2018 11th International Symposium on Computational Intelligence and Design

# An Improved Faster R-CNN for Object Detection

Yu Liu
Antai College of Economics and Management
Shanghai Jiao Tong University
Shanghai, China
yu___liu@126.com

***Abstract*—Among various target detection algorithms, Faster R-CNN is an algorithm with excellent performance both in detection accuracy and in detection speed at present. However, it still has some shortcomings such as too many negative samples. To address the problem of Faster R-CNN, two strategies, hard negative sample mining and alternating training, are introduced. Hard negative sample mining is used to obtain hard negative sample which retrain the model for improving the trained model, and the alternating training make RPN and Fast R-CNN in Faster R-CNN share convolutional layers, rather than learn two independent networks. The simulation result show that the proposed algorithm has great advantages in terms of detection accuracy.**

***Keywords—Faster R-CNN; hard negative sample; alternating training***

## I. INTRODUCTION

With the improvement integration capabilities of electronic hardware device, more and more intelligent electronic devices appear in people's lives, and digital image becomes an indispensable information carrier in intelligent electronic devices. However, the number of digital images is huge, and it is impossible to manually classify these images. Based on this background, object detection came into being. The problem to be solved by object detection is to detect the location of the object on the image and determine the type of object.

The traditional target detection methods training the classifier based on the target feature such as scale invariant feature transform (SIFT) [1], speeded-up robust features (SURF) [2], Harr [3], histogram of oriented gradient (HOG) [4], Local Binary Pattern (LBP) [5], Strip [6]. The detection process has three steps. First, the regions where the target may exist on the image are obtained, and then the corresponding hand-designed features of these regions are extracted, and finally the selected features are classified by the classifier. The traditional target detection methods have made great progress both in detection accuracy and in detection speed. However, there are still many deficiencies. For example, the features are hand-designed, and this requires researchers to have excellent prior knowledge. In addition, the performance is often poor when the background or lack of illumination. The performance is often poor when the shape of the object has big change, the background is complex, or the illumination is insufficient.

As the research of feature extraction develop, researchers have found that convolutional neural networks can learn better features from large-scale data and overcome the shortcomings of hand- designed features. In 2014, Girshick et al. [7] proposed a region-based convolutional neural network (Region-based CNN, R-CNN) model, which became representative of target detection based on classification convolutional neural networks. First, the model uses the selective search algorithm to extract several proposal regions from the image. Then, the proposal regions are changed to a uniform size, and the feature is extracted using convolutional neural network. Finally, the features are classified by multiple support vector machines (SVM). In order to improve the speed and accuracy of the R-CNN model, fast R-CNN model is proposed [8]. Compared with the R-CNN model which extracts the feature for each candidate region, the Fast R-CNN model only extracts once for the detected image, and then the feature corresponding to the proposal region is mapped to a feature vector with fixed length by spatial pyramid pooling. Both R-CNN and fast R-CNN use selective search algorithm to extract proposal region, which takes a lot time. To overcome this issue, the Faster R-CNN which replaces the selective search algorithm with region proposal networks (RPN) is proposed [9]. The RPN which connects to the last convolutional layer of Fast R-CNN is used to generated proposal region by predicting object bounds and objectness scores with a series of anchor boxes.

Based on Faster R-CNN, scholars have conducted a lot of research to further optimize the performance of target detection. The authors proposed a refining block for Fast R-CNN which merge the block and Faster R-CNN into a single network (RF-RCNN) to detect RoadView image that consists of high resolution street images [10]. To address the problem of Faster R-CNN, a novel two-stage cascade multi-scale proposal generation network was proposed. In the proposed network, original RPN is used to initially generate coarse proposals, then another network with multilayer features and RoI pooling layer are introduced to refine these proposals [11]. A different scales face detector (DSFD) based on Faster R-CNN which can improve the precision of face detection while performing as real-time a Faster R-CNN was proposed [12]. A simple but effective baseline that adopted Faster-RCNN's network architecture for small traffic sign detection was present, and it increase anchors' density and feature maps' resolution, decrease reference window size [13].

In this paper, an improved Faster R-CNN merging hard negative sample mining and alternating training are proposed. Hard negative sample mining is used to obtain hard negative sample to retrain the model for superior model, and the alternating training enable the RPN and fast R-CNN in Faster R-CNN to share convolutional layers.

## II. FASTER R-CNN

The structure of Faster R-CNN is shown in Fig. 1. As we can see from the figure that Faster R-CNN contains two components: region proposal network (RPN) and Fast R-CNN. As a full convolutional network, RPN is the core of the Faster R-CNN and is used to generate high-quality region

2473-3547/18/$31.00 ©2018 IEEE
DOI 10.1109/ISCID.2018.10128

proposal. The convolutional layer takes an image with any size as input to generate the feature by feature mapping. The feature is propagated to the RPN which used to generate the region proposal and a unique convolutional layer which is used to produce higher dimensional features. The higher dimensional features and region proposal are given as input to RoI pooling which is used to turn the higher dimensional features into a uniform size according to the region proposal. The feature from RoI pooling is fed into fully-connected layers which is consists of box-regression layer (reg) and a box-classification layer (cls) to obtain the coordinates and scores.

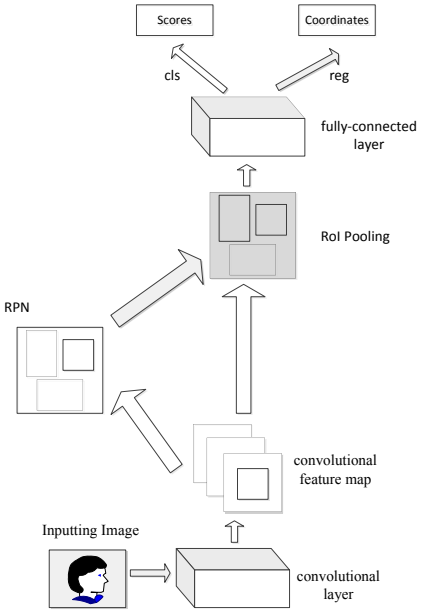


Fig. 1. The structure of Faster R-CNN

### A. Training RPNs

Faster R-CNN uses the RPN to extract the target proposal region. The principle is shown in Figure 2, and the specific process is as follows: A small network slides over the feature map obtained in the last convolutional layer, and it is fully-connected to the window with the same size on the feature map each time it slides; then each sliding window is mapped to a vector with the length of 512 (corresponding to ZFnet); finally the lower dimensional vector is fed into two sibling fully-connected layers which are box-regression layer (reg) and box-classification layer (cls). For each sliding-window location, the cls layer outputs 4k coordinates of each box, and the reg layer outputs 2k scores which estimate the probability of object (foreground) or not object (background).

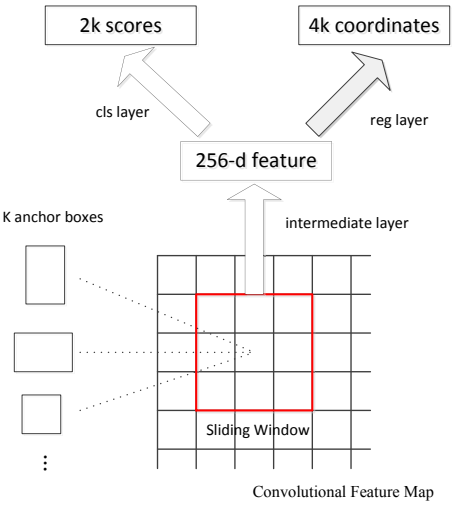


Fig. 2. The principle of RPN

The center of the n × n sliding window is defined as anchor, and each sliding window can simultaneously predict k possible proposals for each sliding-window location. The k possible proposals are k reference boxes, and each reference boxes can be uniquely determined by a scale, an aspect ratio and an anchor. By default three scales and three aspect ratios are used to determine the k=9 reference boxes at each sliding position.

To select the anchor, the anchors are divided into two categories (positive and negative) with intersection-over-union (IoU) overlap ratio between ground-truth box and anchor box as classification index. The classification rules are as follows.

Rule 1: If the anchor box has largest IoU with the ground-truth box, the corresponding anchor is labeled as positive sample.

Rule 2: If the IoU of anchor box is higher than 0.7, the corresponding anchor is labeled as positive sample. Usually, we can find enough positive samples using rule 2. However, we still use rule 1 for some extreme cases. For example, the IoUs for all anchor are not larger than 0.7.

Rule 3: If the IoU of anchor box is lower than 0.3, the corresponding anchor is labeled as negative sample.

Rule 4: The rest anchors are neither positive samples nor negative samples, and they are not used for training objective.

Rule 5: The anchor box that crosses the image boundary is also discarded.

The IoU overlap ratio is defined as

$$IoU = \frac{S_{anchorBox} \cap S_{groundTruth}}{S_{anchorBox} \cup S_{groundTruth}} \tag{1}$$

### B. Loss function

Following the multi-task loss in Faster R-CNN, we minimize the objective function definite as follows:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_{i=1}^{N_{cls}} L_{cls}(p_i, p_i^*) + \frac{\lambda}{N_{reg}} \sum_{i=1}^{N_{reg}} p_i^* L_{reg}(t_i, t_i^*) \tag{2}$$

where i is the index of the i-th anchor box, $p_i$ is the probability of anchor i being an object. If the anchor is labeled as positive sample, the corresponding probability $p_i^*$

is set to 1. In contrast, the probability $p_i^*$ is set to 0 when the anchor is labeled as negative sample. $t_i$ is a vector representing the 4 parameterized coordinates of the predicted bounding box, and $t_i^*$ is the coordinate vector of the corresponding ground-truth bounding box. $L_{cls}$ is the classification loss function, and $L_{reg}$ is the regression loss function. Ncls and Nreg are the normalization coefficients of the classification loss function $L_{cls}$ and the regression loss function $L_{reg}$ respectively. λ is the weight parameter between $L_{cls}$ and $L_{reg}$.

The classification loss function $L_{cls}$ is the logarithmic loss of two categories (target vs. non-target), and it is defined as follow

$$L_{cls}\left(p_i, p_i^*\right) = -\log\left[p_i^* p_i + \left(1-p_i^*\right)\left(1-p_i\right)\right] \quad (3)$$

For the regression loss function, it is defined as

$$L_{reg}\left(t_i, t_i^*\right) = R\left(t_i - t_i^*\right) \quad (4)$$

where R is a robust loss function with the definition as

$$R(x) = \begin{cases} 0.5x^2, |x| < 1 \\ |x| - 0.5, others \end{cases} \quad (5)$$

The four coordinates for bounding box regression are given as follow

$$\begin{cases} t_x = \dfrac{x - x_\alpha}{w_\alpha}, \quad t_w = \log\left(\dfrac{w}{w_\alpha}\right) \\ t_y = \dfrac{y - y_\alpha}{h_\alpha}, \quad t_h = \log\left(\dfrac{h}{h_\alpha}\right) \end{cases} \quad (6)$$

$$\begin{cases} t_x^* = \dfrac{x^* - x_\alpha}{w_\alpha}, \quad t_w^* = \log\left(\dfrac{w^*}{w_\alpha}\right) \\ t_y^* = \dfrac{y^* - y_\alpha}{h_\beta}, \quad t_h^* = \log\left(\dfrac{h^*}{h_\alpha}\right) \end{cases} \quad (7)$$

where x and y are the central coordinates of the anchor box, w and h are the width and height of the anchor box. $x$, $x_\alpha$, and $x^*$ are coordinates for the predicted box, anchor box, and ground-truth box respectively (likewise for y, w, h). The predicted box is obtained from the process of fine-tuning the anchor box by regression until it is approaching ground-truth box. The process of fine-tuning is shown in Fig. 3.

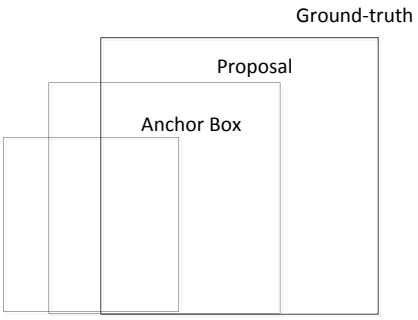

Fig. 3. The process of fine-tuning

## III. IMPROVED FASTER R-CNN

Although Faster R-CNN has superior detection performance, there are still some shortcomings. For example, the number negative sample is far greater than that of positive sample. To overcome these shortcomings, some new strategies are introduced in this section.

### *A. Hard Negative Sample*

The Faster R-CNN model has a big problem in the training process. The target region is much smaller than the background region in the whole image, thus the negative sample space is verge large. If such extremely unbalanced sample is directly used to train the model, the model may tend to divide all the samples into negative samples, and it is difficult to converge during training. To solve such problem, the hard negative mining strategy is introduced to the training process in this paper. The negative sample obtained from the first training process form the hard negative samples set, and the set is used to retrain for the new model. The proposal region whose IoU is less than 0.5 and objectness score is larger than 0.7 is regards as hard negative sample.

### *B. Sharing Features Training*

The RPN and Fast R-CNN trained independently, which will result in the inability to share convolutional layers. Thus, an alternating training method is used in this paper.

Step 1: Initialization. The RPN is trained independently.

Step 2: The region proposal generated by the RPN network in Step 1 is used to train the Fast R-CNN. At this point, RPN and Fast R-CNN are not shared at all.

Step 3: The Fast R-CNN in Step 2 is used to Initialize a new RPN, but we set the learning rate of the convolutional layers shared by RPN and Fast-RCNN as 0 (that is, it doesn't update) and only update the unique layer of RPN. Now the two networks shared all common convolutional layers.

Step 4: The shared convolutional layers are still fixed, and the unique layer of Fast R-CNN is fine-tuned to form a unified network.

### *C. Model Training and Model Testing*

Model training mainly consist of data processing and parameter optimized, and model testing mainly test the effectiveness the trained model using the test samples. The specific training process is as follows.

1) Obtaining the PASCAL VOC dataset and converting the PASCAL VOC dataset into the VOC2007.

2) The ZF model is used as the convolution feature extraction model, and the model is trained using the processed data set in Step 1.

3) The same dataset is used to test the trained model in Step2 to generate the hard negative samples.

4) The hard negative samples are used to retrain the model to improve the classification performance of the model.

5) The model is fine-tuned to get the final model for testing.

Throughout the whole training process, RPN and R-CNN are alternately trained. Thus, they can share the convolutional layer to form a unified deep convolutional neural network. This can improve the computational efficiency and achieve real-time detection.

## IV. Simulation

### A. Dataset

The template is designed so that author affiliations are not repeated each time for multiple authors of the same affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization). This template was designed for two affiliations.

### B. Performance Metric

In order to evaluate the proposed algorithm, recall rate and precision rate are used, and they are definite as follows:

$$recall = \frac{TP}{TP + FN} \tag{8}$$

$$precsion = \frac{TP}{TP + FP} \tag{9}$$

where TP is the number of the positive sample which is predicted as positive sample by the algorithm, FP is the number of the negative sample which is predicted as positive sample by the algorithm, and FN is the number of the positive sample which is predicted as negative sample by the algorithm.

### C. Results and analysis

The YOLO [14], SSD [15], RFCN [16], Faster R-CNN [17] and the improved Faster R-CNN proposed in this paper are used to detect the images in test set, and the results are shown in Fig. 2. As we can see that YOLO which turns the object detection into a regression problem has highest detection speed due to the simple model structure, but its precision rate is lower than other methods. SSD which combines the YOLO and Faster R-CNN has higher precision rate than YOLO, and higher detection speed than YOLO. RFCN which uses the Faster R-CNN architecture and only contains a convolutional network has higher detection speed and precision rate than Faster R-CNN. The proposed algorithm in this paper which introduces sharing features training and hard negative sample mining into Faster R-CNN has slightly inferior to other methods in terms of detection speed, but it has great advantages in precision rate.

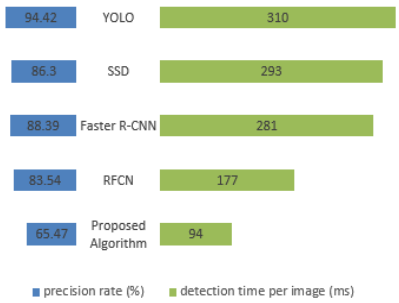


Fig. 4. The performance comparison of different algorithms

In order to further verify the performance of hard negative sample strategy in the proposed algorithm, two strategies (with hard negative sample strategy denoted by strategy 1 and without hard negative sample strategy denoted by strategy 2) are used to train the model. The results after training by two strategies are shown in Figure X. It can be seen that hard negative sample strategy enhances the classification capacity of the model, and the precision rate is improved about 2 percent.

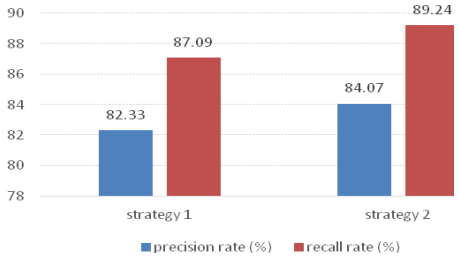


Fig. 5. The performance of hard negative sample strategy

## V. Conclusion

This paper combines feature sharing training and hard negative sample strategies to improve the current state-of-the-art technology Faster R-CNN. Hard negative samples obtained from the first training process is fed into the network again to retrain for the new model. In addition, the alternating training method is used to make RPN and Fast R-CNN in Faster R-CNN share convolutional layers. The simulation result show that the proposed algorithm is slightly inferior to other methods such as YOLO, SSD, RFCN in terms of detection speed, but it has great advantages in terms of detection accuracy.

## References

[1] Lowe, David G, "Distinctive image features from scale-invariant keypoints," international journal of computer vision, vol. 60, no. 2, pp. 91-110, 2004.

[2] H. Bay, A. Ess, T. Tuytelaars, et al, “Speeded-Up robust features,” Computer Vision and Image Understanding (CVIU), vol. 110, no, 3, pp. 404-417, 2006.

[3] P. Viola, M. Jones, “Rapid object detection using a boosted cascade of simple features, ” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 511–518, 2001.

[4] Dalal, Navneet, and Bill Triggs, "Histograms of oriented gradients for human detection," in Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 886-893, 2005.

[5] Ojala, Timo, Matti Pietikainen, and Topi Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Transactions on pattern analysis and machine intelligence, vol. 24, no. 7, pp. 971-987, 2002.

[6] W. Zheng, L. Liang, “Fast car detection using image strip features,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.2703–2710, 2009.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580–587, 2014.

[8] R. Girshick, “Fast R-CNN,” in Proc. of IEEE Conference on Computer Vision, pp. 1440–1448, 2015.

[9] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards realtime object detection with region proposal networks,” IEEE Transactions on pattern analysis and machine intelligence, vol. 39, no. 6, pp. 1137-1149, 2017.

[10] Roh, Myung-Cheol, and Ju-young Lee, “Refining faster-RCNN for accurate object detection,” in Proc. of Fifteenth IAPR International Conference on Machine Vision Applications (MVA), pp. 514-517, 2017.

[11] Han, Guangxing, Xuan Zhang, and Chongrong Li, “Revisiting Faster R-CNN: A Deeper Look at Region Proposal Network,” in Proc. of International Conference on Neural Information Processing, Springer, Cham, pp. 14-24, 2017.

[12] Wu, Wenqi, et al, “Face Detection With Different Scales Based on Faster R-CNN,” IEEE transactions on cybernetics, vol. 99, pp. 1-12, 2018.

[13] Zhang, Zhuo, et al, “Faster R-CNN for Small Traffic Sign Detection,” CCF Chinese Conference on Computer Vision, Springer, Singapore, pp. 155-165, 2017.

[14] REDMON J, DIVVALA S, GIRSHICK R, et al, “You only look once: unified, real-time object detection,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, 2016.

[15] LIU W, ANGUELOV D, ERHAN D, et al, “SSD: single shot multibox detector,” in Proc. of the 2016 European Conference on Computer Vision, Springe, pp. 21-37, 2016.

[16] DAI J, LI Y, HE K, et al, “R-FCN: object detection via region-based fully convolutional networks”, in Advances in neural information processing system, pp. 379-387, 2016.

[17] HE K, ZHANG X, REN S, et al, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no., 9, pp. 1904-1916, 2015.