Final Project: RSNA Pneumonia Detection

Team 19
Teng-Te Yu ID:310551054
Zheng-Yan Chen ID:310552006
Yi-Hsin Chen ID:409551027

Project repository

https://github.com/samuelyutt/Selected-Topics-in-Visual-Recognition-using-Deep-Learning-course/tree/final-project/FinalProject

Abstract

Pneumonia has been one of the top ten causes of death in both the United States and Taiwan for a long time. This project addresses pneumonia detection by chest radiographs (CXRs). In this project, we experiment on two data pre-processing techniques, data cleaning and contrast limited adaptive histogram equalization (CLAHE), and two widely used deep learning object detection methods, YOLOv5 and Faster R-CNN. Experimental results show that Faster R-CNN has better performance and our best model get 0.14062 mAP in Kaggle RSNA Pneumonia Detection Challenge. Ablation studies show that our pre-processing and modification in YOLOv5 do improve the performance.

1. Introduction

Pneumonia is on the list of top 10 causes of death in both United States and Taiwan. The diagnosis of pneumonia requires the analysis of the patient's chest radiographs (CXRs), clinical history, vital signs and laboratory tests by an experienced specialist. However, the diagnosis on CXRs is complicated and reading a large number of images is time-consuming. Since deep learning has performed very well in many fields in the last few years, Radiological Society of North America (RSNA) held a pneumonia detection competition on Kaggle[1] to seek the assistance of deep learning techniques in accelerating pneumonia diagnosis. In this competition, we need to build a model to detect lung opacities on CXRs

Currently, there are two main categories of object detection methods in deep learning. One is one-stage detectors such as YOLO family (e.g. YOLOv1[2], YOLOv2[3], YOLOv3[4], YOLOv4[5], YOLOv5[6]) and the other is two-stage detectors such as R-CNN family (e.g. R-CNN[7], Fast R-CNN[8], Faster R-CNN[9]). The former are proposal-free methods. Without finding proposals, these methods directly using a CNN model to predict the positions and categories of different objects. In contrast, the latter are proposal-based methods, they first generate region proposals and then do classification and regression on each proposal. Since two-stage methods have one more step of generating region proposals, they are slower but more accurate than one-stage methods. In medical field, the accuracy is much more important than speed. Therefore, two-stage methods seem to be more appropriate in this field.

To balance the trade-off between accuracy and speed, both YOLO family and R-CNN family do continuously propose new methods. However, all these approaches are evaluated on real-world datasets which contain various classes and complex background. Those techniques may not all fit for this project data since there is only one target class, pneumonia, and the background are monotonous.

In this project, we dig into both YOLOv5[6], the latest method in YOLO family, and Faster R-CNN[9], the most widely used method in R-CNN family, designs and conduct experiments to compare these two methods. Furthermore, we also do ablation studies on our data pre-processing techniques, data cleaning, contrast limited adaptive histogram equalization (CLAHE), resizing.

2. Related Work

In this project, we try two different object detection models, one-stage object detector YOLOv5[6] and two-stage object detector Faster R-CNN[9]. In this section, we will briefly introduce these two models.

YOLOv5

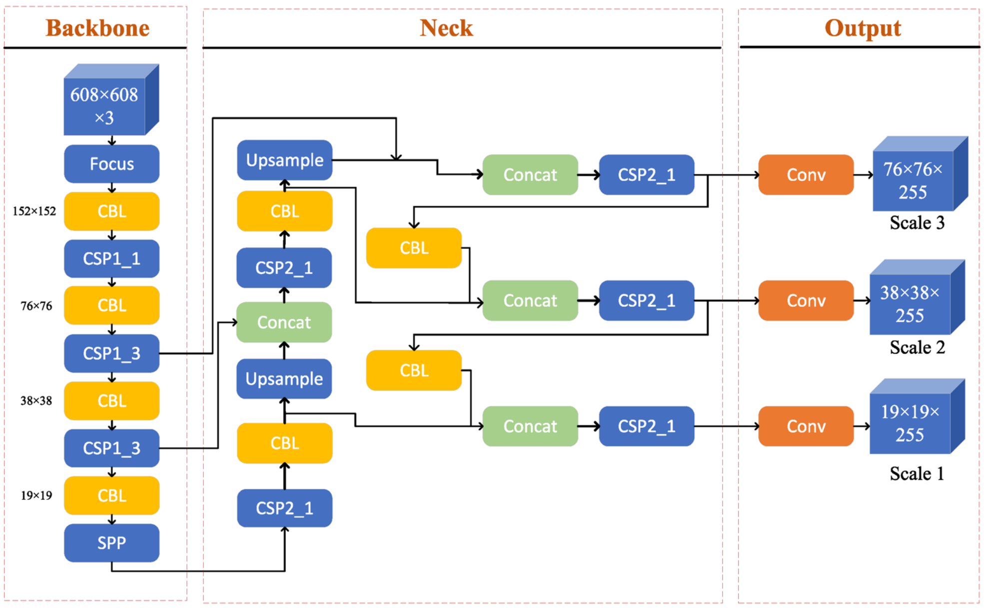


Fig.1 Framework of YOLOv5 (image is from [10]).

YOLOv5 is a one-stage object detector, which formulates object detection problem as a regression problem to spatially predict bounding boxes and class probabilities. Comparing with two-stage detector, one-stage detector only feedforwards the input image once, so the detection speed is much faster than two-stage detector. Since the input is only feedforwards once, the model will extract global information of the entire image and make the model is more robust to various background.

Fig.1 shows the framework of YOLOv5. As shown in Fig.1, we can separate YOLOv5 into backbone, neck and output, where neck are layers between backbone and head collecting feature maps from different stages. Comparing with previous version of YOLO, YOLOv5 propose several techniques to input, backbone, neck and output respectively to improve training speed, the detection accuracy and the inference speed.

Faster R-CNN

Faster R-CNN is a two-stage object detector. Comparing with one-stage object detector, two-stage detector contains an additional step to generate proposal regions.

As shown in Fig.2, Faster R-CNN consists of two stages. The first stage, called region proposal network (RPN), proposes candidate object bounding boxes. The second stage extracts feature of each candidate box by using RoIPool and then performs classification and bounding-box regression for each candidate box.

Comparing with the previous version Fast R-CNN, Faster R-CNN propose a region proposal network replacing selective search of region of interest in Fast R-CNN. As their experiments show, by using region proposal network, Faster R-CNN can get the same mAP in Pascal VOC dataset with 10 times faster than Fast R-CNN.

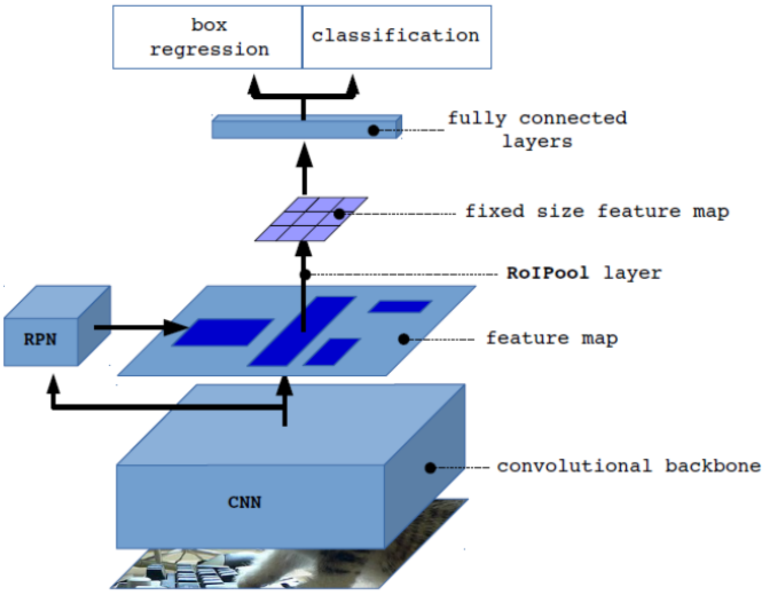


Fig.2 Framework of Faster R-CNN.

3. Methodology

This section presents our methodologies to address pneumonia detection. Fig.3 shows the training pipeline of our method. The pipeline containing two steps, data pre-processing and detection, with the detailed operations given in the following subsections.

3.1 Data Pre-processing

As shown in Fig.3, our data pre-processing includes three steps, data cleaning, contrast limited adaptive histogram equalization (CLAHE) and data transformation.

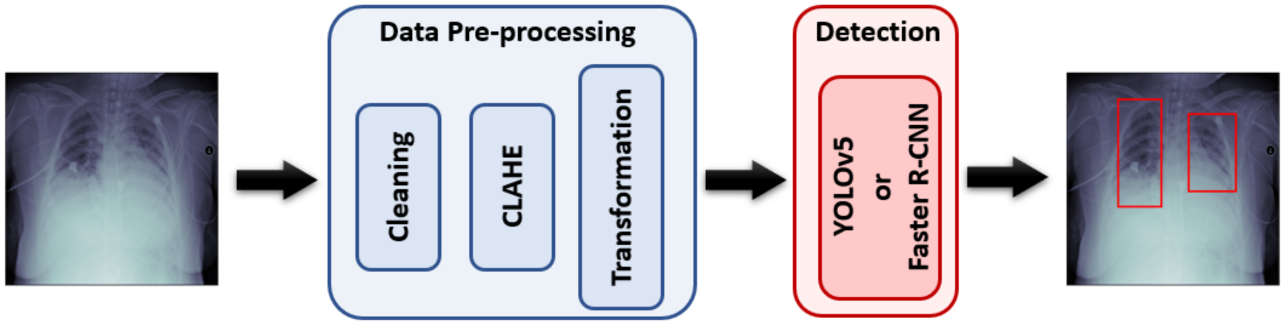


Fig.3 Pipeline of our method.

<u>Data Cleaning</u>

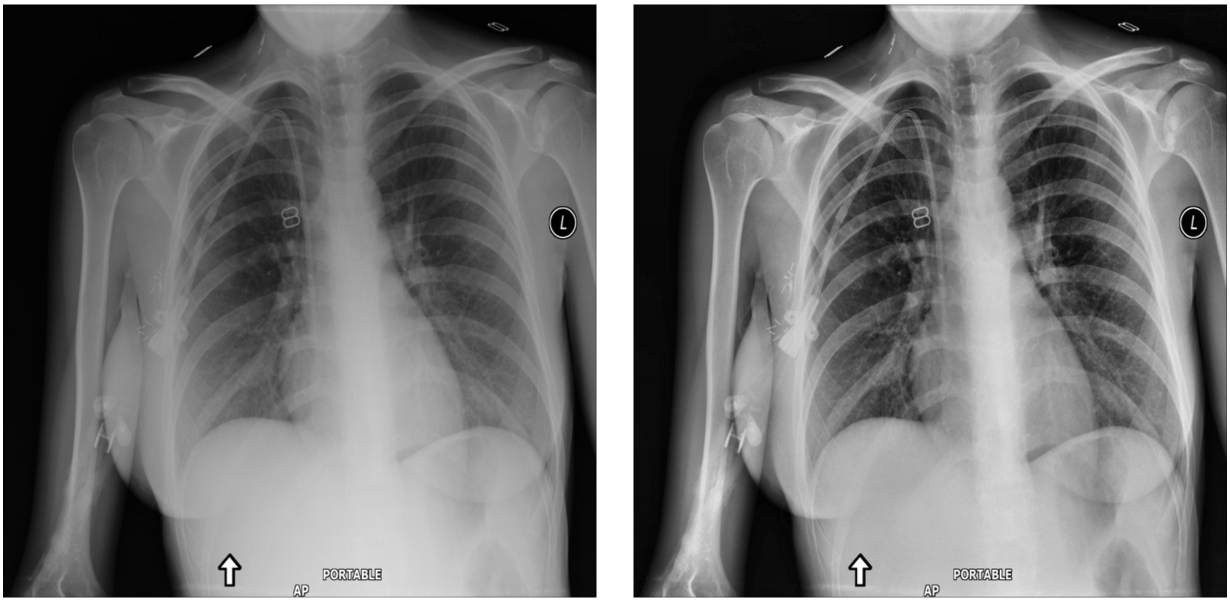
The RSNA pneumonia training dataset contains 26,684 cases with only 6,012 pneumonia images (~22.03%) and the remaining are normal images or images that has no turbidity in the lungs. Since in object detection task, images with annotation are the most helpful data for training and the images without annotation or abnormal data may be harmful for training if using in incorrectly way, we only select those 6,012 images with pneumonia as training data. These images contain 1~4 annotated regions, where the distribution is shown in Table.1. Our experimental result shows that with this data cleaning process, the detection performance does improve.

# region	1	2	3	4
# data	2613	3257	129	13

Table.1 Distribution of lesion areas in used training data.

<u>Contrast Limited Adaptive Histogram Equalization (CLAHE)</u>

In X-ray images, normal lungs do not absorb X-rays, so they appear black, while the pneumonia area is shaded with gray dashed lines or cloudy. Therefore, enhance contrast in images may help the detection. In this project, we use contrast limited adaptive histogram equalization (CLAHE), where CLAHE is an image processing technique used to improve the contrast in images and it is widely used in medical image dataset. Fig.4 shows before and after of processing CLAHE. Our experimental result shows that with CLAHE, the detection performance does improve.



(a) (b)

Fig.4 Illustration of image pre-processing by CLAHE. (a) is the input image. (b) is the image after contrast enhancement by CLAHE.

<u>Data Transformation</u>

Since the original data size is 1024×1024 and it will need a lot of computational resources for training, we resize the input images as 500×500. Our experimental result shows that the performance of using 500×500 and 1024×1024 are quite close but saving more computational resources when resizing input to small size.

Moreover, since there are only 6,012 images with 7,666 annotation in total and the number of data are not enough to train a deep learning model, we do some data augmentations. Detail operations are listed in Sec. 4.1.

3.2 Detection

In this project, we try two different popular object detection methods, YOLOv5 and Faster R-CNN. The details are described in the following subsection.

<u>YOLOv5</u>

As we mentioned in related work, YOLOv5 is a one-stage detector. Comparing with previous version of YOLO, YOLOv5 contains several techniques to improve both the detection accuracy and the processing speed. In specific, these technologies can be categorized into 4 categories, respectively for the 4 parts of the model (input, backbone, neck and prediction). However, we think that not all of the components can be beneficial for this project.

In YOLOv4 and YOLOv5, they use Mosaic as data augmentation. Comparing with CutMix which mixes only 2 input images, Mosaic mixes 4 training images. This not only increases the variety of dataset make the network more robust, especially for detecting small objects, but also reduces the need for GPU. However, there are two reasons that we think these kinds of augmentation techniques is not suitable for the dataset of this project. First, Mosaic can increase the detection of small objects, but the targets of this project's data are not small object. Second, we think that to detect pneumonia, the model should first find the location of lungs and then detect the opacities. However, both CutMix and Mosaic will destroy the integrity of the images and make if difficult for the model to locate the lungs. Fig.5 demonstrates the usage of Mosaic in COCO dataset and the dataset of this project.

(a) (b)

Fig.5 Illustrations of images pre-processed with Mosaic. (a) are the examples of COCO dataset. (b) are the examples of this project's dataset.

Because of the aforementioned reasons, we unplug Mosaic in our YOLOv5. In our experiments, we use YOLOv5X as our model with CSPDarknet53 as the backbone and SPP+FPN+PAN as neck. Our experimental result shows that without Mosaic, the detection performance does improve.

Faster R-CNN

As we mentioned in related work, Faster R-CNN is a two-stage detector, which has additional steps to generate region proposals. In our experiments, we find that compare with using YOLOv5, directly using the original Faster R-CNN with ResNet50 as the backbone already has better performance.

Even though Faster R-CNN perform better than YOLOv5 in this dataset and it is a widely used model in object detection, there is still room for improvement. There are two modifications that we think can improve the performance. First, using K-Means++ in YOLOv3 to obtain the initial anchor box size can speed up the convergence of training process and increase the detection performance. Second, using feature pyramid network to extract feature and feed these multi-scale features into region proposal network. Since original Faster R-CNN only use one scale feature, the features may not be comprehensive enough for the resolution of objects of different sizes in the image even though there are multiple anchor prototypes for region proposal network. However, due to lack of time, we list these two modifications as future work.

4. Experiments

4.1 Setup

Dataset

In this project, we use the dataset provided by Kaggle RSNA Pneumonia Detection Challenge. It contains 26,684 training CXRs and 3,000 test CXRs, where each training case contains 0~4 target regions. As we mentioned in Sec. 3.1, during training, we will do data pre-processing (data cleaning, CLAHE and data transformation). At test time, we also do data pre-processing CLAHE and transformation.

Evaluation Metric

In this project, we use the Mean Average Precision (mAP) as evaluation metric. The higher mAP represents better performance.

Training Details

In this project, we use two different object model, YOLOv5 and Faster R-CNN. The hyper-parameter and data transformation we use for each model are listed as following:

YOLOv5

Hyperparameter

Hyperparameter	Value
Learning rate	0.01
Momentum	0.937
Weight decay	0.0005

Table.2 Hyperparameters of YOLOv5

Data Transformation

Transformation	Probability	Arguments
Resize	1.0	1024,1024
Flip Horizontal	0.5	
Flip Vertical	0.5	
Mosaic	0.0	

Table.3 Data transformation of YOLOv5

Faster R-CNN

Hyperparameter

Hyperparameter	Value
Learning rate	0.001
Momentum	0.9
Weight decay	0.0005

Table.4 Hyperparameters of Faster R-CNN

Data Transformation

Transformation	Probability	Arguments
Resize	1.0	500, 500
Flip	0.5	
Random rotate 90 degrees	0.5	
Motion blur	0.2	
Median blur	0.3	blur_limit=3
Blur	0.1	blur_limit=3
ToTensorV2	1.0	

Table.5 Data transformation of Faster R-CNN

4.2 Experimental Results

Table.6 reports the mAP results of different models on test dataset. The results show that Baseline model performs the best and YOLOv5 performs the worst. The mAP gaps of models are quite huge. Table.7 shows some qualitative detection results of YOLOv5 and Faster R-CNN on both training and test set. We can observe that faster R-CNN does detect much better than YOLOv5.

Model	mAP
Baseline	0.20125
YOLOv5	0.05240
Faster R-CNN	0.14062

Table.6 mAP for different models.

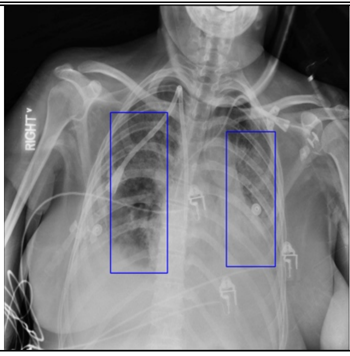
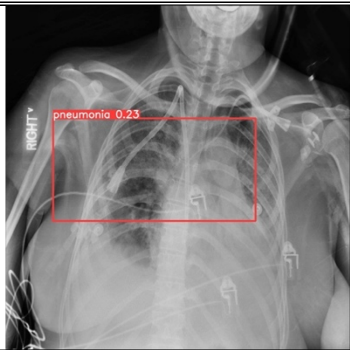
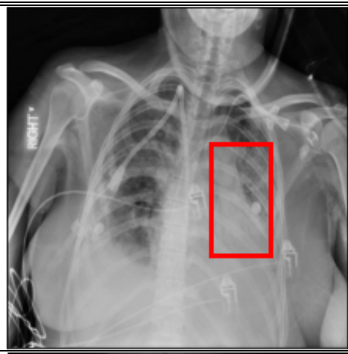
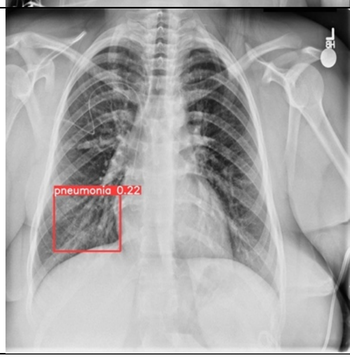
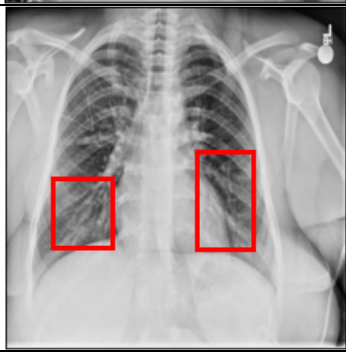
Type	Ground truth	YOLOv5	Faster R-CNN
Training			
Test			

Table.7 Demonstrations of models' outputs

4.3 Ablation Study

In this project, we use several data pre-processing techniques and we also claim that Mosaic is not good for this dataset in Sec. 3.2. Therefore, in this section, we do several ablation studies to evaluate the effectiveness. The rest of the ablation studies were evaluated on the Faster R-CNN, except for the Mosaic experiment which was performed on YOLOv5.

(1) w/o Data cleaning v.s. w/ Data Cleaning

Table. 8 shows that mAP does increase when we do data cleaning before training i.e. only use the data with annotations.

Setting	mAP
CLAHE + rotate90 + resize	0.12755
CLAHE + cleaning + rotate90 + resize	0.14062

Table.8 Ablation study for data cleaning.

(2) w/o CLAHE v.s. w/ CLAHE

Table. 9 shows that mAP does increase when we use CLAHE to enhance the contrast in each image before training.

Setting	mAP
cleaning + rotate90 + resize	0.13948
CLAHE + cleaning + rotate90 + resize	0.14062

Table.9 Ablation study for CLAHE.

(3) w/o Resizing v.s. w/ Resizing

Table. 10 shows that mAP is comparable or even better when we resize the original 1024×1024 images to 500×500. Since the performance does not drop when we resize the input to smaller size and using smaller size of input can save the usage of computational resources, we resize the inputs to smaller size, 500×500 in our final model.

Setting	mAP
CLAHE + rotate90	0.12240
CLAHE + rotate90 + resize	0.12755

Table.10 Ablation study for resizing.

(4) w/o Rotation v.s. w/ Rotation

Table. 11 shows that mAP does increase when we randomly rotate input with 90 degree as data augmentation technique.

Setting	mAP
cleaning + resize	0.09697
cleaning + rotate90 + resize	0.13948

Table.11 Ablation study for randomly rotation input with 90 degree.

(5) w/o Mosaic v.s. w/ Mosaic

Table. 12 shows that mAP does increase when we remove Mosaic in YOLOv5.

Setting	mAP
w/o Mosaic	0.05240
w/ Mosaic	0.01470

Table.12 Ablation study for Mosaic in YOLOv5.

4.4 Discussion

Overfitting or Inappropriate Hyper-parameter Issue

Fig.6 shows the training epoch and mAP of test set in our different Faster R-CNN models. We can observe that as the training epoch increases, the mAP of the test set will first increase and then stay flat or even drop in our different settings. We think that this may due to the overfitting to training data or our learning rate schedule is inappropriate.

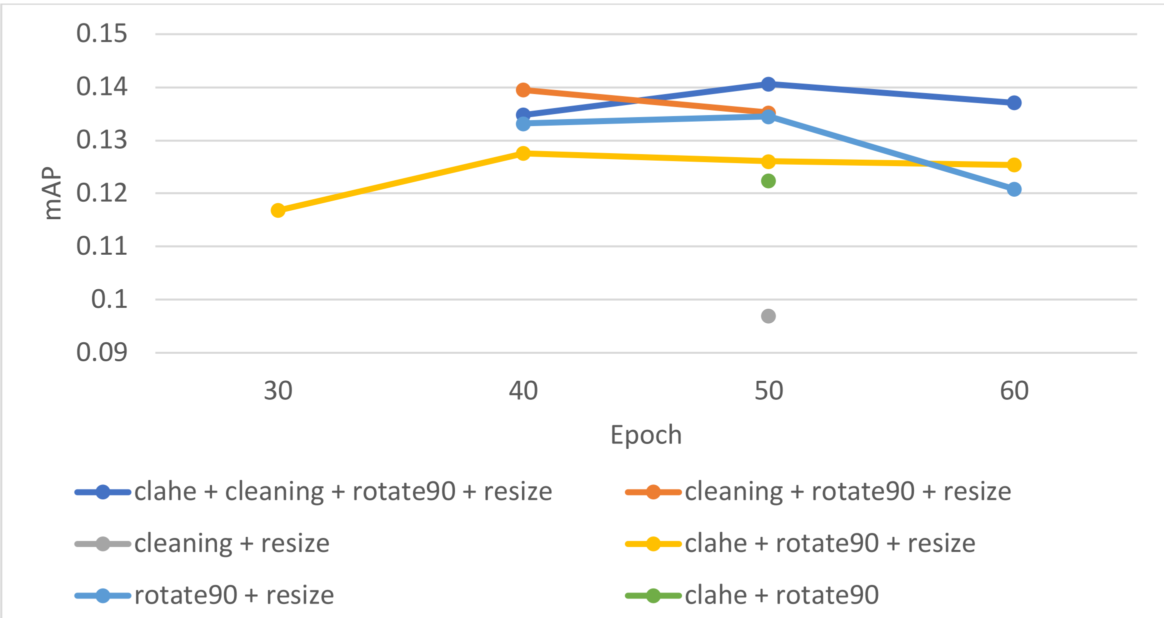


Fig.6 Training epoch-test mAP curves of different Faster R-CNN models.

5. Conclusion

In this project, we try to use two different widely used object detection models, YOLOv5 and Faster R-CNN, with some data pre-processing and data augmentation techniques to detect pneumonia from CXRs. Our experimental results show that the original Faster-RCNN with some data pre-processing and data augmentation techniques can get 0.14062 mAP in Kaggle RSNA Pneumonia Detection Challenge. Extensive ablation studies show that our pre-processing and modification in YOLOv5 do improve the performance. Although Faster R-CNN performs better than YOLOv5 in our experiments, there sill has a room for improvement. As we mentioned in Sec. 3.2, using K-Means++ in YOLOv3 to obtain the initial anchor box size and using more robust feature extractor may help to speed up the convergence of training process or increase the detection performance. However, due to lack of time, we list these two modifications as future work.

Team Member Contribution

Tasks	Contributors
Literature survey	310551054 (33%), 310552006 (33%), 409551027 (33%)
Approach design	310551054 (33%), 310552006 (33%), 409551027 (33%)
Approach implementation (experiment)	310551054 (40%), 310552006 (40%), 409551027 (20%)
Report writing	310551054 (30%), 310552006 (30%), 409551027 (40%)
Slide making and oral presentation	310551054 (30%), 310552006 (30%), 409551027 (40%)

Reference

[1] https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/overview

[2] Redmon, Joseph, et al., "You only look once: Unified, real-time object detection," Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2016.

[3] Redmon, Joseph, and Ali Farhadi, "YOLO9000: better, faster, stronger," Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2017.

[4] Redmon, J., and Farhadi, A, “Yolov3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.

[5] Bochkovskiy et al. "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[6] https://github.com/ultralytics/yolov5

[7] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation," Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2014.

[8] Girshick and Ross, "Fast r-cnn," Proceedings of the IEEE international conference on computer vision, (CVPR) 2015.

[9] S. Ren et al. “Faster R-CNN: Towards real-time object detection with region proposal networks,” Advances in neural information processing systems (NIPS), 2015.

[10] Zhu, Linlin, et al. "Improving YOLOv5 with Attention Mechanism for Detecting Boulders from Planetary Images." Remote Sensing, 2021, 13.18: 3776.