Liver Lesion Detection: Image Preprocessing for Faster R-CNN

Keyword: image cropping, data augmentation, feature pyramid, Faster R-CNN

Zheng Chuanyang(17251311), Gong Yikai(17251567)

Duty List:

Zheng Chuanyang: Discussing ideas, coding, doing the experiment and modifying the report

Gong Yikai: Discussing ideas, modifying code and writing report

Introduction

Background

The liver is the largest organ in humans' bodies, which carries lots of functions [1]. Nowadays, its health issues are getting increasingly more attention. As a group of abnormal cells [2], the liver lesion is a signal of the liver healthy problem, and it may lead to a more serious problem such as tumors and cancer. Early detection and early treatment are the keys to patient recovery. In this article, using Faster R-CNN as a baseline, we are designing an end-to-end, efficient, and accurate algorithm to detect liver lesions, which improve the performance 7.36%.

Preliminary Studies

Related work

Object Detection. Faster R-CNN is one of the most efficient methods for object detection and classification since 2015. It introduces a Region Proposal Network (RPN) which enables almost cost-free region proposals [3]. Faster R-CNN is designed for RGB images, which needs three channels.

Feature Pyramid. Feature Pyramid [5], or multi-scale, the method tries to use different scales information, compared traditional one-scale model. The main method of feature pyramid is using different kernel sizes. Different kernel sizes give different receptive fields and give different feature sizes. Feature Pyramid uses different kernel sizes to get different feature sizes and computer combined results by giving these features different weights. Compared to the original simple feature pyramid method[5], several research papers are discussing a new feature pyramid method [4][6][7].

Motivation

As mentioned before, the input of Faster R-CNN needs three channels. However, in this task, only grey images are given. That is to say, we have only one channel. There are three ways to solve this problem. First, input the same data into three channels by duplicate the gray images. Second, modify the Faster R-CNN code to make it compatible with one channel condition. Third, preprocess grey image by image convolution. The first and second ways make the remain channels underutilized. And the third method only uses some of the images to generate the next one, which may lead to data loss.

Preliminary experiments.

The first one is the temporary baseline. No image pre-cropping, data augmentation, or feature pyramid. And same information was inputted into three channels. Then faster r-CNN was adopted to train. The accuracy is 58%. It takes four hours to train, and it takes one minute to test.

In the second experiment, we just add simple convolution methods to test whether our direction is correct. **Our simple model (not the final model)** is the following.

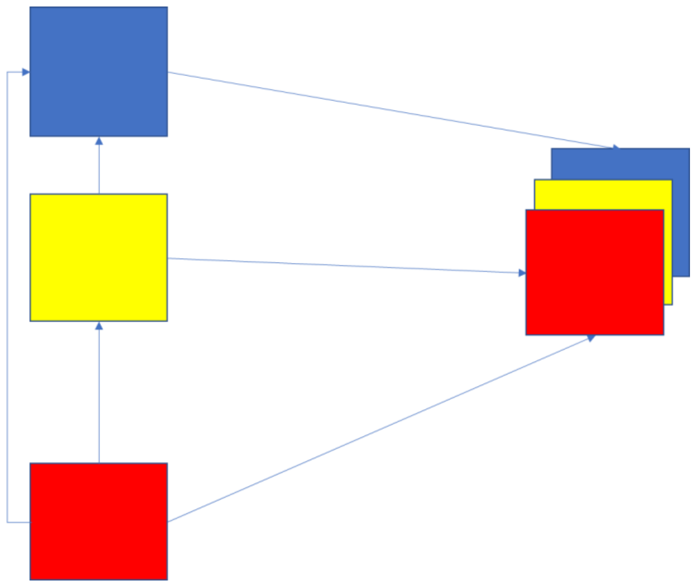


Figure 1. Input is a red image. To get yellow image (1 channel), convolute red image (1 channel) with kernel size=3, stride=1, padding=1. To get a blue image (1 channel), convolute red image (1 channel), convolute red image, and yellow image with kernel size=3, stride=1, padding=1. Then, stack red image, yellow image, and blue image

Method	IOU Accuracy
Temporary Baseline (duplicate 2 other images)	58%
Temporary Baseline 2 (original image + lapalican method processed image + max pool image)	54.3%
Simple Model	58.4%

Table 1. Comparison with baselines and simple model

This result shows that our method may improve accuracy, so further study is to research for an advanced way to generate the three channels.

Project Objective

To find the best way to detect liver lesions. Meanwhile to explore the performance of the feature pyramid. And to observe how other preprocessing methods influence the result.

Algorithm and Rationale

Image pre-cropping

It is easy to observe that there is a large black area containing no useful information surrounding the main part. The goal of pre-cropping is to cut a rectangle of the abdominal section, which is equivalent to enlarging and focusing on the useful area. This algorithm repeats four times from four directions. First, determine a dark threshold (e.g. 10) and stopping threshold (e.g. 95%). For example, scan every column from the left-hand side to the right-hand side, if in one column, the ratio of dark pixels (value<=10) is lower than 95%, stop scanning and record the column. Repeat from four directions so that we can get a bounding box. In this project, we choose stop threshold as 0.8 and dark threshold as 0 and it performs well. **After the pre-cropping image, the image will be resized to 3200*3200,** which is a preparation of the following steps. With the simple method, the detection field is expanded, which may help to improve detection accuracy.

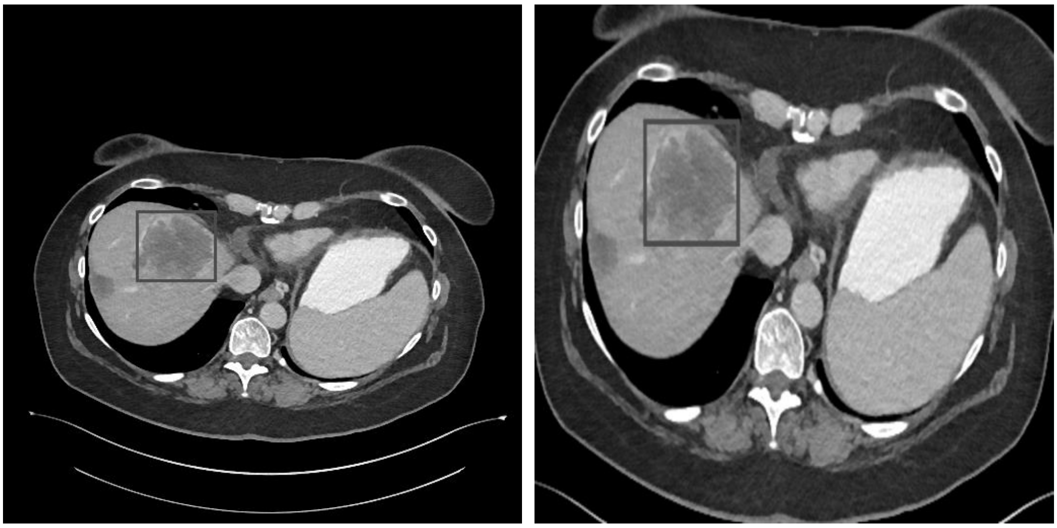


Figure 2. Left image: Original image. Right image: image after pre-cropping.

Data augmentation: Flip

Data augmentation is conducted in three ways: Left-right flip, top-bottom flip, and diagonal flip. The diagonal flip consists of the former two. That is to say, first left-right, then top-bottom. This method enlarges the number and diversity of the training dataset, which provides higher stability and accuracy. We use python package PIL to realize them. After flipping, there are two methods “vertical” and “horizontal” to change the position of bounding boxes accordingly. Randomize idea is applied in flipping. In each epoch, the direction of flipping is randomly selected, which saves the memory comparing to flip all of them and add all of them into the training dataset.

Here is a left-right flip example.

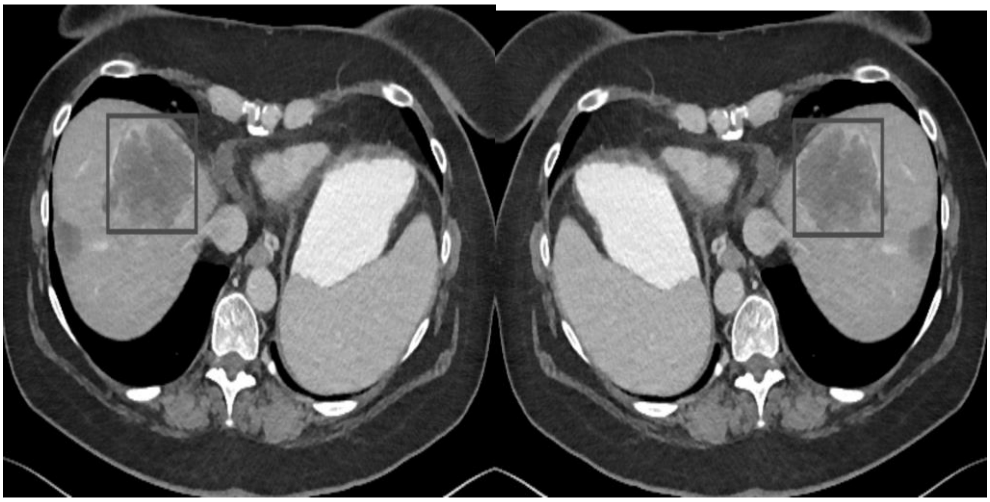


Figure 3. Left image: original image. Right image: left-right flip

Feature Pyramid.

We mainly use the feature pyramid method to get a three-channel image from a grey image. Faster R-CNN takes the output of the feature pyramid. Today's feature pyramid only considers from top to bottom or from bottom to top. Just like the following.

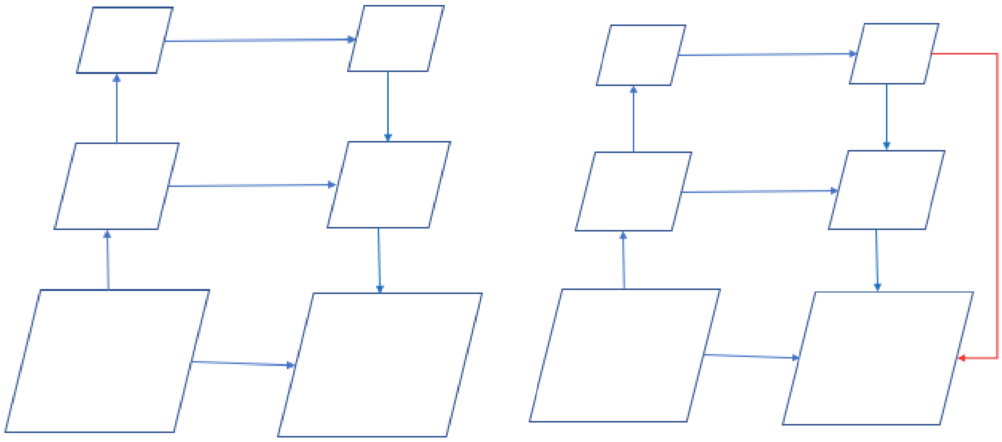


Figure 4. Original feature pyramid [6] and new feature pyramid.

We all know that after convolution, some information may be lost. Therefore, according to ResidualNet [8] and DenseNet [9] idea, we use the following method for the feature pyramid.

Theoretically, we believe that the feature pyramid method will be better than another pyramid method. We not only consider top-down or button-up method feature pyramid, but also use the residual and dense idea. Therefore, we have strong confidence that our model will improve object detection IOU accuracy.

Final Model

The following is a proposed model, which combines all methods.

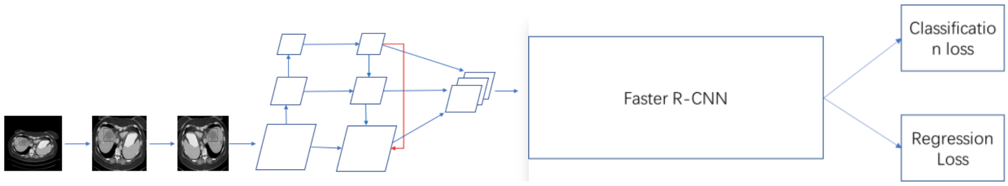


Figure 5. Proposed Model. First, using pre-cropping to crop the image. Then, using data augmentation, such as flipping the image. After that, using the feature pyramid method. Finally, the image will go to Faster R-CNN[3] to train the model.

The above is our final model. We use pre-cropping, data augmentation, and feature pyramid sequentially. After that, Faster R-CNN will be trained by the processed images. The loss is classification Loss and regression Loss.

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*).$$

Figure 6. classification Loss and regression Loss defined in Faster R-CNN[3]. p^* *and* t^* are ground truth, and p_i *and* t_i are predicted. N_{cls} *and* N_{reg} are coefficients defined to balance the weight of

classification loss and regression loss.

L_{close} is cross-entropy loss, and L_{reg} is smooth L1 loss defined in Fast R-CNN[10]

In the model, we mainly change the image pre-processing part, including pre-cropping, data augmentation, and feature pyramid. For the Faster R-CNN part, we hardly change the model structure.

Experimental result.

Result analysis from statistics

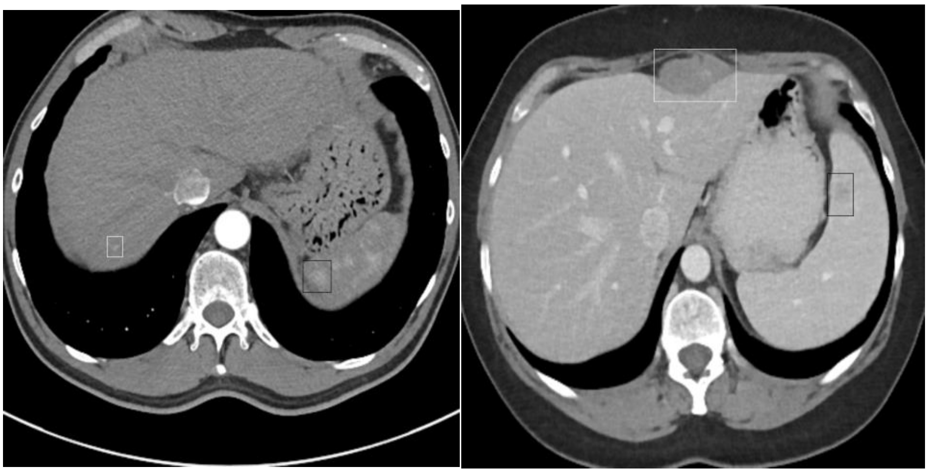
As we focus on preprocessing for faster-RCNN, we adopt faster-RCNN without any preprocessing as the baseline, which simply duplicates the grey image as the three input channels of faster-RCNN. The IOU accuracy is 61.92%. To explore the influence of each method of the model, experiments conduct based on a different combination of methods. To get a reliable experiment result, we run five times for each model and calculate average IOU accuracy as a result. The following table is the result. It can be seen that our final model increases by 7.36% IOU accuracy. Then we analyze by deleting every single method so that the influence and contribution of each method are clear. After this experiment, we know that the importance rank of these processes is: pre-crop, flip, new FPN. And their influences are close to each other since the results after deleting each of them is all-around 67%. Besides, we change our new FPN into the original FPN and compare them. It shows that a new FPN is better than the original one since there is an increase of 0.78%. The following is the experimental result table.

Method	IOU accuracy
Baseline (only faster-RCNN)	61.92%
Baseline + Pre-cropping + flipping	67.76%
Baseline + New FPN	63.92%
Baseline + Pre-cropping + flipping + Original FPN	68.5%
Final model (Baseline + Pre-cropping + flipping + New FPN)	**69.28%**

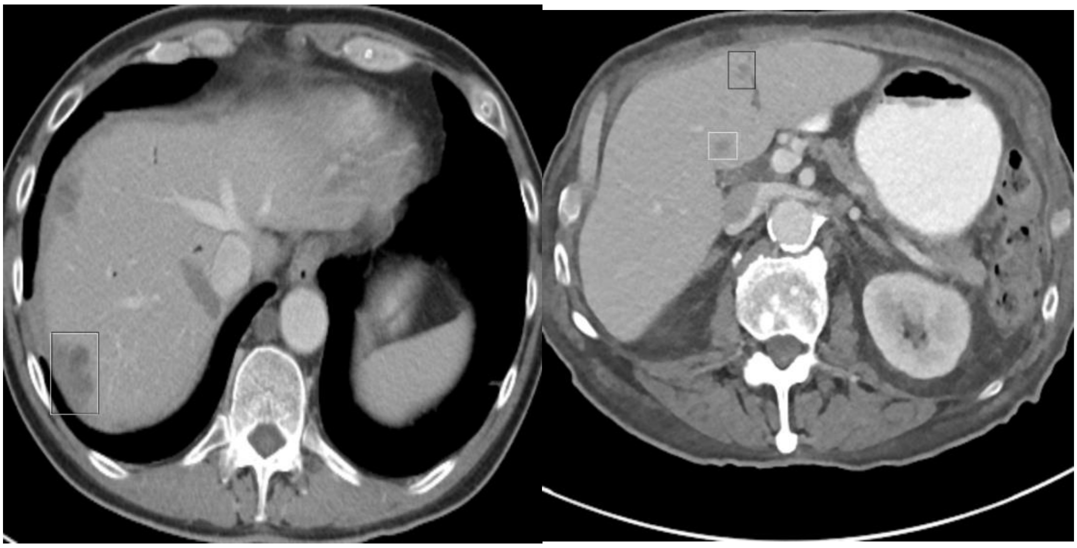
Result analysis from images

In this part, the predicted bounding box is assigned as black while the ground truth is in a white box. Our model's performance has the following characteristics:

1. **Almost all the bounding box is on the liver**. From this point, we know that flipping does not influence its judgment about direction and location. There are only two predicted bounding boxes, not on the liver among 250 images. To solve this little problem, maybe we can use a more advanced pre-cropping method to just crop the liver out. Here are the two examples.

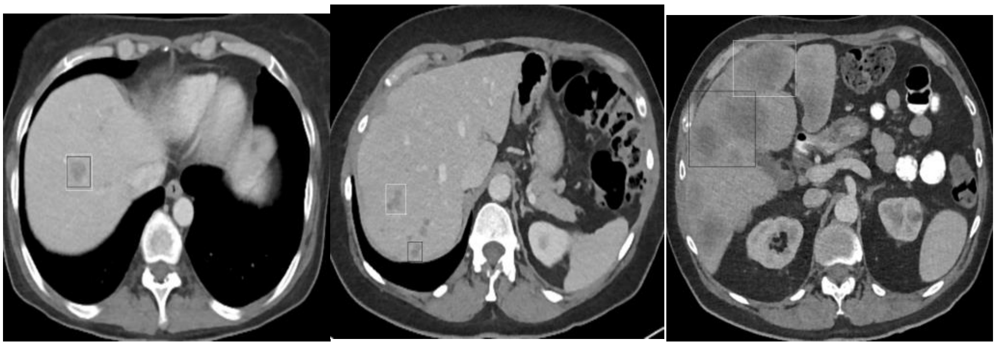


2. **IOUs cluster at 0 and 1**. In other words, it will give a perfect bounding box or it will give a wrong answer. In the left image, two bounding boxes nearly overlap each other. In the right image, two bounding boxes are distant.



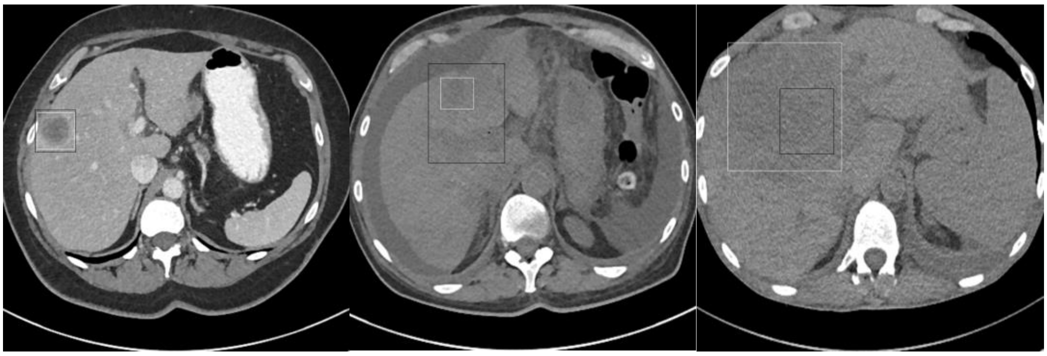
3. **It performs quite well when there is only one possible choice**. However, it is difficult for this model to make a choice when there are some possible lesions. Most of the time, the model can find one, but it does not know how to find the

best. A large number of mistakes happen due to this. The left one is an image with less interruption, and it performs perfectly. The middle one and the right have many possible choices, then the model confuses.

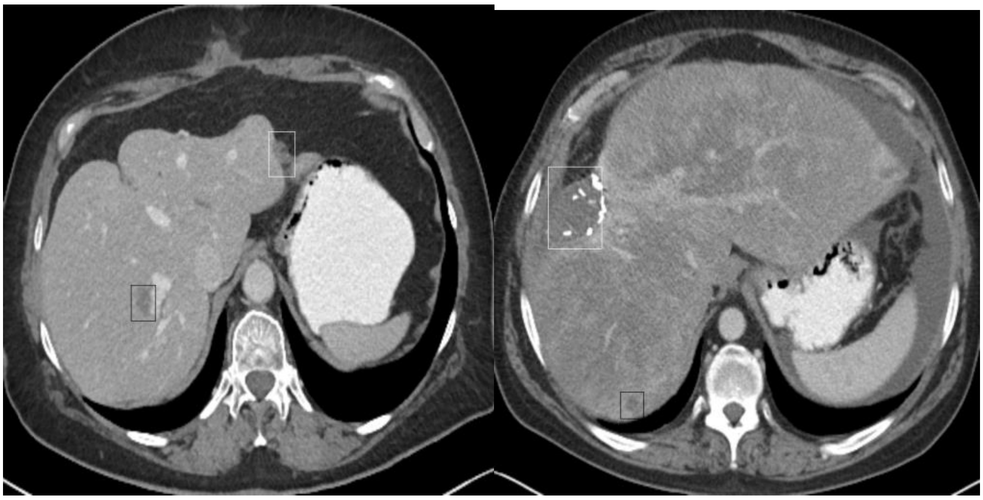


4. **This model often ignores the fact of size when comparing possible choices**. The goal is to choose the lesion with the most probability. One of the key points is the size of possible lesions. But our model cannot realize the importance of size, and it often chooses the smaller one among the candidates. This point is a complement of point 3, and it is one of the reasons for the wrong choice. There are three examples.

5. **It performs well when the contrast is high**. However, mistakes in range sometimes happen when the contrast is not high because the edge is relatively difficult to distinguish. To solve this problem, we can add a contrast enhancement process. The left one is a high-contrast image, and it can give the result accurately. The middle one and the right one are low-contrast images so that the ranges are not accurate.



6. **It performs not well when the lesion is on the edge of the liver**. One possible reason is that there is so many noise component on the edge, so it is easy to classify the real lesion into noise. To solve this problem, we can increase the number of training cases with lesions on the edge by data augmentation method. Here are two miss-detected examples with lesions on the edge.



Future Research.

1. **Best thresholds for pre-cropping**. Further experiments are required to find out the best thresholds for pre-cropping to get higher accuracy.

2. **Bounding box flipping**. Only flip the bounding box to realize data augmentation. It needs to consider the compatibility problem between the flipped area and neighbor area. Maybe we can use the average filter on the edge to make this fusion natural

3. **A more complete FPN model**. Due to the limit of memory, we simplified our FPN structure in the experiment. The following is our original design. It uses more images to generate one image. We believe it will give a better result.

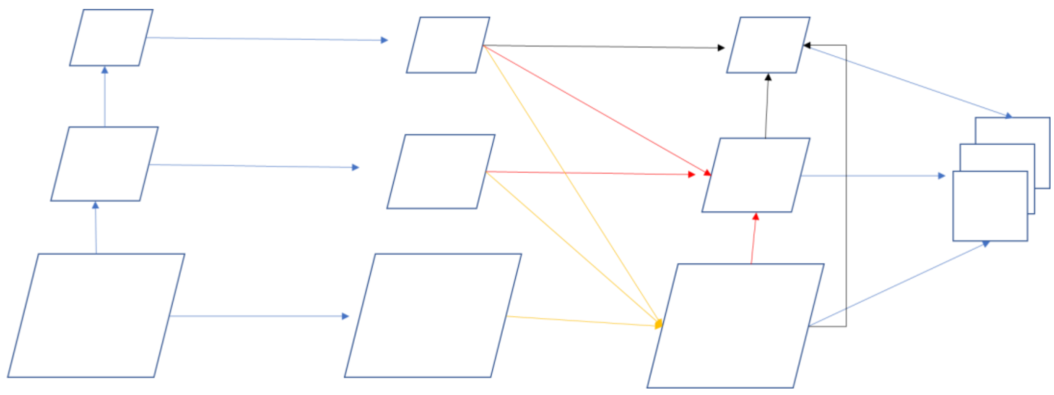


Figure 7. Yellow Line: Create The biggest image. Red Line: Create a Medium image. Black Line: Create the smallest image.

4. **Faster R-CNN can be modified**. Since faster R-CNN is a method for more than one object, we can make it special for our goal, modify it to focus on only one object, and increase its accuracy.

Conclusion

In this project, we focus on preprocessing for Faster R-CNN to design a method to detect liver lesions. Flipping, pre-crop and new FPN are the three main parts in our method. Together with Faster R-CNN, the IOU accuracy is 69.28%, increase 7.36 than baseline (just use Faster R-CNN), which is a significant improvement. The three main parts have a similar contribution between 2% to 3%, especially, new FPN surpasses original FPN 0.78%. There are still some problems according to the image result. The model performs better when it has fewer possible choices, but it is hard for it to choose among all possible ones. Lesion size, image contrast, lesions on the edges, and other elements are problems that we should consider in the future.

References:

[1]. Lee, S. Y., Kim, H. J., & Choi, D. (2015). Cell sources, liver support systems, and liver tissue engineering: alternatives to liver transplantation. International journal of stem cells, 8(1), 36.

[2]. https://www.webmd.com/hepatitis/liver-lesions#1

[3]. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[4]. Chen L C, Papandreou G, Schroff F, et al. Rethinking atrous convolution for semantic image segmentation[J]. arXiv preprint arXiv:1706.05587, 2017

[5]. Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.

[6]. Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern

Recognition. 2018: 8759-8768.

[7]. Pang J, Chen K, Shi J, et al. Libra R-CNN: Towards Balanced Learning for Object Detection[J]. arXiv preprint arXiv:1904.02701, 2019

[8]. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770–778).

[9]. Gao H., Liu, Z., Weinberger, K. Q., & Maaten, L. van der. (2016). Densely Connected Convolutional Networks. ArXiv Preprint ArXiv:1608.06993.

[10]. Girshick, R.B. (2015). Fast R-CNN. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1440-1448.