Study Of Object Detection Based On Faster R-CNN

BIN LIU, Wencang ZHAO and Qiaoqiao SUN
College of Automation & Electronic Engineering
Qingdao University of Science and Technology
Qingdao, China

***Abstract*—Faster R-CNN (R corresponds to "Region") which combined the RPN network and the Fast R-CNN network is one of the best ways to object detection of R-CNN series based on deep learning. The proposal obtained by RPN is directly connected to the ROI Pooling layer, which is a framework for CNN to achieve end-to-end object detection. The feasibility of Faster R-CNN implementation of ResNet101 network and PVANET network is discussed based on the implementation of Faster R-CNN in VGG16 network. Different Faster R-CNN models can be obtained by training with deep learning framework of Caffe. A better model can be obtained by comparing the experimental results using mean average precision (mAP) as an evaluation index. Numerical results show that Faster R-CNN trained by PVANET network obtained the highest mAP.**

Keywords—Faster R-CNN; Fast R-CNN; object dection; RPN

I. INTRODUCTION

Faster R-CNN has gone through the development process from R-CNN, Fast-RCNN and Faster R-CNN. Firstly, about 2000 Region Proposals are extracted from top to bottom in the image by R-CNN using the selective search algorithm [1-2]. Each Region Proposal is scaled to 227 * 227 as the CNN input. The output of the fc7 layer is used as a feature. Features which were extracted by each Region Proposal are input to the SVM to classify. The classified Region Proposal from SVM will be done border regression with the Bounding box regression value to correct the original suggested window and generating predictive window coordinates. The defects of R-CNN are multiple stages training, the steps cumbersome, training time-consuming, and taking up disk space. VGG16 model requires about 47s to deal with an image by using GPU. The test speed of R-CNN is slow, and each candidate area needs to run the entire forward CNN calculation, and CNN features are not updated during SVM and regression.

Secondly, the selective search algorithm is used to extract about 2000 Region Proposals by Fast R-CNN from top to bottom in the image [3]. The whole picture is input to CNN to extract feature. The proposed window is mapped to the last layer of the CNN convolution of feature map, so that each proposed window generates a fixed size of the feature map through the Roi Pooling layer, and uses Softmax Loss and Smooth L1 Loss to combine the classification probability with Bounding box regression to train together. Compared with R-CNN, the mainly different of Fast R-CNN that is Roi Pooling layer is added to the last layer of the convolutional layer, and the multi-task loss function is used to join the Bounding box regression to the CNN network training directly [4].

The image is normalized by Fast R-CNN directly, and the proposed box information is added in feature map which is output by the final convolution layer (conv), so that operation can be shared before the CNN which made the test speed [5]. Fast R-CNN only need to send features and suggested areas which are extracted one-time into the network in training, and training data in the GPU memory directly is sent into the layer, so that the candidate area of the first few layers of features do not need to repeat the calculation and no longer need to store a lot of data on the hard disk, making training faster. In addition, the class judgment and the location regression were realized in unison by using the depth of the network by the Fast R-CNN, no longer need additional storage.

The remainder of the paper is organized as follows: Faster R -CNN network is overviewed in Section II; object detection based on Fast–R-CNN is introduced in Section III; some experimental results are contained in Section IV, and Section V concludes the paper.

II. OVERVIEW OF FASTER R-CNN

After the accumulation of R-CNN and Fast R-CNN, proposed Faster R-CNN was proposed by Ross B. Girshick in 2016. Faster R-CNN has integrated feature extraction, proposal extraction, rectification in a network as shown in Fig. 1 [6]. The overall performance is greatly improved especially in terms of detection speed. Faster R-CNN creatively uses the convolution network to generate the proposed box and shares the convolution network with the object detection network which reduces the number of proposed frames from about 2000 to about 300 [7]. The quality of the proposed box is also improved.

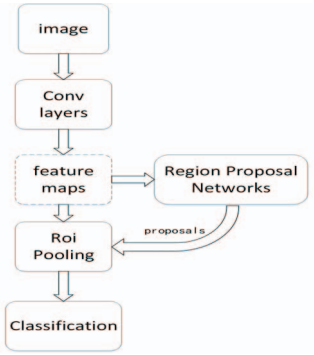

Fig. 1. Faster R-CNN structure

This work was supported in part by the Chinese State Scholarship Fund (Grant No. 201608370049); National Natural Science Foundation of China (Grant No. 61171131) and Science and Technology Development Project of Shandong (Grant No. 2013YD01033).

Compared to Fast R-CNN, RPN (Region Proposal Network) is instead of the original Selective Search method to generate the suggestion window by Faster R-CNN, and the CNN of object detection shares with The CNN of suggestion window.

The structure of Faster R-CNN can be divided into four main contents:

A. Convolution layers.

As one of the object detection methods, firstly, image feature maps are extracted by a set of basic conv with relu and pooling layers by Faster R-CNN [10]. The feature maps are shared for subsequent RPN layers and full connected layers. CNN contains convolution layer and pooling layer [11]. In the python version of the VGG16 model of the network structure, for example, the part of Convolution layers contains a total of 13 convolution layers with activation function of relu and 4 pooling layers. The parameter are set as follows: (1). All of the credibility layers are: kernel_size = 3, pad = 1. (2). All pooling layers are: kernel_size = 2, stride = 2. In the Faster R-CNN convolution layers, each convolution is done by padding (pad = 1, which is filled with a circle 0), resulting in the original image becoming (M+2) × (N+2) size, the size of image becomes M×N after 3×3 convolution. The benefits of such setting parameters are that the size of the input and output matrix in convolution layers don't be changed [8-9]. The pooling layers in Convolution layers set kernel_size = 2, stride = 2. Thus each M × N matrix becomes (M/2)*(N/2) size through the pooling layers. In summary, in the convolution layers, convolution layers and relu layers do not change the input and the output. Only the pooling layers make the output length and width become the 1/2 of input. Then, an M×N-sized matrix is fixed by convolution layers to the size of (M/16) × (N/16) so that feature map generated by the convolution layers can be mapped with the original map.

B. Region Proposal Networks

The regions proposals can be generated by the RPN network, which determines anchors belong to foreground or background through the softmax. And accurate proposals can be gotten by using bounding box regression correction anchors.

The classical detection method which generates a detection frame consumes time very much, such as adaboost of OpenCV, using a sliding window with image pyramid to generate a detection frame or R-CNN, using the SS (Selective Search) method to generate a detection box. However, Faster R-CNN abandons the traditional sliding window and SS method, and uses RPN directly generate detection box, which is a huge advantage which can greatly enhance the generation speed of detection frame.

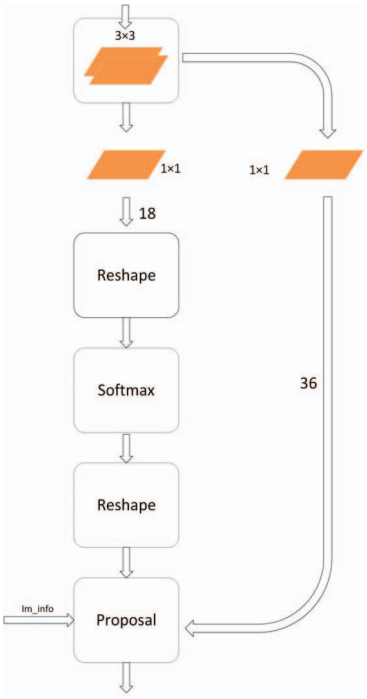

Fig. 2. RPN network structure

Fig. 2 shows the specific structure of the RPN network. It can be seen that the RPN network actually has two branches: the first branch gets the foreground and background by the softmax classification anchor, and the second branch is used to calculate the bounding box regression offset for anchors to get accurate of the proposals. And the final proposal layer is responsible for the synthesis of foreground anchors and bounding box regression offset to get proposals, while excludes too small and beyond the boundaries of the proposals.

C. Roi Pooling.

The input feature maps and proposals can be collected through roi pooling layer. The proposal feature maps can be extracted after synthesizing the information, which is sent to the subsequent full connected layer to determine the object category.

D. Classification

Classification layers use the proposal feature maps to calculate the proposal's class, and bounding box regression to get the final exact position of the checkbox [12]. Classification layers get the 7×7 = 49 size of the proposal feature maps from the Roi Pooling layers , and calculate which category each proposal and specifically judge of belonging to[13] (such as people, cars, horses, etc.) through the full connected layer with softmax and a output class probability vector can be obtained. Classification layers use the bounding box regression again to get the position offset bbox_pred for each proposal, and return a more accurate target detection box. To obtain a more accurate rect box, part of the network structure of classification layers is shown in Fig. 3.

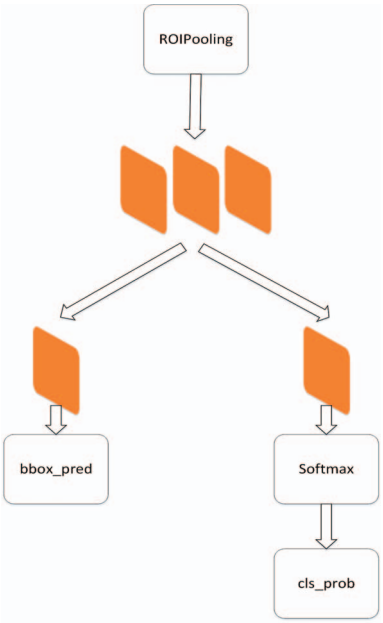

Fig. 3. Part of the network structure of classification layer

III. Object Detection Based On Faster R-CNN

The VGG16 network is began to train the Faster R-CNN in this paper, which is scaled to a fixed size M×N for an arbitrary size P×Q image, and then the M×N image are sent to the network, then extracts the feature maps through the 13 conv layers which the activation function are all relu, and 4 pooling layers. The RPN network firstly passes the 3x3 convolution and generates the foreground anchors and bounding box regression respectively, and then calculates the proposal. The Roi Pooling layer uses the proposals from the feature Maps to extract the proposal feature into the subsequent full connection and softmax network for classification [14] .

In this experiment, the deep learning framework of Caffe was used to train the network. In the experiment process, the Faster R-CNN training network of ResNet101 and PVANET are obtained by replacing Conv layers and fine tuning the RPN network layer. We use PASCAL VOC2007 [15-16] as train set and test set, and all of evaluations were done on Intel i7-7700K CPU and NVIDIA 1080TI GPU. Test results are shown in Table 1, Table 2 and Fig. 7.

IV. Experimental Results

The training model and the following classification results of mean average precision (mAP) are obtained by studying the Faster R-CNN of different networks. The examples of model to object detection are shown in Fig.4 to Fig. 6. The object's classification results and probability are identified by each chart. The mean average precision of object detection of the three methods is shown in Fig. 7.

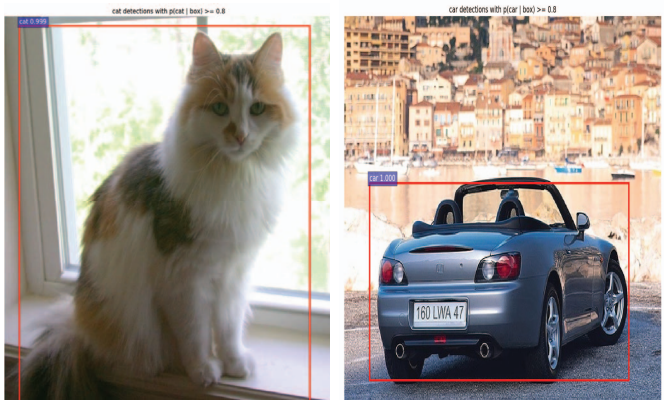

Fig. 4. The probability of object detection

The detection probability of the cat in Fig. 4 (left) is 99.9% and the detection probability of the car in Fig. 4 (right) is 100%.

Fig. 5. The probability of object detection

The detection probability of the person in Fig. 5 (left) is 89%,99.8% and 99.2%. The detection probability of the person in Fig. 5 (right) is 99.1%

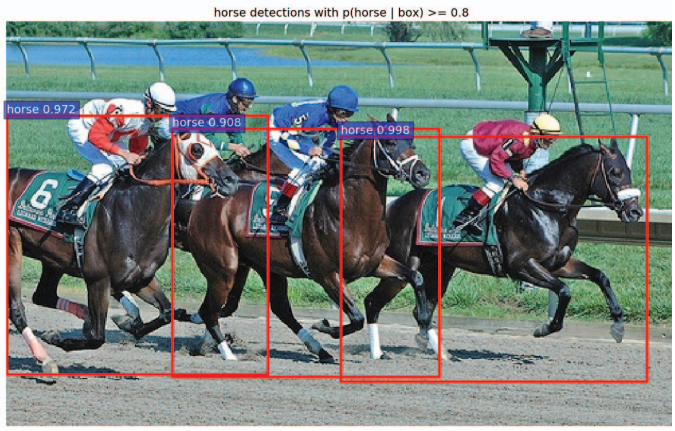

Fig. 6. The probability of object detection

The detection probability of the horse in Fig. 6 is 97.2% ,90.8% and 99.8% in turn.

TABLE I. TEST RESULTS BASED ON VOC2007 DATA SET

Models	mAP(%)
Faster R-CNN + VGG16	69.4
Faster R-CNN + ResNet101	72.5
Faster R-CNN + PVANET	84.9

TABLE II. AVERAGE ACCURACY OF EACH CATEGORY OF TEST BASED ON THE VOC2007 DATA SET

NO	Models / Category AP	VGG16	ResNet101	PVANET
1	aeroplane	0.691	0.733	0.887
2	bicycle	0.783	0.754	0.883
3	bird	0.674	0.761	0.858
4	boat	0.546	0.559	0.811
5	bottle	0.526	0.561	0.745
6	bus	0.764	0.796	0.899
7	car	0.799	0.788	0.893
8	cat	0.797	0.838	0.897
9	chair	0.494	0.569	0.745
10	cow	0.753	0.835	0.887
11	diningtable	0.697	0.644	0.832
12	dog	0.777	0.841	0.885
13	horse	0.811	0.845	0.881
14	motorbike	0.751	0.737	0.893
15	person	0.773	0.759	0.874
16	pottedplant	0.410	0.464	0.638
17	sheep	0.682	0.751	0.874
18	sofa	0.672	0.737	0.861
19	train	0.742	0.766	0.894
20	twmonitor	0.736	0.765	0.842

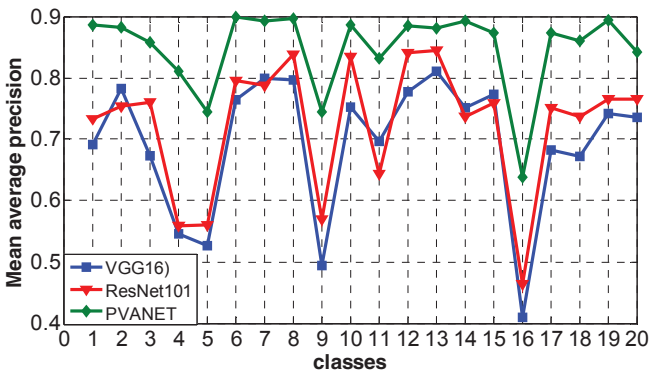

Fig. 7. The mean average precision of object detection

V. CONCLUSION

In this paper, different neural networks have been demonstrated to achieve classification using the Faster R-CNN. Experimental results show that its effectiveness comes from the convolutional layers and region proposal network (RPN) module. Providing research ideas for Faster R-CNN is the purpose of this paper, although the mean average precision result is not high. The foundation is laid for the follow-up to the direction of further research.

REFERENCES.

[1] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun, "HyperNet: Towards accurate region proposal generation and joint object detection," In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[2] Lenc K, Vedaldi A, "R-CNN minus R," Computer Science, 2015.

[3] R. B. Girshick. Fast R-CNN. In ICCV, pp. 1440–1448, 2015.

[4] Dai J, Li Y, He K, et al, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," 2016.

[5] Wang X, Shrivastava A, Gupta A, "A-Fast-RCNN: Hard Positive Generation via Adversary for Object Detection," 2017.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Trans. Pattern Anal. Mach. Intell, 2016.

[7] Salvador A, Giró-I-Nieto X, Marqués F, et al, "Faster R-CNN Features for Instance Search," IEEE Conference on Computer Vision and Pattern Recognition Workshops. IEEE Computer Society, 394-401, 2016.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[9] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.

[10] Sean Bell, C. Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[11] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[13] He K, Zhang X, Ren S, et al, "Deep Residual Learning for Image Recognition," pp. 770-778, 2015.

[14] Krizhevsky A, Sutskever I, Hinton G E, "ImageNet classification with deep convolutional neural networks," International Conference on Neural Information Processing Systems. Curran Associates Inc, 1097-1105, 2012.

[15] Hong S, Roh B, Kim K H, et al, "PVANet: Lightweight Deep Neural Networks for Real-time Object Detection," 2016.

[16] Everingham M, Winn J, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Development Kit," International Journal of Computer Vision, vol. 111, pp. 98-136, 2015.