DCRN: Densely Connected Refinement Network for Object Detection

Shihui Gao*, Zhenjiang Miao, Qiang Zhang and Qingyu Li

Institute of Information Science, School of Computer and Information Technology, Beijing Jiaotong University, Haidian District, 100044, Beijing, China

*Corresponding author’s e-mail address: 17125156@bjtu.edu.cn

Abstract. Object detection has made great progress in recent years, the two-stage approach achieves high accuracy and the one-stage approach achieves high efficiency. In order to inherit the advantages of both while improving detection performance, this manuscript presents a useful method, named Densely Connected Refinement Network (DCRN). It adds the dense connection based on RefineDet. Compare to the RefineDet, our approach can take full advantage of the bottom feature information. DCRN is formed by three interconnected modules, the dense anchor refinement module (DARM), the dense object detection module (DODM) and the dense transfer connection block (DTCB). The former module can make better use of the features from different layers to initially adjust anchors by attaching dense connection. The latter module takes the refined anchors to further improve the regression and predict multi-class label. Due to the dense connection in DCRN, the network parameters are reduced and the computing costs of this approach are also saved. Extensive experimental results on PASCAL VOC 2007 and PASCAL VOC 2012 demonstrate that DCRN achieves higher accuracy than the one-stage method and higher efficiency than the two-stage method.

1. Introduction

For object detection in images, there are mainly two types of approaches. The two-stage approaches including R-CNN [1], Fast R-CNN [2] and Faster R-CNN [3] have the advantage of high accuracy. However, it has not achieved well efficiency because of the computational complexity. And the one-stage approaches including YOLO [4] and SSD [5] have been achieving high efficiency, although accurate, their accuracy still behind the former. Therefore, both approaches have their own limitations. The former cannot achieve high efficiency due to the complexity of calculations, so it cannot be applied to real-time object detection machines. The latter is the opposite.

To overcome the shortcomings, we put DCRN which inherits the advantages of the two-stage and the one-stage. The main framework of this manuscript contains three modules, named DARM, DODM, DTCB. The design of the former is similar to the SSD [5]. DARM takes advantage of multi-layer features, and generates bounding boxes, and makes roughly adjustments. Furthermore, DRCN introduces the dense connection to the module. Thanks to the dense block, the module can make more full use of the feature from multi-layer. DODM takes the anchors generated by DARM as input, and then make further multi-class classification and regression. Thanks to the dense connection structure, DCRN reduces the parameters which greatly save the calculation cost. So DCRN achieves higher accuracy than the RefineDet [6] while maintaining the well efficiency. Therefore, our approach can be easily ported to real-time detection devices. DCRN has achieved a good performance on the PASCAL
VOC 2007 [7] and PASCAL VOC 2012 dataset [7].

Figure 1: Architecture of DCRN. The C in the circle denotes the concatenation operation. The light colored squares above represent the feature maps of different scales in the DARM module, and the dark squares below represent the feature maps of different scales in the DODM module.

2. Related Work

Traditional object detection methods such as Scale-invariant Feature Transform (SIFT) [8], Histogram of Oriented Gradient (HOG) [9], Support Vector Machine (SVM) [10], AdaBoost [11], Deformable Parts Model (DPM) [12] have many limitations with the reason of the Manual design, so these methods are not widely used in technology. Thanks to the emergency of Convolutional neural network (CNN) [13], it acts in a quite different way from traditional methods. The mainly contribution of CNN [13] is that the network have a very good performance in the process of extracting features.

2.1. Network architectures for detection

Figure 2: Dense block [18]. $x_t = h_t([x_0, x_1, x_2, x_{t-1}])$. $x_0$ is the input of $h_1$, $x_0$ and $x_1$ are the input of $h_2$, $x_1$ is the output of $h_1$. 
With deeper architectures, the approach can learn more complex features than the shallow ones. Many networks have emerged, such as AlexNet [14], VGGNet [15], GoogLeNet [16], ResNet [17], DenseNet [18] and Mobilenet [19]. Meanwhile, several regularization techniques have also been proposed to further enhance the model capabilities. And most detection methods directly utilize pre-trained ImageNet models as the backbone network.

2.2. Object detection

The approaches of detection can be parted two categories, i.e., the two-stage and one-stage approach. The process of the two-stage is divided into two steps. First, the network generates a lot of candidate boxes, second, the images are accurately located and classified. These representative methods (e.g., R-CNN [1], Fast R-CNN [8], Faster R-CNN [3] to SPPnet [20]) achieve increasingly high accuracy on many challenging dataset (e.g., PASCAL VOC 2007, PASCAL VOC 2012 [7] and MS COCO [16]). While maintaining high accuracy, we also need to consider the efficiency of detection, without a doubt, for one-stage, it does not need to generate the region proposals, and it directly obtains the class confidence and location of the objects. After a single test, the final detection result can be directly obtained, so that the detection speed is fast, such as YOLO [4], SSD [5], DSSD [21] and DSOD [22]. Extensive experiments demonstrate that although the later detectors have made good progress, their accuracy still behind the former methods.

3. DCRN

Based on a feedforward convolutional network, the DCRN produces a fixed number of bounding boxes, then scores the categories for each bounding box, finally, we use the non-maximum suppression [6] to produce the final test results. This manuscript will introduce the three modules of the DCRN. As shown in Figure 1.

3.1. Dense anchor refinement module

Similar to the RPN network in the Faster R-CNN [3], DARM generates bounding boxes based on four layers of features, at last two branches obtained, one is the coordinate regression branch of bounding boxes, and the other is the second classification branch of bounding boxes, removes some negative samples can reduce the search space of the classifier. On the other hand, the position and size of the anchor can be modified to make it more conducive to the regression of coordinates, providing a better initial regression for the follow-up. Figure 2 shows the Dense Blocks [18], each layer has a dense connection to the previous layer. Assuming a network with an $M$ layers, the number of highway dense connections is as follows.

$$C = \frac{M \times (M + 1)}{2}$$

Where $C$ is the number of dense connections in the dense block, $M$ is the number of layers.

3.2. Dense object detection module

![Figure 3: Dense Transfer Connection Block.](image-url)
DODM uses the refined anchors generated by DARM as input, then it can make label predictions about different classes and locations. DODM uses the refined anchors generated by DARM as input and make concatenation operations on multi-layer features. This includes the application of Dense block [18], as is shown in Figure 2, which can make full use of the features of all layers to further improve the prediction results. Then the softmax classification is performed, and finally we use the smooth L1 regression to find the location of objects.

3.3 Dense transfer connection block

Figure 3 shows the structure of DTCB, it mainly performs the feature conversion operation, which converts the output feature map of the DARM module into the input of the DODM module. The module uses deconv to enlarge the feature map, next adds them using element-wise sum operation, then connects a convolution to increase the resolution of the feature map. The upper layer of the module is the input of the ARM. Performing the element-wise sum on the layers which are operated by twice 3×3 conv and the features of the deep deconv, then it passes the results into the DODM module.

3.4 Loss function

The loss function of our method contains the loss in DARM and the loss in DODM. We define it as:

\[
\mathcal{L}(p_i, x_i, c_i, t_i) = \frac{1}{N_{\text{d Carm}}} \left( \sum_i \mathcal{L}_b(p_i, [l_i^* ≥ 1]) + \sum_i \left[ l_i^* ≥ 1 \right] \mathcal{L}_r(x_i, g_i^*) \right) \\
+ \frac{1}{N_{\text{d Dodm}}} \left( \sum_i \mathcal{L}_m(c_i, l_i^*) + \sum_i \left[ l_i^* ≥ 1 \right] \mathcal{L}_r(t_i, g_i^*) \right)
\]

(2)

Where \(i\) is the index of anchors in a mini-batch, \(p_i\) and \(x_i\) are the predicted confidence of the anchor \(i\) in the DARM, \(c_i\) and \(t_i\) are the predicted object class in the DODM. \(l_i^*\) is the class label of the ground truth, \(g_i^*\) is the location and size of ground truth. \(N_{\text{d Carm}}\) and \(N_{\text{d Dodm}}\) are the number of positive anchors in the DARM and DODM module. The multi-class classification loss \(\mathcal{L}_m\) is the softmax loss, and the regression loss \(\mathcal{L}_r\) is the smooth L1 loss [6]. \(\mathcal{L}_b\) is the binary classification loss [6], it is the cross-entropy loss over two class. \(\mathcal{L}_m\) is the multi-class classification loss [6], it is the softmax loss over different classes confidence.

4. Experiment Details and Results

4.1 Dataset and experiment

Our experiments are done on the widely used PASCAL VOC 2007, PASCAL VOC 2012 [7] dataset which have 20, 20 objects categories respectively. We use mean Average Precision (mAP) to measure the performance of the object detection. We implement DCRN in Mxnet. Our experiments are all based on VGG16 [15], we fine-tune the resulting model using SGD with initial learning rate \(10^{-3}\), 0.9 momentum, 0.0005 weight decay and batch size 16. All modules are trained on the PASCAL VOC 2007 and PASCAL VOC 2012. This manuscript compares DCRN with other object detection methods in Table 1. All experiments result of the Table 1 are conducted on the servers on our lab. We carry the experiment with a NVIDIA GeForce RTX 2080Ti graphi card and the memory is 11G.

| Model     | Data       | Backbone  | Input size    | Speed (FPS) | Map (%) |
|-----------|------------|-----------|---------------|-------------|---------|
| Fast R-CNN [2] | 07+12     | VGG-16    | ~1000 × 600   | 0.5         | 65.5    |
| Faster R-CNN [3] | 07+12     | VGG-16    | ~1000 × 600   | 7           | 67.3    |
| YOLO [4] | 07+12     | GoogleNet | 448 × 448     | 45          | 56.2    |
| SSD300 [5] | 07+12     | VGG-16    | 300 × 300     | 46          | 73.1    |
| DSOD [22] | 07+12     | DS/64-192-48-1 | 300 × 300 | 20.6        | 74.0    |
| Reffinedet [6] | 07+12    | VGG-16    | 512 × 512     | 24.1        | 77.4    |
| DCRN     | 07+12     | VGG-16    | 512 × 512     | 25.7        | 78.1    |

Table 1: Result on PASCAL VOC 2007 and PASCAL VOC 2012
4.2. Results and Analysis

From Table 1 we can see that the two-stage approaches (e.g., Fast R-CNN [2], Faster R-CNN [3]) achieve higher accuracy than YOLO [4], but the one-stage approaches (e.g., YOLO [4], SSD [5], DSOD [22]) are far more efficient than the two-stage approaches. As baseline, the Refinedet [6] has great performance on both accuracy and efficiency. Compared to other one-stage and two-stage, DCRN improves accuracy while improving efficiency. Mainly thanks to the design of dense connection and three inter-connected modules, which enable DCRN to achieve the highest accuracy among these approaches. Although the YOLO [4] and SSD [5] are faster than our DCRN, their accuracy are 21.9% and 5% lower than ours. Moreover, to compare the Refinedet [6] with our DCRN, we set a series of identical parameters for the two methods, such as the backbone, the input size, the dataset. The result indicates that DCRN has increased both the accuracy and efficiency by adding dense connection block, the main reason is that the transition layer in the dense connection has a parameter indicating how much the output is reduced to the original. Generally, the default is 0.5, so the number of channels will be reduced by half when passed to the next dense block. Therefore, the amount of computation is saved by reduce the parameters. In summary, the DCRN improves the shortcomings of the previous method and maintains the advantages.

5. Conclusion

In the paper, we introduce a method based on the Refinedet [6], named DCRN. Our approach inherits the advantages of the one-stage methods and the two-stage methods, and introduces the idea of dense connection from the DenseNet [18]. DCRN not only increases the accuracy of object detection but also improves the efficiency by adding these dense connection block to the DARM module and the DODM module. This is a great process in the object detection. Next, our research will focus on the attention mechanism. I will apply the attention mechanism to our DCRN for object detection and get better result in the future.

References

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014. 1, 2, 3
[2] R. Girshick. Fast r-cnn. In ICCV, 2015. 1, 2, 3
[3] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In NIPS, 2015. 1, 2, 3, 6, 7
[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In CVPR, 2016. 1, 2, 3
[5] W. Liu, D. Anguelov, D. Erhan, et al. Ssd: Single shot multibox detector. In ECCV, 2016. 1, 2, 3, 4, 6, 7
[6] Zhang S, Wen L, Bian X, et al. Single-Shot Refinement Neural Network for Object Detection[J]. 2017.1.3
[7] M.Everingham,L.J.V.Gool,C.K.I.Williams,J.M.Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. IJCV, 88(2):303–338, 2010. 1, 3
[8] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. J. of Comput. Vision, vol. 60, no. 2, pp. 91–110, 2004.
[9] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in CVPR, 2005.
[10] C. Cortes and V. Vapnik, “Support vector machine,” Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.
[11] Y. Freund and R. E. Schapire, “A desicion-theoretic generalization of on-line learning and an application to boosting,” J. of Comput. & Sys. Sci., vol. 13, no. 5, pp. 663–671, 1997.
[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, pp. 1627–1645, 2010.
[13] hao Z Q, Zheng P, Xu S T, et al. Object Detection with Deep Learning: A Review[J]. 2018.7.15
[14] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012. 1, 3
[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015. 1, 3
[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, et al. Going deeper with convolutions. In CVPR, 2015. 1, 3, 4
[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016. 1, 3, 4
[18] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In CVPR, 2017. 1, 2, 3, 4
[19] Howard A G, Zhu M, Chen B, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. 2017.1.17
[20] Iandola F N, Han S, Moskewicz M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size[J]. 2016.4.6
[21] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. DSSD: Deconvolutional single shot detector. CoRR, abs/1701.06659, 2017. 3, 5, 6, 7
[22] Z. Shen, Z. Liu, J. Li, Y. Jiang, Y. Chen, and X. Xue. DSOD: learning deeply supervised object detectors from scratch. In ICCV, 2017. 3, 6, 8