Project report: Mammogram abnormality detection

Vu Quang Truong[a]

[a]*Hanoi University of Science and Technology, Vietnam*

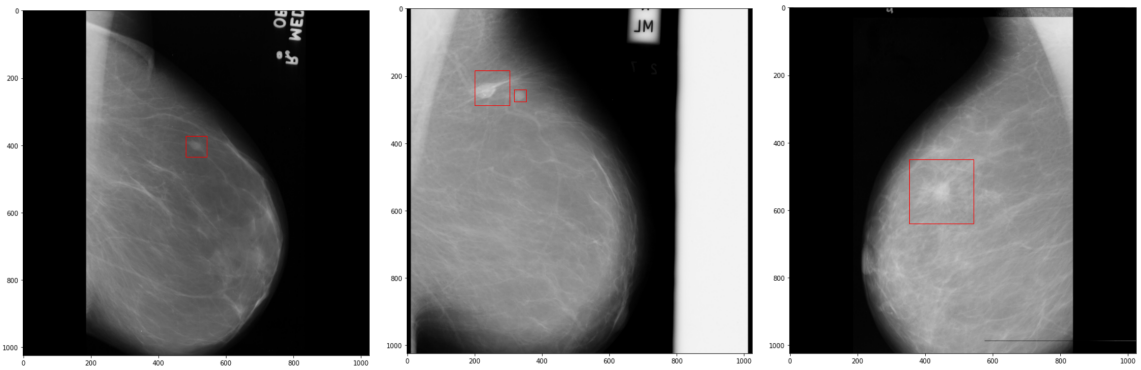
1. Problem Definition

1.1. Dataset

The data is extracted from `http://peipa.essex.ac.uk/info/mias.html` which consists of 322 mammograms. There are 111 images with bounding boxes included and 211 images without any bounding boxes. The bounding boxes are given in the form of $(x_{center}, y_{center}, radius)$.

1.2. Problem

We need to detect suspicious area(s) in the mammogram.



(a) (b) (c)

Figure 1: Images with ground truth boxes

2. Proposed methods

In this report, the Faster R-CNN model with ResNet50 as the backbone is used together with Feature Pyramid Network (FPN) to extract features at multiple scale levels. The structure of the network is described below.

Email address: `truong.vq194198@sis.hust.edu.vn` (Vu Quang Truong)

2.1. ResNet50 backbone

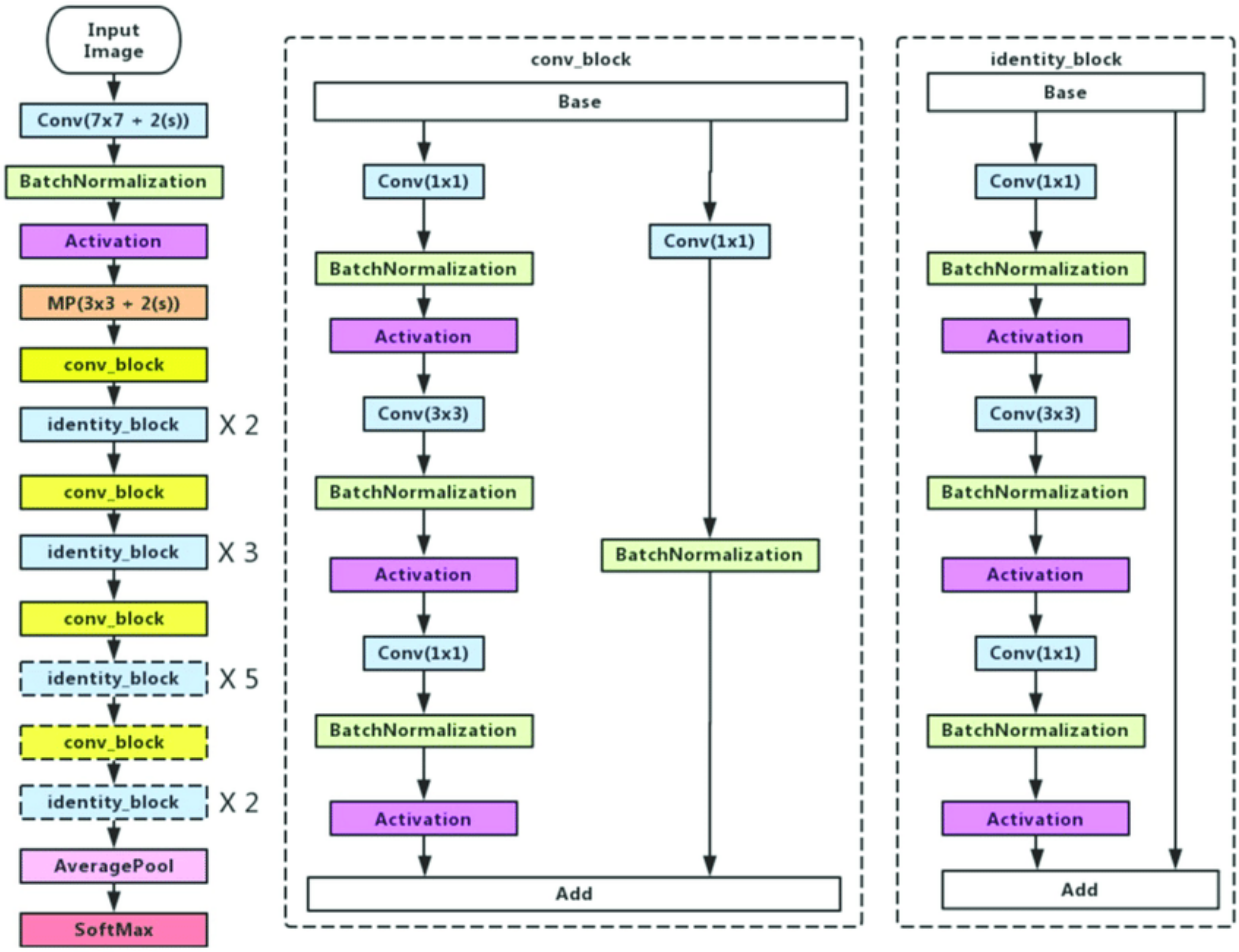


Figure 2: ResNet50 architecture

Residual Network (ResNet) was first introduced in [1] and soon became a game-changer in Deep Learning. It allows us to train much deeper networks with better performance.

ResNet50 is a version of ResNet which is 50 layers deep. It includes five stages. In each stage (except stage 1), there are one convolution block and some identity blocks, as shown in 2.

In each convolution block, three convolution steps are executed. The first step is used to downsample the feature map from the previous layer (i.e., halve the height and width). Skip connection is also utilized so that we can train a deeper model without concerning about the vanishing gradient problem. Note that in the convolution block, we need to use a 1×1 convolution layer with the stride of 2 in the skip connection to keep the input size equal to the output size so that they can be added up.

The identity block has the same architecture as the convolution block, except the fact that it does not downsample the feature map, and as a consequence, it does not need to include the convolution step in the skip connection to resize the input.

Skip connections have a significant effect on the performance of ResNet. In fact, ResNet is the first to introduce the concept of skip connections. Since then, many different architectures have taken advantage of this novel concept, such as U-Net, DenseNet, FPN, etc. The effects of skip connections are listed below:

- They mitigate the problem of vanishing gradient by allowing this alternate shortcut path for the gradient to flow through.
- They allow the model to learn an identity function which ensures that the higher layer will perform at least as good as the lower layer, and not worse.

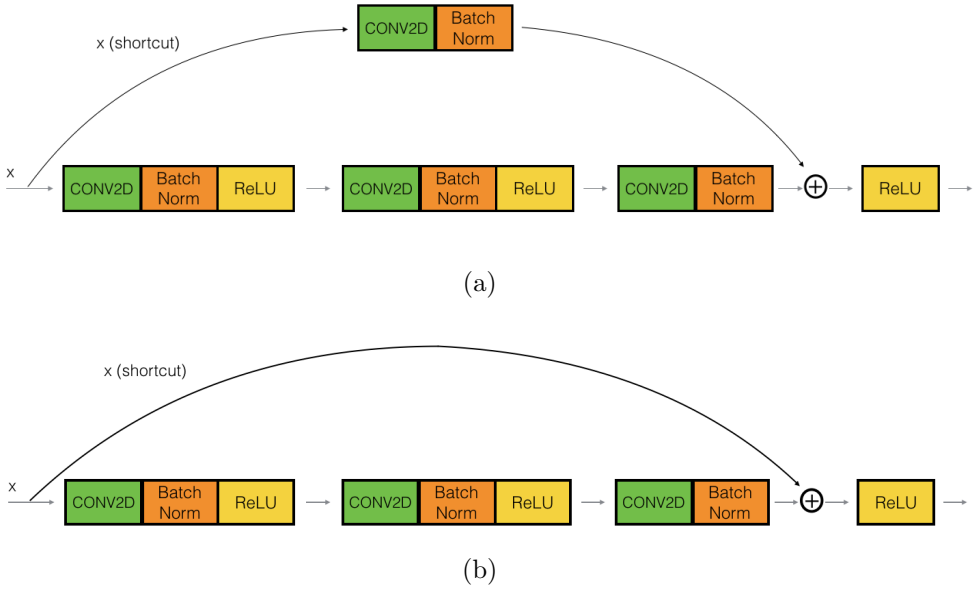


Figure 3: Skip connections with (a) and without (b) downsampling

2.2. Feature Pyramid Network (FPN)

FPN is a feature extractor used for pyramid concept in convolutional network and first introduced in [2]. It replaces the feature extractor of detectors like Faster R-CNN and generates multiple feature map layers (multi-scale feature maps) with better quality information than the regular feature pyramid for object detection. Its architecture is shown in Fig 4.

FPN consists of two pathways: the bottom-up and the top-down. The bottom-up pathway is a regular convolutional network that is used to extract features. As we go up following this pathway, the spatial resolution of the feature maps decreases, but the semantic value increases.

With the aim of taking advantage of high-resolution feature maps to detect small objects while still maintaining semantic value from upper layers, FPN proposed a top-down pathway to construct higher resolution layers from a semantic-rich layer.

The reconstructed layers are semantic strong but the locations of objects are not precise after all the downsampling and upsampling. Hence, the authors of FPN introduced skip lateral connections between reconstructed layers and the corresponding feature maps to help the detector to predict the location betters.

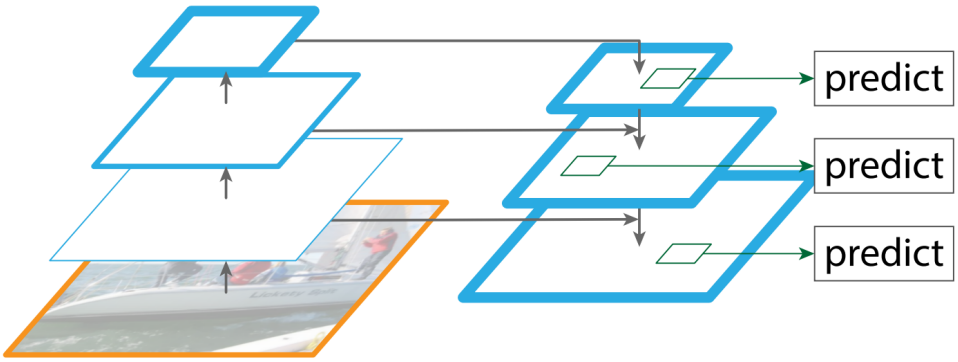


Figure 4: FPN architecture

2.2.1. Bottom-up pathway

The bottom-up pathway of FPN makes use of ResNet. As mentioned in the previous section, ResNet comprises five stages and after each stage, the spatial resolution is divided by two. The output of each convolution stage is labeled as Ci and later used in the top-down pathway. The architecture of the bottom-up pathway is illustrated in Fig 5.

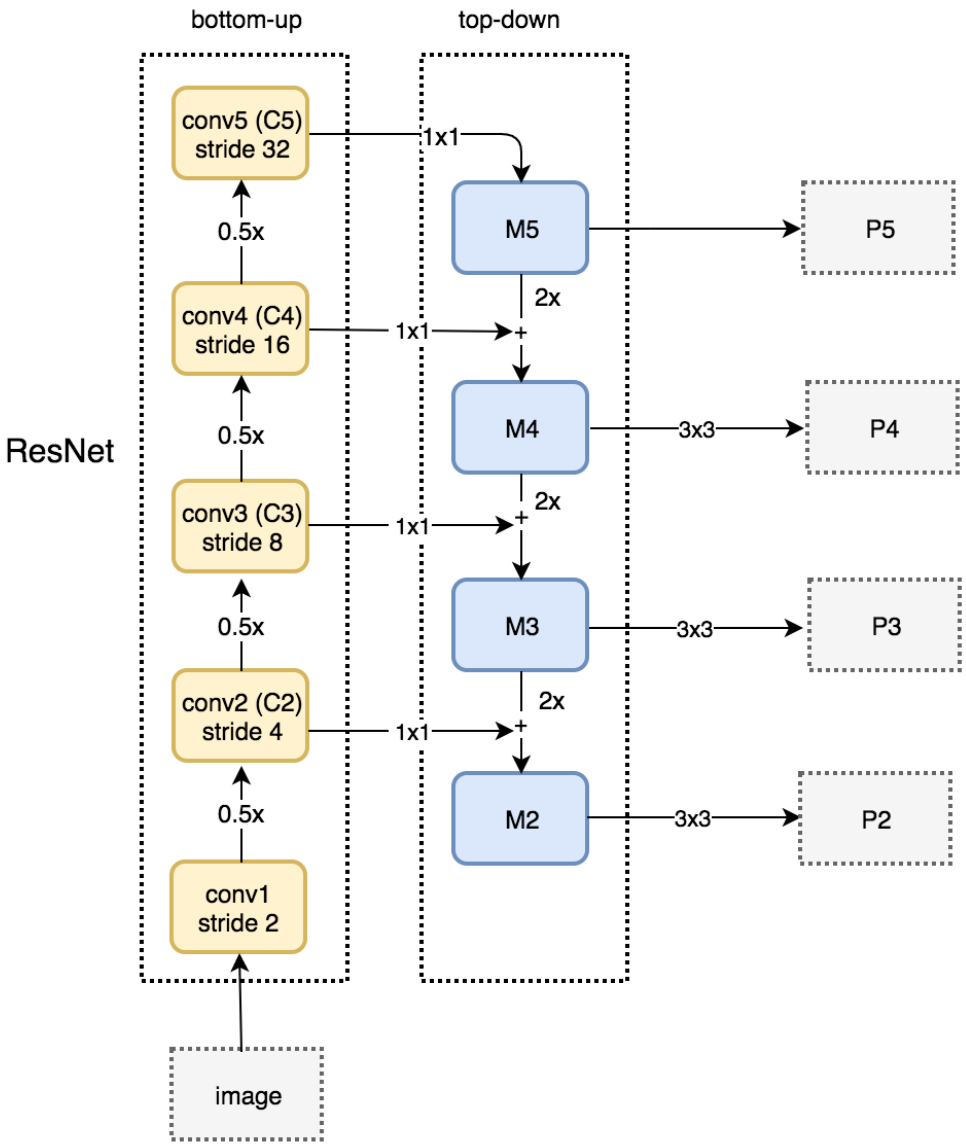


Figure 5: FPN pathway architecture

2.2.2. Top-down pathway

The top-down pathway starts by applying a 1×1 convolution to $C5$ to reduce its channel depth to 256 and create $M5$. As we go down the path, the feature map from the previous layer is upsampled by 2 using nearest neighbor upsampling. We again apply a 1×1 convolution on the corresponding feature maps in the bottom-up pathway. Subsequently, we add them up element-wise and apply a 3×3 convolution on the result to reduce the aliasing effect of upsampling. The final set of feature maps is denoted by $\{P1, P2, P3, P4\}$, corresponding to $\{C2, C3, C4, C5\}$ that are respectively of the same spatial sizes.

Because all levels of the pyramid use shared classifiers/regressors as in a traditional featurized image pyramid, the authors of FPN fix the feature dimension (numbers of channels,

denoted as d) in all the feature maps.They set $d = 256$ and thus all extra convolutional layers have 256-channel outputs.

2.3. Faster R-CNN model with FPN and ResNet50

2.3.1. FPN with Region Proposal Network (RPN)

Region Proposal Network is a network used in Faster R-CNN to make proposals of RoI in the feature maps, which is first introduced in [3]. When FPN is utilized, it feeds the feature maps at all scale levels to RPN, which allows Faster R-CNN to detect objects at multiple scale levels.

At each scale level, a 3×3 convolution filter is used, followed by two 1×1 convolutions for objectness prediction and bounding box regression. Those convolutions are called *RPN head*. The same RPN head is applied at every level of FPN, as demonstrated in Fig 6.

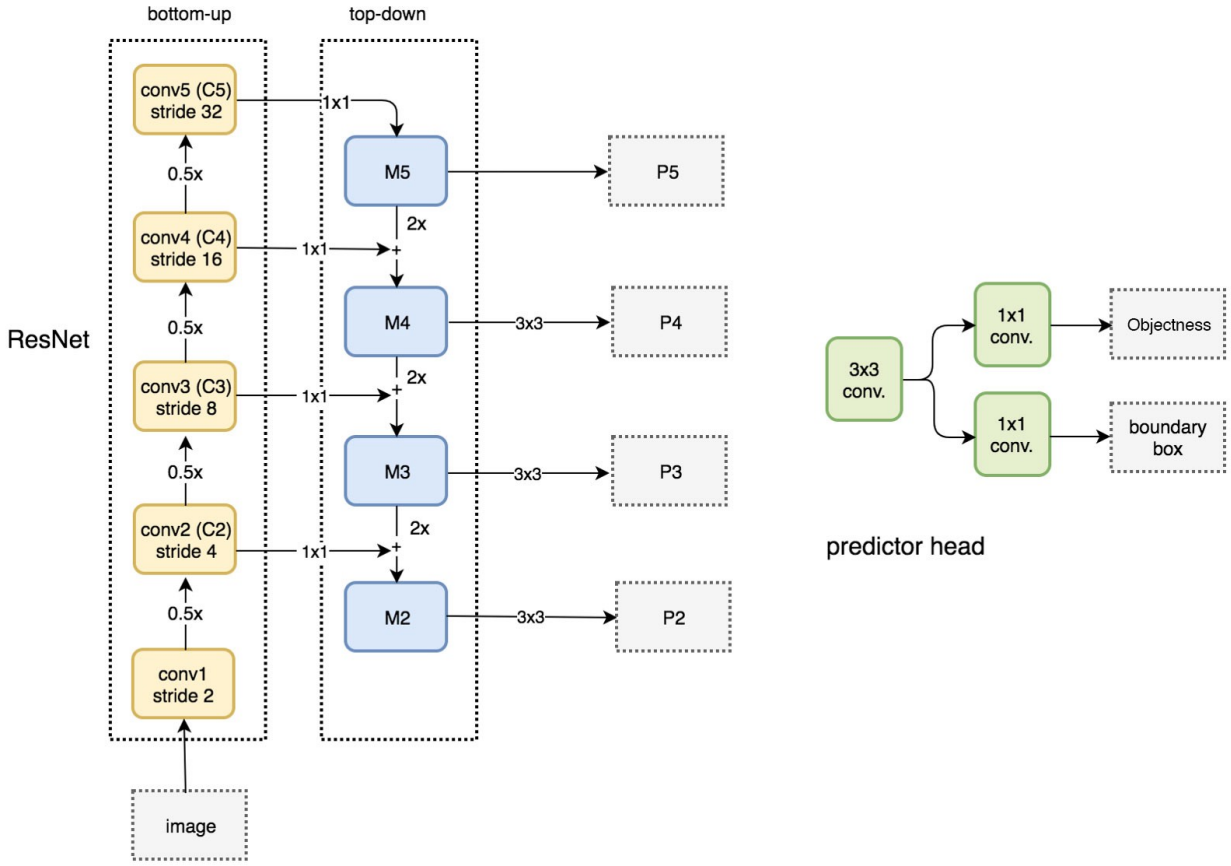


Figure 6: FPN and RPN

2.3.2. FPN with Faster R-CNN

The architecture of the original Faster R-CNN without FPN is first introduced in [3] and shown below in Fig 7. It has only one feature map to feed into RPN to create RoIs. We then use RoI and the feature map to create feature patches which is then fed into the RoI pooling.

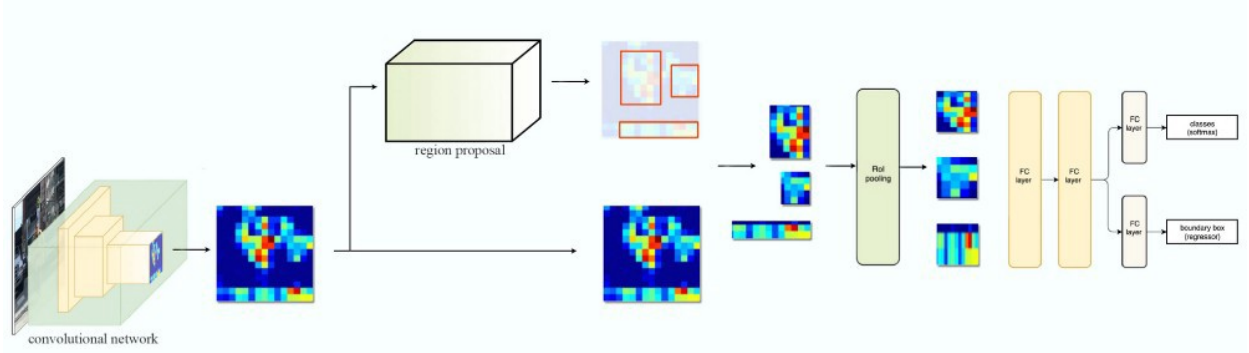


Figure 7: Architecture of the original Faster R-CNN

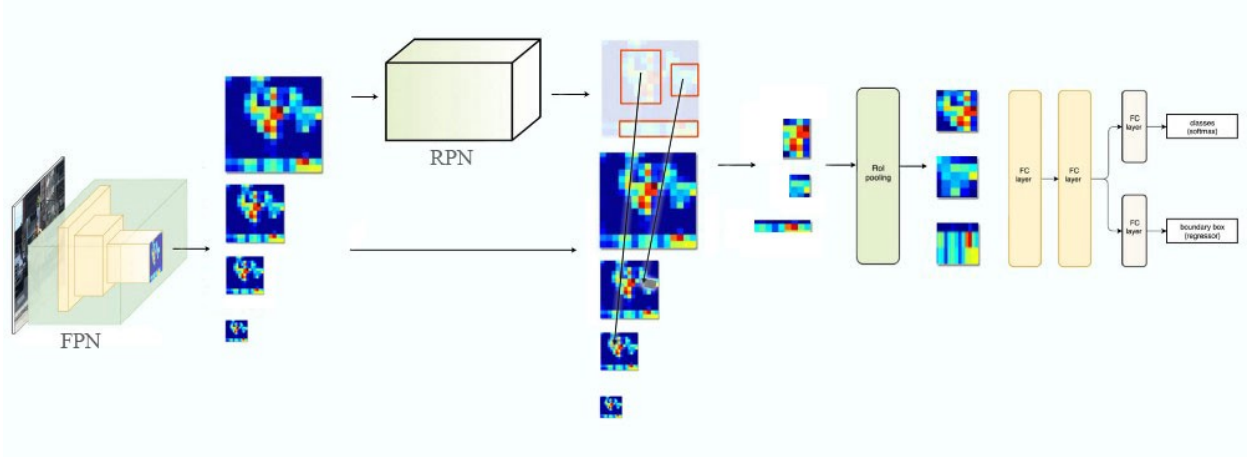


Figure 8: Architecture of Faster R-CNN with FPN included

When FPN comes into use, it fed multiple feature maps into RPN to generate the RoIs as illustrated in Fig 8. Based on the size of the RoIs, we can select the most proper layer scale to get the feature patches. In the original paper of FPN, the authors proposed a formula that is used to select the scale:

$$k = \lfloor k_0 + log_2(\sqrt{wh}/224) \rfloor \tag{1}$$

where $k_0 = 4$; w and h are the weight and height of the RoI.

The feature patches are then put through the RoI pooling and fully connected layers to get the final prediction results.

2.4. Data augmentation

Due to the lack of mammograms in the dataset, data augmentation is necessary to improve the performance of the model. In this project, I applied five methods to augment the dataset with the probability of 0.5 for each, as shown in Fig 9.

- Horizontal Flip

- Vertical Flip
- Random Brightness Contrast
- Random Snow
- Random Contrast

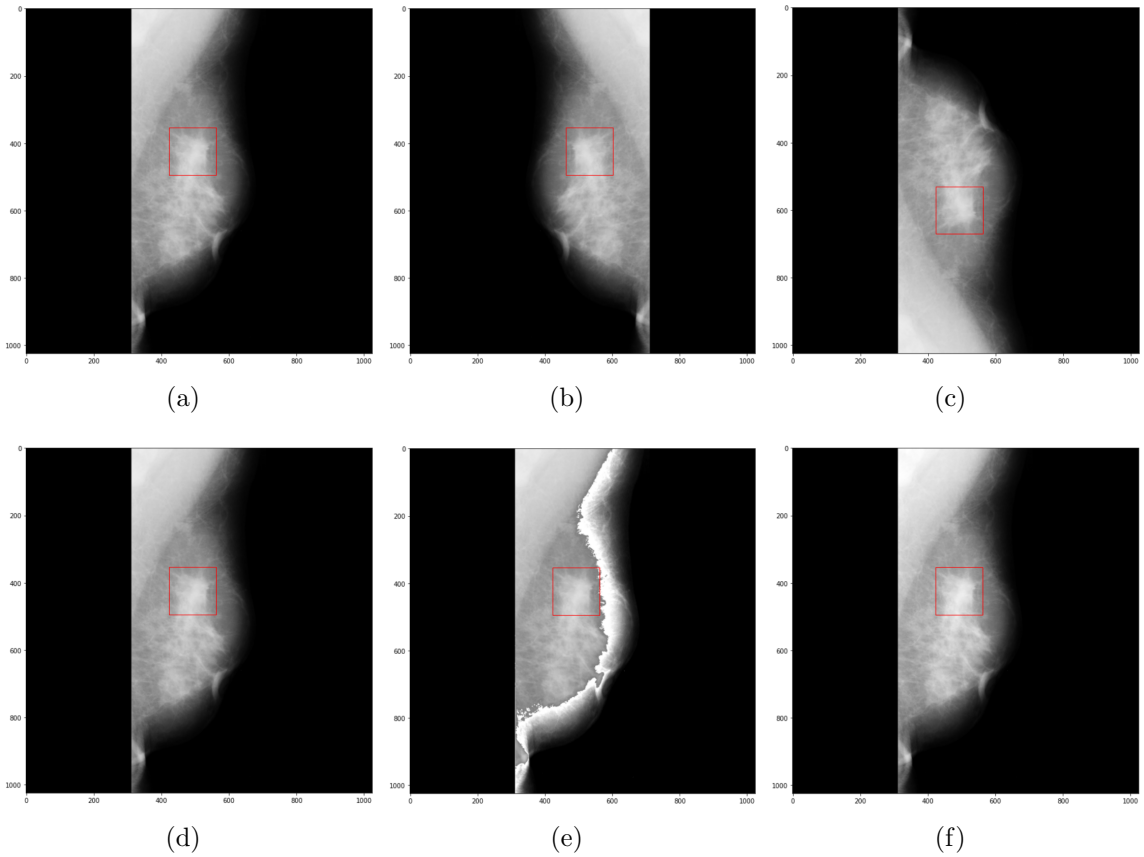


Figure 9: Data augmentation methods: a) Original image, b) Horizontal Flip, c) Vertical Flip, d) Random Brightness Contrast, e) Random Snow, f) Random Contrast

2.5. Non-maximum Suppression (NMS)

Non-maximum Suppression is an algorithm to select good bounding boxes out of many overlapping bounding boxes in the prediction. The pseudo-code for NMS is shown below in Fig 10.

Algorithm 1 Non-Max Suppression

procedure NMS(B,c)
  $B_{nms} \leftarrow \emptyset$ Initialize empty set
  for $b_i \in B$ **do** => Iterate over all the boxes
    $discard \leftarrow$ False Take boolean variable and set it as false. This variable indicates whether b(i) should be kept or discarded
    for $b_j \in B$ **do** Start another loop to compare with b(i)
      if same$(b_i, b_j) > \boldsymbol{\lambda_{nms}}$ **then** If both boxes having same IOU
        if score$(c, b_j) >$ score(c, b_i) **then**
          $discard \leftarrow$ True Compare the scores. If score of b(i) is less than that of b(j), b(i) should be discarded, so set the flag to True.
    if not $discard$ **then** Once b(i) is compared with all other boxes and still the discarded flag is False, then b(i) should be considered. So add it to the final list.
      $B_{nms} \leftarrow B_{nms} \cup b_i$
  return B_{nms} Do the same procedure for remaining boxes and return the final list

Figure 10: Non-maximum suppression pseudo code

Because in the dataset, almost every image has only one ground truth bounding box, I applied one more step after NMS, in which all bounding boxes with score less than a threshold are removed.

3. Experiment Results

3.1. Implementation

I trained the Faster R-CNN model for 10 epochs with SGD optimizer. A learning rate schedule in which the learning rate shrinks 10 times after 3 epochs is used. The train-test split ratio is 80-20. The IoU threshold and the score threshold for NMS are 0.01 and 0.4, respectively.

The whole implementation is based on the tutorial on `https://www.youtube.com/watch?v=Egz4bXMlmDM&list=LL&index=4&ab_channel=FormulaTrinity` and executed on Google Colab. I posted the code on `https://colab.research.google.com/drive/19snTe21jq-qU3r4KKjrvfBq0GZP6Eh7O?usp=sharing`.

3.2. Result

After training, the model is evaluated on the test set which consists of 22 mammograms. The metrics used are Average Precision (AP) and Average Recall (AR) with multiple IoU thresholds and area sizes (small, medium, large). The result is shown in Table 1.

Table 1: Evaluating result

Metric	Score
Average Precision (AP) @[$IoU = 0.50 : 0.95 \mid area = all \mid maxDets = 100$]	0.205
Average Precision (AP) @[$IoU = 0.50 \mid area = all \mid maxDets = 100$]	0.450
Average Precision (AP) @[$IoU = 0.75 \mid area = all \mid maxDets = 100$]	0.163
Average Precision (AP) @[$IoU = 0.50 : 0.95 \mid area = small \mid maxDets = 100$]	0.000
Average Precision (AP) @[$IoU = 0.50 : 0.95 \mid area = medium \mid maxDets = 100$]	0.236
Average Precision (AP) @[$IoU = 0.50 : 0.95 \mid area = large \mid maxDets = 100$]	0.270
Average Recall (AR) @[$IoU = 0.50 : 0.95 \mid area = all \mid maxDets = 1$]	0.223
Average Recall (AR) @[$IoU = 0.50 : 0.95 \mid area = all \mid maxDets = 10$]	0.286
Average Recall (AR) @[$IoU = 0.50 : 0.95 \mid area = all \mid maxDets = 100$]	0.341
Average Recall (AR) @[$IoU = 0.50 : 0.95 \mid area = small \mid maxDets = 100$]	0.000
Average Recall (AR) @[$IoU = 0.50 : 0.95 \mid area = medium \mid maxDets = 100$]	0.379
Average Recall (AR) @[$IoU = 0.50 : 0.95 \mid area = large \mid maxDets = 100$]	0.417

It can be concluded that the model does not have a good performance on small object detection (AP and AR for small objects are both 0.000). The result is better with larger objects.

Besides, NMS also plays an important role in the final output. Fig 11 illustrates the effect of NMS (with an additional step).

4. Conclusion

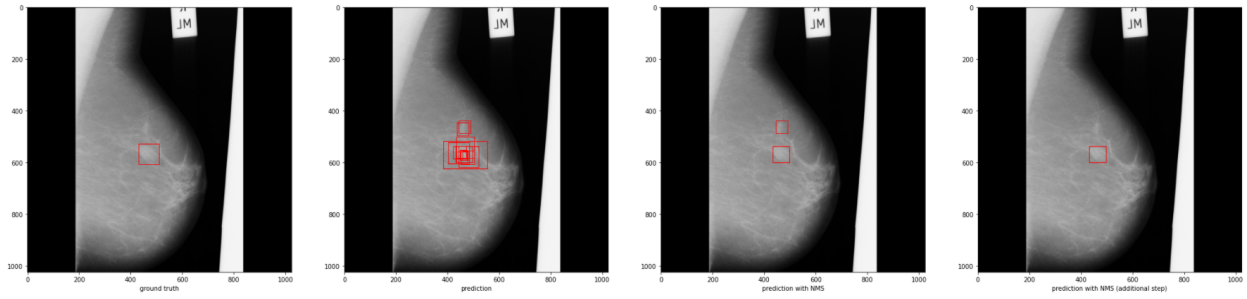
The Faster R-CNN model with ResNet50 as backbone and FPN as a feature extractor can have an acceptable result on such a small dataset. However, it can not detect small suspicious areas on the mammogram. NMS with an additional step significantly improves the quality of the final result. Maybe we can take advantage of one more level in the top-down pathway of FPN to detect small objects.

References

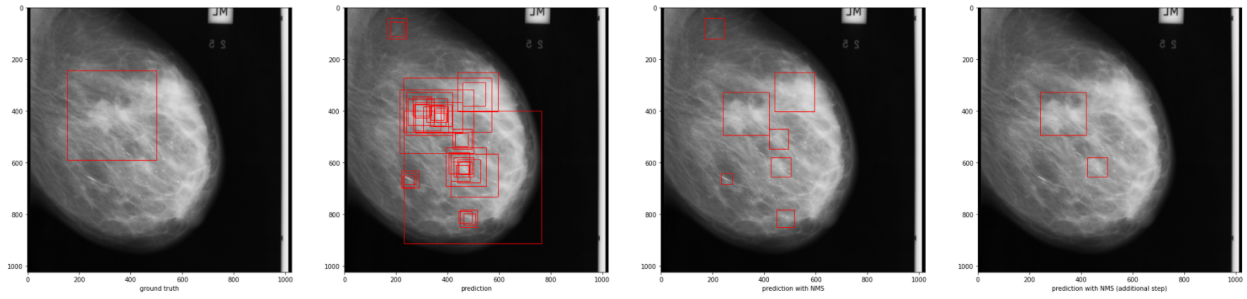
[1] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[2] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.

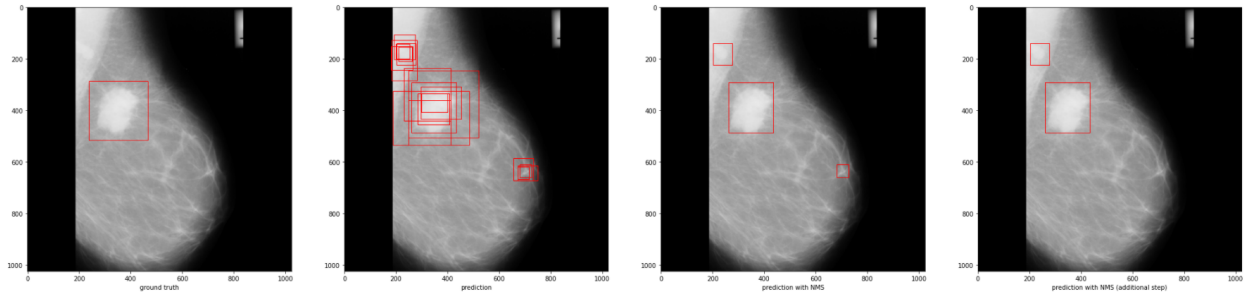
[3] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28.



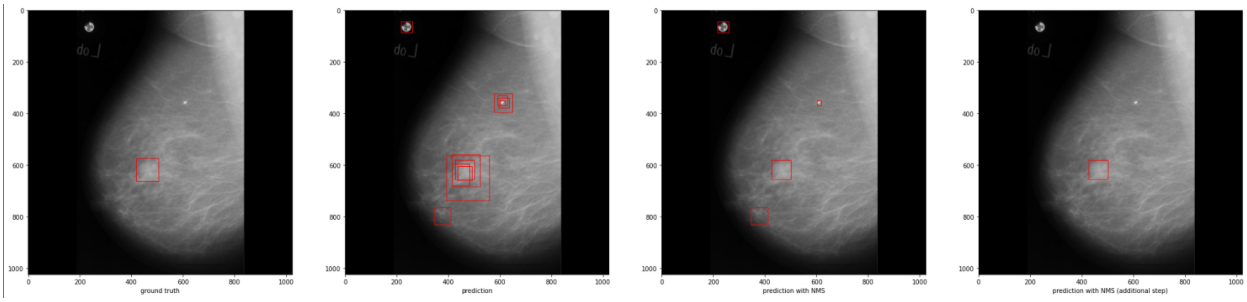
(a)



(b)



(c)



(d)

Figure 11: NMS effect on prediction result