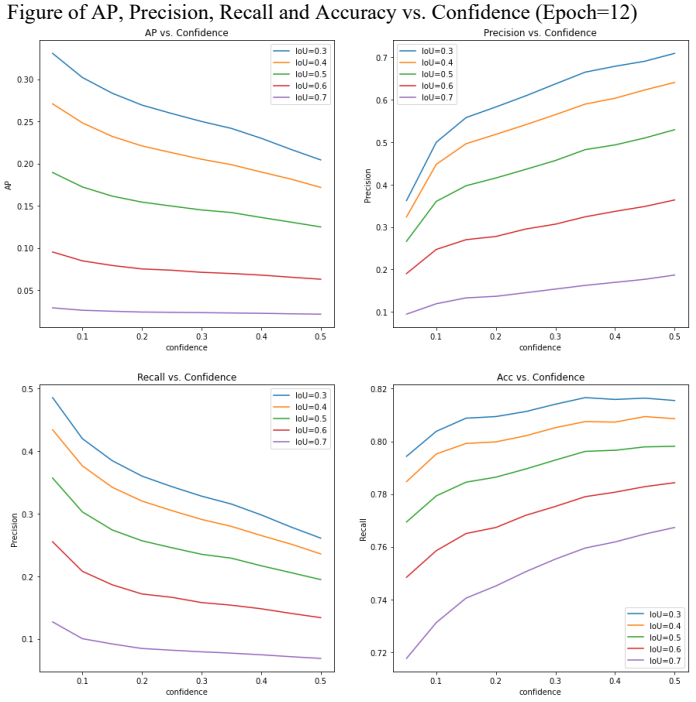


Figure of AP, Precision, Recall and Accuracy vs. Confidence (Epoch=12)



Appendix B.1 Using Pre-trained model on RSNA CT images

Figure of AP, Precision, Recall and Accuracy vs. Epochs (box confidence>0.05)

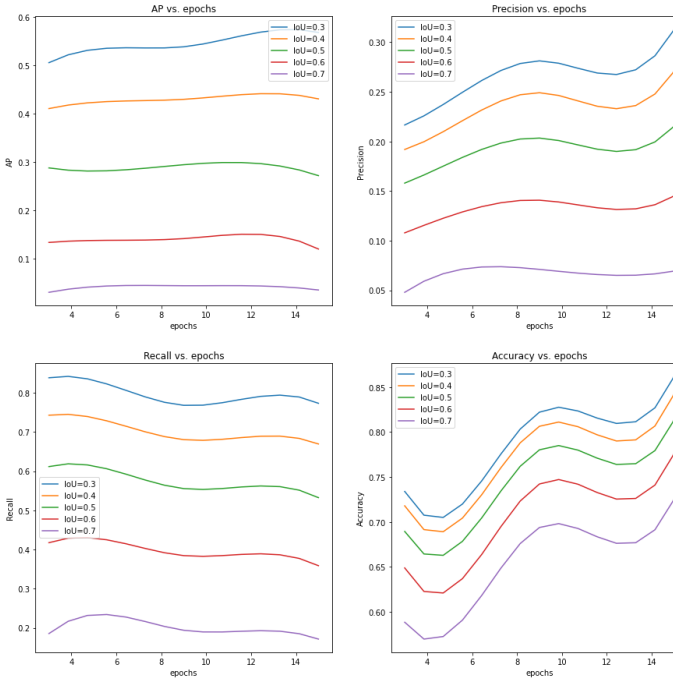


Figure of AP, Precision, Recall and Accuracy vs. Epochs (box confidence>0.25)

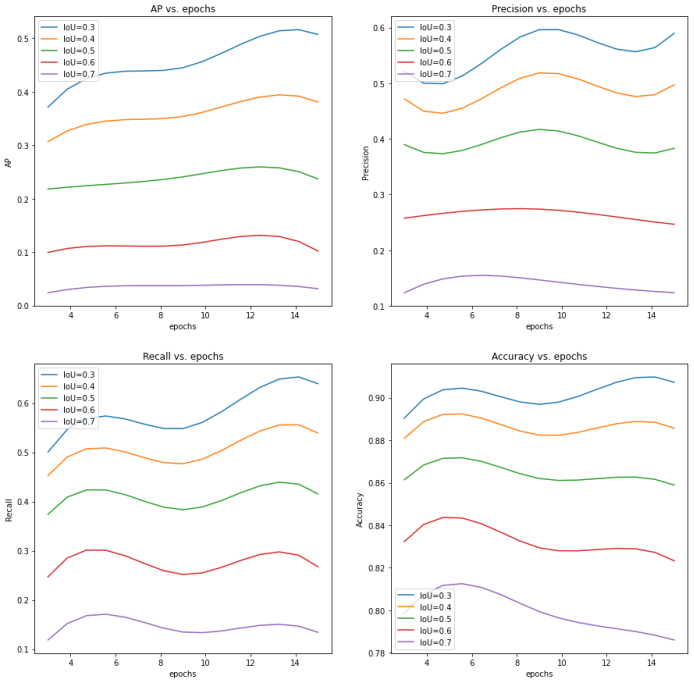


Figure of AP, Precision, Recall and Accuracy vs. Epochs (box confidence>0.15)

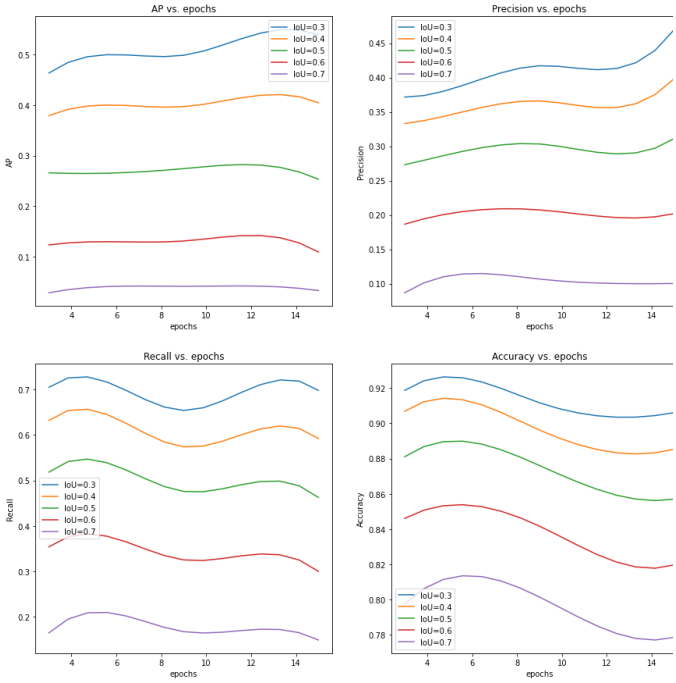
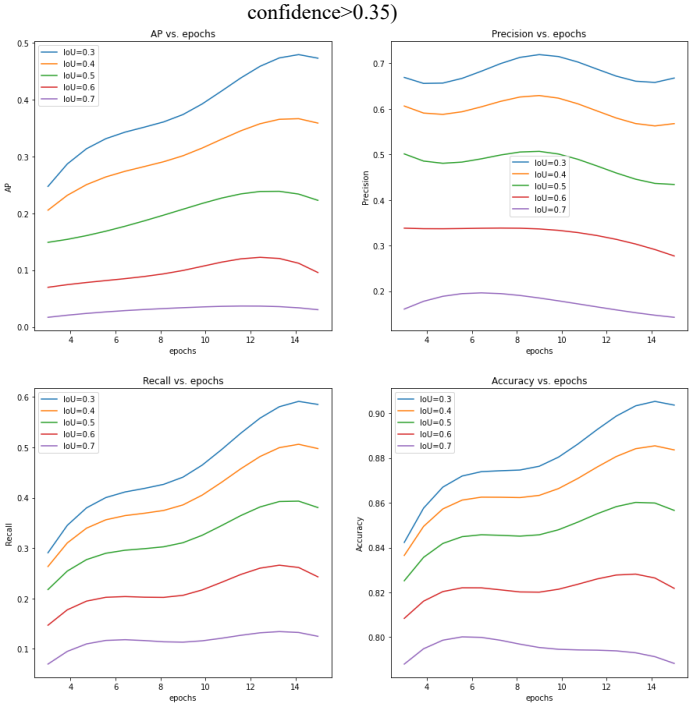


Figure of AP, Precision, Recall and Accuracy vs. Epochs (box confidence>0.35)



Appendix B.2 Using Pre-trained model on RSNA CT images

Figure of AP, Precision, Recall and Accuracy vs. Confidence (Epoch=3)

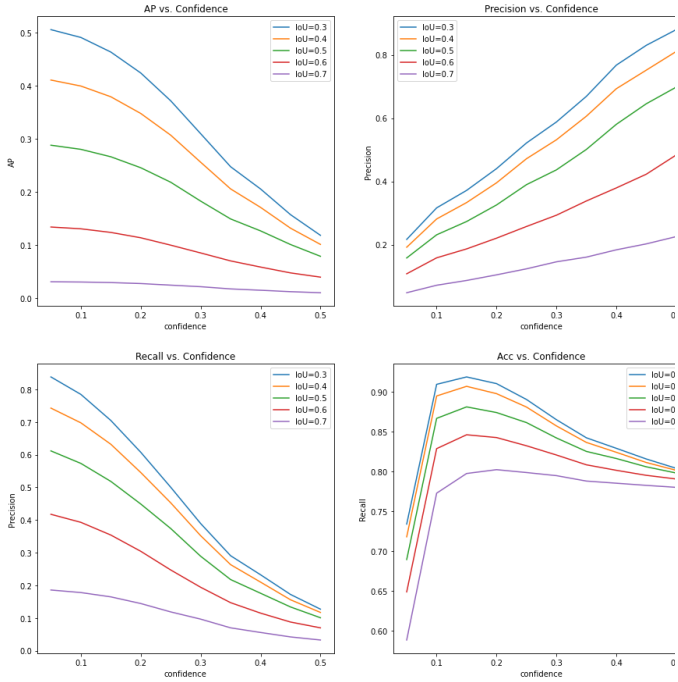


Figure of AP, Precision, Recall and Accuracy vs. Confidence (Epoch=6)

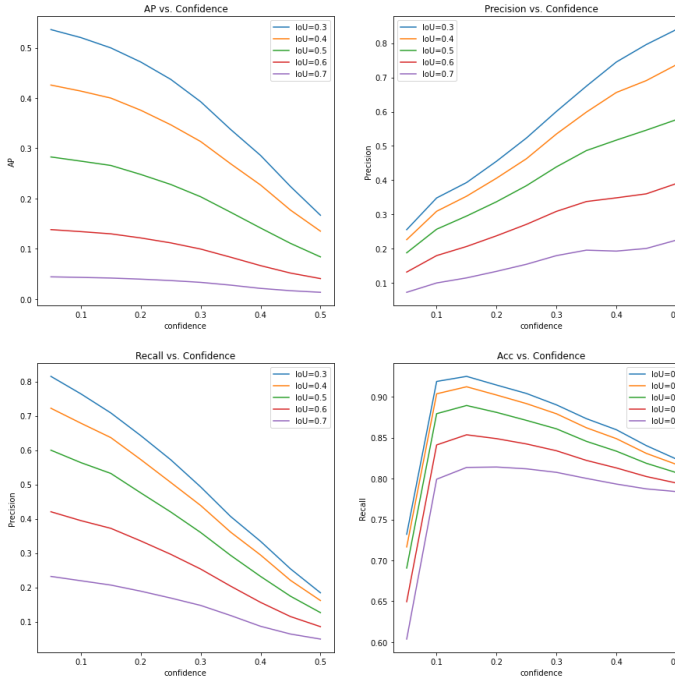


Figure of AP, Precision, Recall and Accuracy vs. Confidence (Epoch=9)

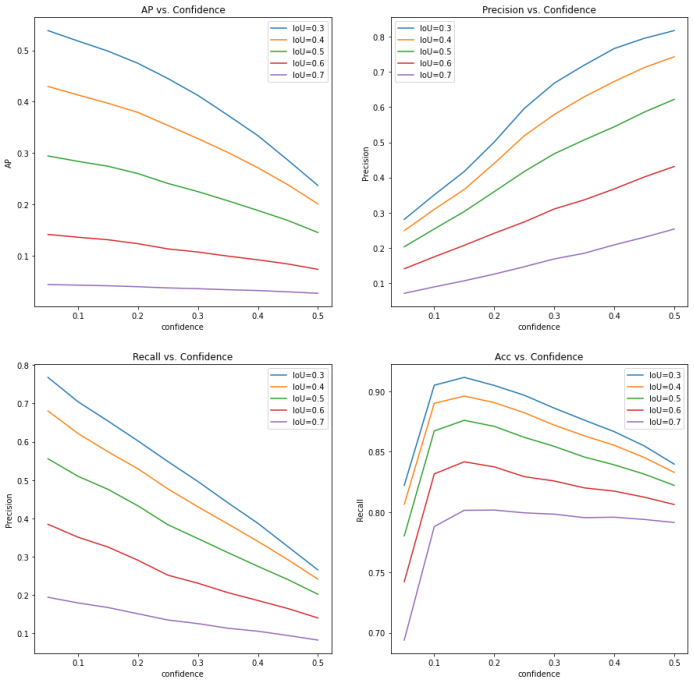


Figure of AP, Precision, Recall and Accuracy vs. Confidence (Epoch=12)