Pneumonia Detection Through Fine-Tuning FRCNN and FCOS Models

Colin Sullivan
Department of Computer Science
Stanford University
cms548@stanford.edu

Surya Prakash Sanapala
Department of Computer Science
Stanford University
svsunet@stanford.edu

Abstract

We examine and fine-tune one-stage and two-stage object detection models to detect pneumonia infections in chest radiographs. Our final model takes chest radiograph images as input and outputs bounding boxes indicating the location of the infection if pneumonia is present. Throughout our experiments, various fine-tuning strategies and advanced techniques were leveraged, including implementing Generalized Intersection over Union (GIoU) [13] loss on Faster Region-Based Convolutional Neural Network (FRCNN) [12] and customizing the loss function of the Fully Convolutional One-Stage Object Detection (FCOS) [15]. Data augmentation, hyper-parameter tuning, and Transfer Learning techniques were also implemented. Our best model resulted from fine-tuning FCOS with a weighted loss function.

1 Introduction

Pneumonia is a lung infection that accounts for over 15% of child deaths under the age of 5 worldwide. Identifying pneumonia typically requires highly skilled radiologists and is challenging as other lung conditions have similar visual symptoms. We explore generating a computer vision deep learning model that accurately identifies whether a chest scan indicates that a patient has pneumonia. If pneumonia is present, the system draws a bounding box around the location of the infection.

The input to the system is the raw radiograph chest-scan of a patient, which is in the medical imaging format DICOM. The chest-scan then goes through a fine-tuned model, either FRCNN [12] or FCOS [15], and outputs bounding box values indicating the location(s) of the infection if present. A key challenge in training this model is the limited availability of labeled data. Steps must be taken to ensure that the model can generalize well to unseen examples and not overfit to the training data. The task is also complicated by the class-imbalance problem, where the training data has far fewer pneumonia cases compared to normal cases.

2 Related work

For determining the best loss function for FRCNN, we leverage techniques described in Sapakova, Saya et al. 2022.[13] Here, Sapakova tests how various loss functions, including Intersection over Union (IoU), GIoU, Distance IoU (DIoU), and Complete IoU (CIoU) loss, affect image classification networks. GIoU loss returned the best results as it takes into account the overlapping area of the predicted box, the real box, and the non-overlapping area of the two. However, this analysis was tested on a binary image classification task that is different from our pneumonia detection model.

For data augmentation, we leverage techniques described in Gabruseva, Tatiana et al 2020.[7] Gabruseva implemented mild rotations, shift/scale/shear, horizontal flip, blur, and brightness adjustments

CS230: Deep Learning, Winter 2024, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

and found a 2% improvement in mean average precision when augmentations were used versus no augmentations. Gabruseva's dataset had a near equal number of positive and negative training images.

For hyperparameter tuning, we analyzed data from Akiba, Takuya et al. 2018.[4] Here, Akiba introduces the framework Optuna which includes a sampling and pruning algorithm that searches for optimal values for certain hyperparameters. While the results from Optuna are promising, there are concerns that it takes too long on large models and is better for smaller networks.

To evaluate our model against previously built computer vision pneumonia detectors, we reviewed Wu, Linghua et al. 2024.[16] Here, Wu et al. delivered a final model with an average recall of 73% and an average precision (AP) of 45.7% at 50% IoU and 7.5% at 70% IoU. Wu also fine-tuned an FRCNN model, but did not perform many additional techniques beyond data augmentation.

For Transfer Learning techniques, we leveraged concepts defined in Alapat, Daniel et al. 2022.[2] Alapat evaluated multiple computer vision models on their effectiveness to detect pneumonia in chest radiographs. Alapat found Resnet, Inception net, and other hybrid models to be most effective in detecting pneumonia. Resnet-50 performed better for the pneumonia detections than Resenet-101.

3 Dataset and Features

We use the RSNA Pneumonia Detection Challenge dataset [3], which consists of 26,685 chest radiographs, each with a binary label indicating whether the patient has pneumonia. If pneumonia is present, one or more sets of bounding box coordinates are included that indicate the location of the infection. Since positive pneumonia scans may have more than one bounding box, the dataset has a total of 30,227 labels, where each row includes the image ID, binary classification of whether pneumonia is present, and up to one bounding box. We have classified bounding boxes into 3 categories: small are 32x32 pixels, medium are between 32x32 to 96x96 pixels, and large are greater than 96 pixels.

Description	Values
Positive Pneumonia Radiographs	6,012
Negative Pneumonia Radiographs	20,672
Total Positive Labels	9,555 (due to multiple bounding boxes)
Total Negative Labels	20,672
Bounding box distribution (small / medium / large)	0 / 86 / 30,141

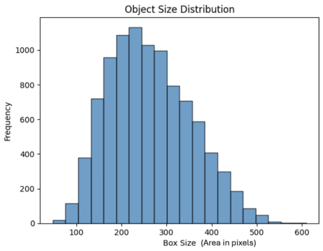


Figure 1: Data Distribution

Every image in the dataset is 1024x1024 pixels. We randomly sample 6000 positive and 6000 negative examples from the dataset, then shuffle and divide this into 80% train (9,600 examples), 10% validation (1,200 examples), and 10% test (1,200 examples). Additionally, on 10% of the training data we perform five data augmentation techniques which are described in Section 4.

4 Methods

When designing our pneumonia detection system, our approach focused on five key tasks: 1) fine-tuning FRCNN and customizing its anchor generation and loss function, 2) fine-tuning FCOS and modifying its loss function, 3) hyperparameter tuning, 4) data augmentation, and 5) metric evaluation. We describe each of these below.

Model Selection: We chose FRCNN model to train RSNA Pneumonia dataset as FRCNN is a two-stage object detection model and two-stage object detection models are known to perform better compared to one-stage object detection models like YOLO, RetinaNet, FCOS, etc. Below, we describe how we implemented a custom anchor generator and GIoU loss to improve baseline FRCNN performance. *See Figure 6 in Appendix for FRCNN architecture diagram.*

Custom Anchor Generator: The default anchor generator of RPN uses anchor sizes of (32, 64, 128, 256, 512). Since our dataset X-ray images are dominated by large objects (distribution provided in

Section 3), a custom anchor generator is created with anchor sizes of (128, 256, 512, 1024, 2048). Both size and aspect ratios of anchors are crucial for RPN in determining what specific objects it needs to focus on. A distribution of aspect ratios of all objects of the dataset was drawn and applied ‘Kmeans’ with K=5, which resulted in aspect ratios [1.00792981, 2.14550116, 0.48373241, 1.4092083, 0.71511279]. These values are approximated to (0.5, 0.75, 1.0, 1.5, 2.0) and replaced default aspect ratios (0.5, 1.0, 2.0).

Loss Functions: FRCNN computes 4 loss values. ’loss_objectness’, evaluates how well RPN classifies anchors as foreground vs background. ‘loss_rpn_box_reg’, measures how accurately RPN refines anchor boxes into region proposals. ‘loss_classifier’, measures how well ROI head classifies the proposals into correct object classes. ‘loss_box_reg’, measures how accurately ROI head refines bounding boxes for the proposals that contain boxes. Smooth L1 loss is used for the bounding box regression loss function. When predicted bounding boxes do not overlap with the ground truth bounding boxes, the Smooth L1 loss function does not give a meaningful gradient. GIoU [13] enhances the learning process by providing meaningful gradients even when bounding boxes do not overlap. We implemented the following GIoU function and replaced Smooth L1 loss function of ROI head with GIoU

$$GIoU(A,B) = IoU(A,B) - \frac{|C \setminus (A \cup B)|}{|C|}$$

Fully Convolutional One-Stage Object Detection (FCOS): FCOS is a One-stage Object detection model that implements Focal loss to handle class imbalance problems and also uses GIoU for bounding box regression. Unlike FRCNN, FCOS predicts bounding boxes without reliance on predefined anchors. This makes FCOS free of anchors, fewer hyperparameters, and easier to train. *Refer to Figure 7 in the Appendix for architecture*

Weighted loss computation: FCOS computes 3 loss functions. ‘Classification’ measures the loss of how well background vs foreground is classified. ‘Bbox_regression’, measures how well the predicted bounding box is aligned with the ground truth bounding box. ‘Bbox_ctrness’, measures the centeredness of the object, ensuring the model focuses on the central region of the object. Our earlier attempt of FCOS training gave equal weight for each of these losses. Instead of giving equal weights, we fine tuned weights for ‘bbox_regression’ and ‘bbox_ctrness’ and computed total loss accordingly.

losses = loss_cls + lambda_reg * loss_reg + lambda_cen * loss_centerness [17]

Hyperparameter Tuning and Optimizer: We use the python library Optuna for hyperparameter tuning.[4] Optuna utilizes a sampling and pruning algorithm that searches for optimal values within a user defined range. This framework helped us settle on a learning rate of 0.00001 and a learning rate scheduler with a step size of 10 and gamma of 0.1. We utilized the Adaptive Moment Estimation (Adam) optimizer and trained the system for 20 epochs, where each epoch consisted of 12,000 training images.

Data Augmentation: All training data is first normalized to speed convergence. We experimented with five data augmentation techniques including horizontal/vertical flips, Gaussian Blur, rotation, shear, and brightness/color adjustments. We utilized the Albumentations [5] python library to perform data augmentations as this library applies the transformations to the training images and the training labels when applicable. For example, flipping a training image horizontally also requires adjusting the training labels bounding boxes. Data augmentations were only applied to the training dataset, not to the validation or test datasets. *Refer to Figure 8 in the Appendix for data augmentation examples*

Evaluation: We calculate mAP at given Intersection over Union (IoU) thresholds. Intersection over union (IoU) is used to evaluate the accuracy of generated bounding boxes. We also calculate recall at 100 hits, which is the recall for the top 100 items returned. We experimented with two evaluation implementations. First, we implemented our own functions where model predictions were filtered by a score threshold of 0.83, then IoU was calculated, and finally non-max suppression filtered outputs with a threshold of 0.5. However, we ultimately decided to use the python library TorchMetrics [10] to evaluate our model as it ran more efficiently.

5 Experiments/Results/Discussion

Baseline: Our baseline model leverages FRCNN as an object detection framework and utilize Resnet-50 for feature extraction. We freeze the backbone, and finetune both the Region Proposal Network

(RPN) and Region of Interest (ROI) layers. The model is trained with a mini-batch size of 6 (this fit optimally on the NVIDIA T4 GPU we used) using cross-entropy loss for 5 epochs on all 21,349 chest radiographs. The model's hyperparameters are as follows: learning rate = 0.0001, momentum = 0.9, and weight decay = 0.0001. Results for the baseline model are shown in Table 1.

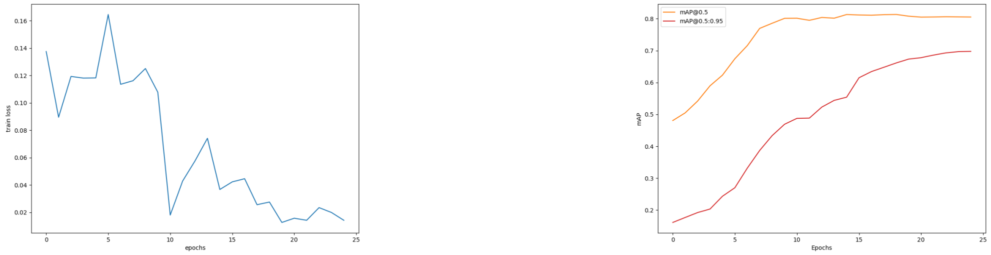
Results: For all models except the baseline (described above), the model was trained with Adam optimizer, a learning rate of 0.00001, and a learning rate scheduler with a step size of 10 and gamma of 0.1. We trained the system for 20 epochs, where each epoch consisted of 12,000 training images (6,000 negative and 6,000 positive). Mini-batch size was 6. Below are results of our four models:

Model	**mAP at 50%**	**mAP at 50%:95%**	**Recall w/100 Hits**
1) Baseline	46.3%	16.1%	39.4%
2) FRCNN + GIoU	**80.52%**	69.72%	74.21%
3) FCOS + Weighted Loss (WL)	80.21%	**71.25%**	**78.24%**
4) FCOS + WL + Data Augmentation	58.05%	25.0%	51.56%

Table 1: Model Results

Analysis of Baseline: Possible reasons for the baseline model's low performance include: 1) RSNA dataset has a significant class imbalance issue where there are few positive samples of pneumonia compared to many negatives. 2) Chest X-rays with lesions may not have well-defined boundaries or consistent aspect ratios. 3) The default Smooth L1 loss function may not be optimized well for ROI bounding box alignment. 4)The model needs more examples or augmented data.

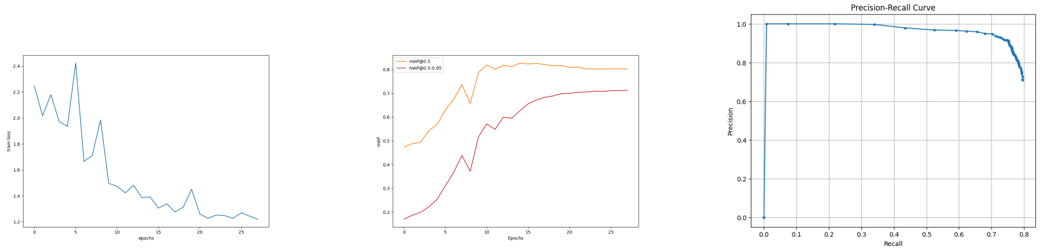
FRCNN with Custom Anchor Generator and GIoU: Custom anchor generator and GIoU loss functions improved the performance of the base FRCNN model. We believe GIoU improved performance as Smooth L1 loss not provide meaningful gradients for when predicted bounding boxes do not overlap with ground truth boxes, an issue GIoU solves.



(a) Train Loss

(b) mean Average Precision

Figure 2: Training, evaluation performance of FRCNN



(a) Training Loss

(b) Mean Average Precision

(c) Precision Recall curve

Figure 3: Training, Evaluation and test performance of FCOS

Analysis of our Best Model, FCOS and Weighted Loss: Our best model featured FCOS and weighted loss. For $lambda_{reg}$ and $lambda_{cen}$, we experimented with values 1.5 and 2.0 and noticed better performance with $lambda_{reg} = 2.0$ and $lambda_{cen} = 2.0$. One key observation from training

FRCNN is it is sensitive to anchor settings and requires careful tuning to perform optimally. FRCNN implementation is not addressing the class imbalance issue though it allows replacing the existing loss function with a function that can handle class imbalances. We believe FCOS performed equally as well as FRCNN as it addresses the class imbalance problem and uses DIoU for bounding box regression. Figure 3 shows the mAP curve over epochs along with train loss.

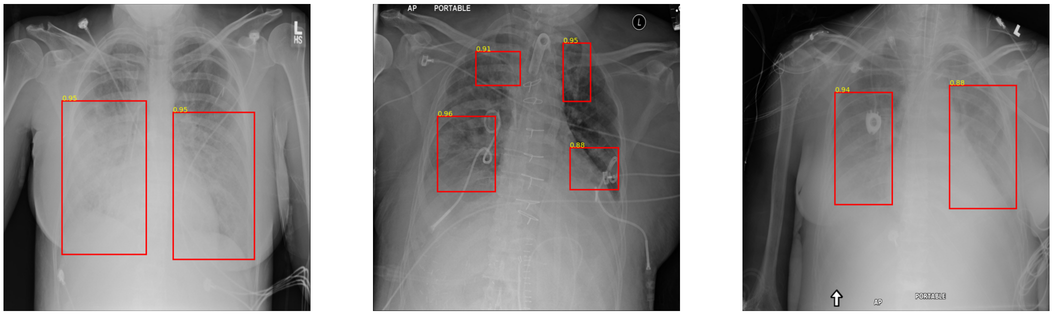


Figure 4: Predictions with bounding boxes

Data Augmentation: Data augmentation surprisingly hurt model performance. We believe there are two reasons for this. First, some of our applied augmentations were too severe. For example, radiographs were rotated by 15° which is too high as the position patients during a radiograph should rotate by approximately 6°.[7] Second, we applied some augmentations that were not relevant such as vertical flips. Horizontal flips are valid as the lungs are relatively symmetric, however a vertical flip (or upside down radiograph) is not something the model would see in the test set. Below is the loss curve, recall, and precision when training the FCOS model with data augmentations. The loss curve keeps decreasing while the precision and recall values also decrease, indicating that the model may be overfitting to the training data.

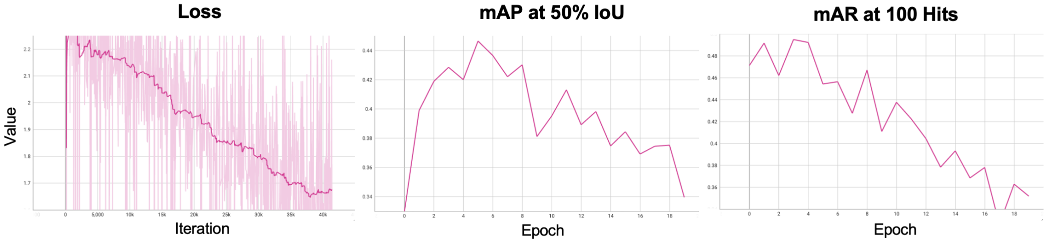


Figure 5: Loss, Precision, and Recall During Data Augmentation Training.

6 Conclusion/Future Work

We have fine-tuned the Faster R-CNN and FCOS model to detect pneumonia infections in patient chest-scans. Our best model resulted from fine-tuning FCOS with a weighted loss function.

For future work, We could apply weighted loss computation of FRCNN training losses, and fine-tune lambda values to get better results. We will implement multi-GPU training and acquire additional GPU resources to improve iteration speed. We could explore augmentations suitable to X-ray images that help enhance model performance. We could also look into fine-tuning the DEtection TRansformer or DETR model, first proposed by *in End-to-End Object Detection with Transformers* (Carion, Nicolas et al. 2020).[6] The DETR model uses a set-based global loss function that forces unique predictions through a transformer encoder-decoder architecture and bipartite matching. Initial results from DETR show improved performance for object detection over FRCNN and FCOS models.

7 Code

Project code can be found on GitHub here: `https://github.com/ColinSulli/CS230_Project/tree/main`

8 Contributions

Colin Sullivan: Worked on data augmentation, data initialization, evaluation functions, training loop order, setting up Virtual Machines for training, analysis, TensborBoard / visualizations, data loading, related work research.

Surya Prakash Sanapala: Worked on fine-tuning of FRCNN and FCOS models. Implemented custom anchor generator for fine-tuning. Implemented GIoU loss function for FRCNN ROI Head. Implemented and fine-tuned logic for 'weighted loss computation' for FCOS model. Implemented computation and visualization of evaluation metrics using Torch metrics. Performed Hyperparameter tuning using optuna.

References

[1] Adrian Rosebrock. "Intersection over Union (IoU) for Object Detection," https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/, 2016-11-7

[2] Alapat DJ, Menon MV, Ashok S. A Review on Detection of Pneumonia in Chest X-ray Images Using Neural Networks. J Biomed Phys Eng. 2022 Dec 1;12(6):551-558. doi: 10.31661/jbpe.v0i0.2202-1461. PMID: 36569568; PMCID: PMC9759647.

[3] Anouk Stein, MD, Carol Wu, Chris Carr, George Shih, Jamie Dulkowski, kalpathy, Leon Chen, Luciano Prevedello, Marc Kohli, MD, Mark McDonald, Peter, Phil Culliton, Safwan Halabi MD, Tian Xia. (2018). RSNA Pneumonia Detection Challenge. Kaggle. https://kaggle.com/competitions/rsna-pneumonia-detection-challenge

[4] Akiba, Takuya, et al. "Optuna: A Next-Generation Hyperparameter Optimization Framework." Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery data mining. 2019.

[5] Buslaev, Alexander, et al. "Albumentations: fast and flexible image augmentations." Information 11.2 (2020): 125.

[6] Carion, Nicolas, et al. "End-to-end object detection with transformers." European conference on computer vision. Cham: Springer International Publishing, 2020.

[7] Gabruseva, Tatiana, Dmytro Poplavskiy, and Alexandr Kalinin. "Deep learning for automatic pneumonia detection." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020.

[8] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. 2019. CheXpert: a large chest radiograph dataset with uncertainty labels and expert comparison. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI'19/IAAI'19/EAAI'19). AAAI Press, Article 73, 590–597. https://doi.org/10.1609/aaai.v33i01.3301590

[9] Nayem, J., Hasan, S. S., Amina, N., Das, B., Ali, M. S., Ahsan, M. M., Raman, S. (2023). Few-Shot Learning for Medical Imaging: A Comparative Analysis of Methodologies and Formal Mathematical Framework. arXiv [Eess.IV]. Retrieved from http://arxiv.org/abs/2305.04401

[10] Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, William Falcon, "TorchMetrics - Measuring Reproducibility in PyTorch," https://www.pytorchlightning.ai, 2022-02-11

[11] Paszke, Adam, et al. "Pytorch: An imperative style, high-performance deep learning library." Advances in neural information processing systems 32 (2019).

[12] Ren, Shaoqing. "Faster r-cnn: Towards real-time object detection with region proposal networks." arXiv preprint arXiv:1506.01497 (2015).

[13] Sapakova, Saya, and Yelidana Yilibule. "Deep learning-based face mask detection using YOLOV5 model." Scientific Journal of Astana IT University (2022): 5-13.

[14] Tian, Zhi, et al. "Fully convolutional one-stage 3d object detection on lidar range images." Advances in Neural Information Processing Systems 35 (2022): 34899-34911.

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollar. Focal Loss for Dense Object Detection. Facebook AI Research (FAIR). https://arxiv.org/pdf/1708.02002.

[16] Wu, Linghua et al. “Pneumonia detection based on RSNA dataset and anchor-free deep learning detector.” Scientific reports vol. 14,1 1929. 22 Jan. 2024, doi:10.1038/s41598-024-52156-7 [17] Chunhua Shen et al. "Fully Convolutional One-Stage Object Detection". https://arxiv.org/pdf/1904.01355

Appendix

FCRNN Architecture:

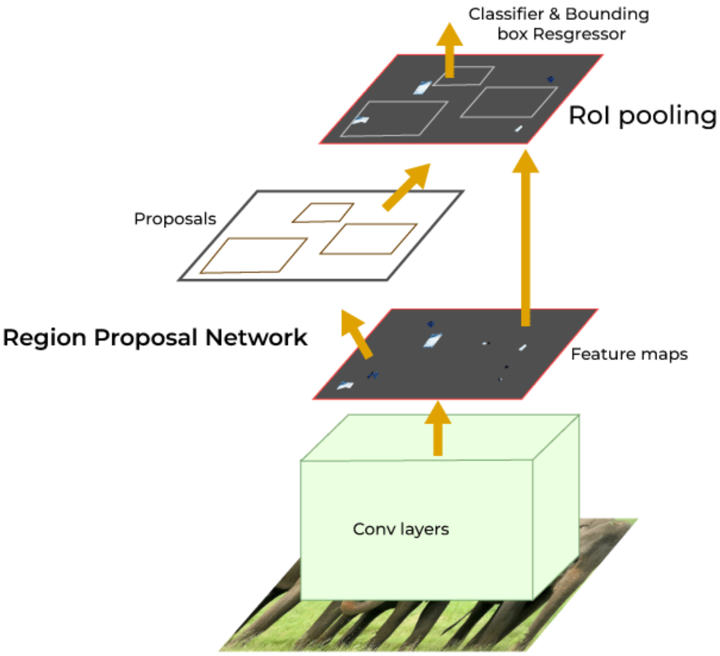


Figure 6: FRCNN architecture showing the pipeline of Convolution network to extract features, RPN for first stage proposals and ROI for Object classification. [12]

FCOS Architecture:

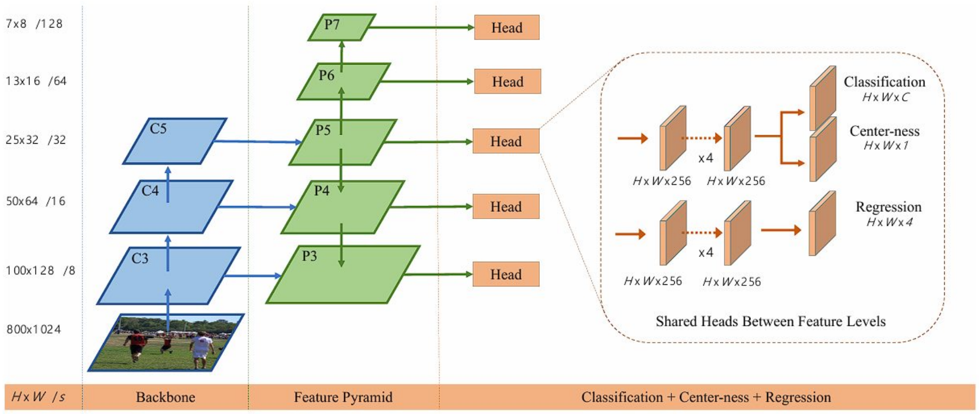


Figure 7: FCOS network shows feature maps from backbone fed to FPN and feature levels of FPN used for final prediction

Data Augmentation Examples:

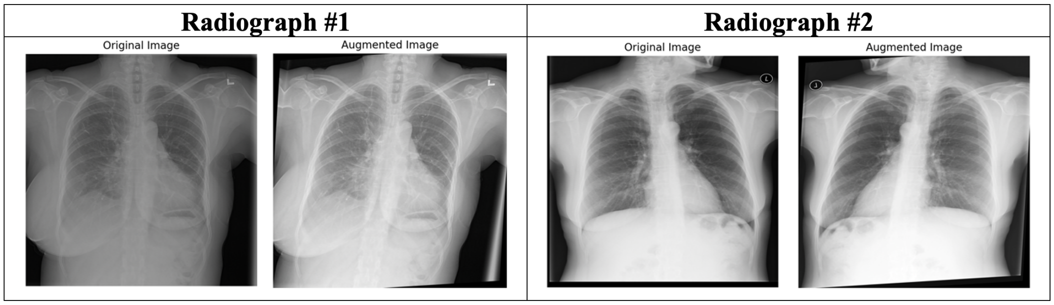


Figure 8: Radiograph #1 shows how the brightness and shear of the original image was augmented. Radiograph #2 shows how the original image was flipped horizontally and sheared.