2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506038

REINFORCEDET: OBJECT DETECTION BY INTEGRATING REINFORCEMENT LEARNING WITH DECOUPLED PIPELINE

Man Zhou[1,2] *Liu Liu*[3,†] *Rujing Wang*[2]

[1] University of Science and Technology of China, P.R. China
[2] Institute of Intelligent Machines, and Hefei Institute of Physical Science, Chinese Academy of Sciences, Hefei 230031, China
[3] Department of Computer Science, Shanghai JiaoTong University, P.R. China
[†] Corresponding Author

ABSTRACT

Recent object detection methods largely rely on numerous pre-defined anchors that suffer from huge computational cost and resource consumption. To solve this issue, we propose a low-memory deep reinforcement learning based anchor-free object detection approach, namely ReinforceDet, which computes few but accurate region proposals for detection. Specifically, the extracted feature maps are fed into a reinforcement learning network to localize objects as initial region proposals with our re-designed reward function and then adopt another neural network to refine them. To speed up this process in test phase, we decouple the two-branch CNN networks as light-head cascaded subnetworks, named IoU-net and bounding box net. Experimental results show that ReinforceDet could obtain the state-of-the-art performance with much lower compitational and memory cost.

***Index Terms*—** Reinforcement Learning, Object Detection, Decoupled Pipeline

1. INTRODUCTION

Object detection is a crucial computer vision task and has garnered much attention in recent years. Current object detection methods use Convolutional Neural Network (CNN) to densely predict each instance's category and localization with numerous pre-defined anchors, which leads to excessive computational consumption [1, 2, 3, 4]. In contrast, Reinforcement Learning (RL) based methods [5, 6, 7] usually require much fewer region proposals to search the location by training an agent with the sequential interaction. Nevertheless, these methods might not acheieve state-of-the-art performance due to unappropriated state representation and lack of action for box regression.

In this paper, we propose a novel architecture namely ReinforceDet for active object detection that adopts deep RL algorithm to obtain few and accurate region proposals and CNN for bounding box refinement. During network training, a method based on imitation for RL and an reasonable reward function for RL optimization. Specifically, we re-design the reward function that considers both IoU and its disparity between two adjacent actions while current RL methods only concentrate on variation of IoU. Moreover, to speed up the training, we propose an apprenticeship learning [8] strategy using prior computed ground truth.

Another problem in current methods lies on RL strategy might not handle bounding box regression procedure since the parameters of pre-defined action space have a marked impact on the result of region refinement on the account of the pre-defined actions not covering the target size space. Here, we propose a parallel branch that unifies action space RL and feature space into a complete network for further regression. During test optimization, the two-branch CNN network is decoupled into two weights-shared subnets, recorded as IoU-net and bounding box net, which are responsible for optimal region proposal selection and downstream regression respectively.

To sum up, the major contributions of this paper are summarized as follows: (1) We design a novel network ReinforceDet that hybridizes reinforcement learning and convolutional neural network for object detection, in which we redesign reward function and utilize apprenticeship learning optimize the training process. (2) A decoupled and weights-sheared transformation for speed testing is implemented to relieve resource consumption during testing phase. (3) Experimental results show the superior performance of ReinforceDet over the current state-of-the-art approaches.

2. RELATED WORK

CNN based object detection The majority of existing object detection approaches on the basis of CNN deep detectors can be approximately categorized into two-stage [2] and single-stage detectors [3, 9, 10, 11]. Two-stage detectors

This work was supported in part by the National Natural Science foundation of China under grant 31671586.

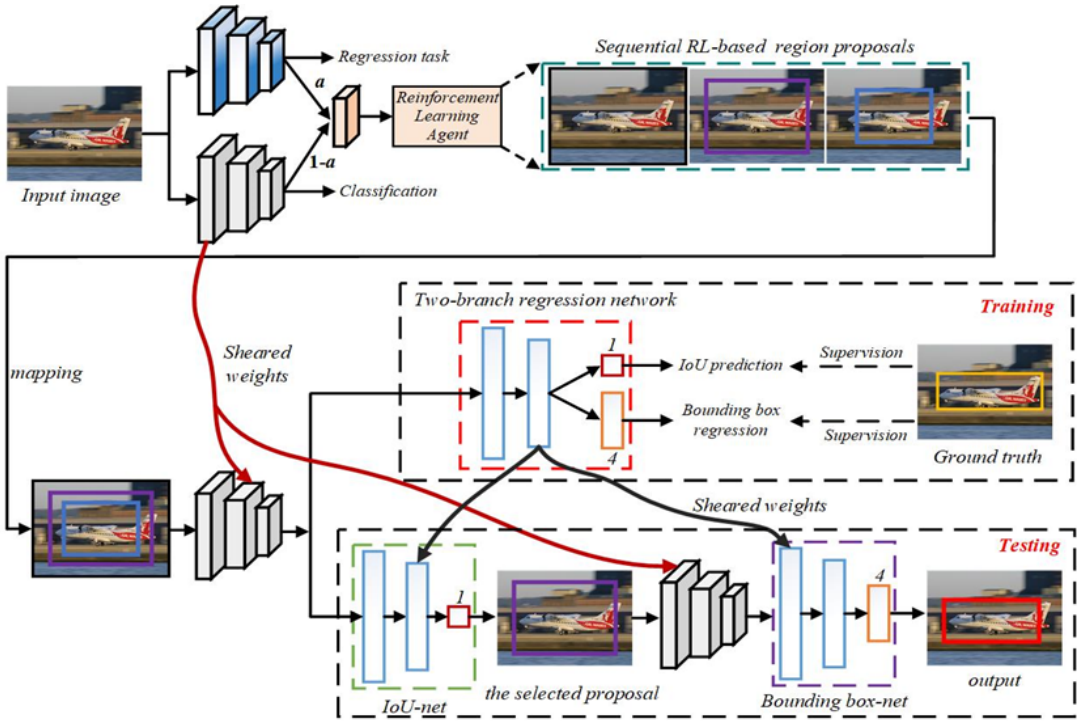

Fig. 1: Pipeline of our ReforncеDet. There are two major components in our method: multi-task training procedure and decoupled low-memory testing procedure.

usually adopt Region Proposal Network (RPN) that provide large number of region proposals and use RoI pooling or RoI Align to enhance the local features for each instance's classification and regression [2, 1, 4]. These methods require to pre-compute tens of thousands of redundant region proposals without taking account of image context. On the contrary, one-stage detectors directly predict objects' categories and locations from the global feature maps, which do not need "crop" each instance's region from original image so they usually could achieve faster inference speed.

Reinforcement learning based Detector RL is another path to object detection task. Caicedo et al. [12] design an active object detection model, which applies Deep Q-learning Network to learn action-decision policy to search target until triggering terminal action, and obtains comparative results with RCNN. Besides, Bueno et al. [13] propose a top-down hierarchical search strategy with five actions, where a trained agent only focuses on regions with adequate object information and then narrows down the local regions for further search. However, the above method only detects a fixed number of objects. To overcome this issue, Yang Li et al. [14] use restricted Edge Boxes to get more appropriate high-quality candidate boxes through the prior knowledge, which achieves high accuracy and recall. Among these methods, they might not consider integrating CNN into RL strategy and jointly learn an object detection model while our ReinforceDet aims to solve this issue towards state-of-the-art detection performance.

3. MATERIALS AND METHODS

ReinforceDet integrates RL and CNN into one joint network at the same time. In this section, we will discuss these two components: Reinforcement Learning Network and two-branch CNN network.

3.1. Reinforcement Learning for Region Proposal

Formally, deep RL training has a set of states $\mathcal{S}$, action space $\mathcal{A}$ and a reward function R. In our method, state $\mathcal{S}$ consists of feature vectors F_f, where F_f is outputted from final layer of feature extraction network (we use VGG16 [15] here). Given F_f, we design 8 actions to build action space $\mathcal{A}$, in which 7 actions indicate the locations of targeted object in input image (upper left, upper right, lower left, lower right, horizontal, vertical, central) and 1 action as stop trigger. The region proposals by our RL method is shown in Fig. 2. During object searching, an inhibition-of-return (IoR) mechanism is used to prevent the same region from being attended again.

A reasonable reward function takes the most essential part in our RL based region proposals. In our work, we redesign the reward function to consider both IoU variations and change magnitude between adjacent regions. We define the IoU variation as $IoU'(S_t, S_{t+1}) = IoU(S_{t+1}, gt) -$

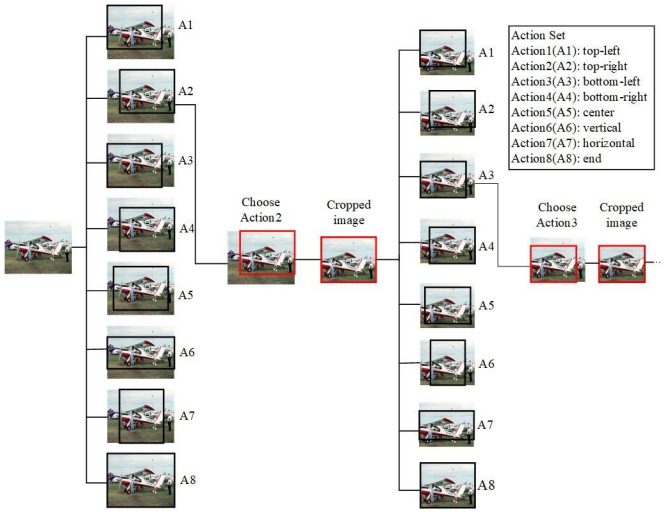

Fig. 2: The region proposals generation process by Reinforcement learning network with pre-defined action space.

$IoU(S_t, gt)$. When $IoU'(S_t, S_{t+1}) > 0$, the reward $R_a(S_t, S_{t+1})$ for action a is:

$$R_a(S_t, S_{t+1}) = r + \beta IoU'(S_t, S_{t+1})^\gamma - \lambda t \quad (1)$$

When $IoU'(S_t, S_{t+1}) \leq 0$:

$$R_a(S_t, S_{t+1}) = -r - \beta |IoU'(S_t, S_{t+1})|^\gamma - \lambda t \quad (2)$$

where r is set to 1, λ is penalty factor to speed training and set as 0.001 and γ is set to 0.5. For the action of end sign, the reward $R_e(S_t, gt)$ is:

$$R_e(S_t, gt) = \begin{cases} \eta & IoU(S_t, gt) \geq \tau \\ -\eta & IoU(S_t, gt) \leq \tau \end{cases} \quad (3)$$

where η and τ are set to 3.0 and 0.5.

In terms of training the RL network, most of reinforcement learning algorithms suffer from unstable and inefficient training procedure in result of the continuity nature. To relieve this issue and speed up the training procedure, we introduce apprenticeship learning in our method. In detail, we adopt a memory replay strategy that is filled with trajectory of region selection process and make up of four parts (viewed as transition [current state, action, reward, transformed state]). Then, using prior greedy strategy, the ground truth is referred to compute the IoU between pre-defined 8 actions and ground truth. Then, the reinforcement learning agent is guided to select the corresponding action with highest IoU to take. To avoid the limitation of prior knowledge, we build two memory replay of 1000 capacity to and implement exploration. One is updated by reinforcement learning agent with random ε-greedy strategy. Another is created before training with apprenticeship learning of prior greedy strategy. During training of DQN, we randomly sample each batch size of 50 tuples to off-line train.

3.2. Two-branch CNN Network for Box Refinement

Given several region proposals by reinforcement learning network, we build two-branch CNN network to refine these regions. Motivated by two-stage object detectors that exploit local features for box regression, we feed the region proposals into two parallel slim CNN networks for confidence prediction and bounding box regression. Specifically, confidence prediction branch is used to select the optimal region proposal from the outputs of reinforcement learning network as final region location, and bounding box regression is designed to refine the box as in [2]. As a result, our proposed network achieves the promising performance with low-memory, which is advantageous to practice application in limited resource.

In test phase of the two-branch network, we decouple the two branches into two sequential weights-sheared subnetworks to further relieve resource consumption as shown in Fig. 3. These two subnetworks are named as IoU-net and bounding box net. To be specific, the region proposals are first fed into IoU-net to select the optimal one with maximal IoU. The selected one is then passed into downstream bounding box-net to regress the object position. In our decoupled pipeline, only one region proposal is processed with bounding box net in testing while all the regions are refined in training phase, so we can further speed up the inference time.

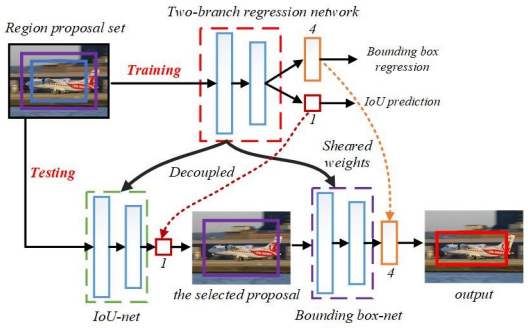

Fig. 3: The decoupled and weights-sheared subnetworks for low-memory testing.

4. EXPERIMENTS

4.1. Experimental Settings

All the experiments in this section are conducted on two widely-used object detection benchmark datasets, i.e., PASCAL VOC 2007 and 2012 [18]. Specifically, we use 50 epochs, as advised by the authors of prior works [13], to train RL agent with ε-greedy policy. Experimental results

Method	Category ID																				mAP
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Caicedo-RL[12]	55.9	61.9	38.4	36.5	21.4	56.5	58.8	55.9	21.4	40.4	46.3	54.2	56.9	55.9	45.7	21.1	47.1	41.5	54.7	51.4	46.1
Hierarchical–RL[13]	28.3	30.1	24.1	20.6	17.3	31.0	28.3	44.4	17.8	15.1	30.0	37.6	33.9	36.0	41.1	19.1	11.3	40.8	38.6	15.7	28.1
Multitask-RL[16]	59.2	62.3	40.2	41.2	24.1	59.5	67.1	55.6	24.1	64.1	50.2	54.1	57.1	54.7	46.5	29.4	48.5	44.2	54.1	35.9	48.6
Tree-RL[17]	71.2	**82.4**	72.0	62.3	50.4	80.0	79.3	83.4	**57.8**	79.3	72.0	82.7	83.3	77.2	**77.2**	**44.4**	76.4	**76.5**	82.2	**71.5**	73.1
ReinforceDet	**76.5**	82.0	**74.8**	**65.2**	**52.3**	**80.8**	**81.1**	**85.7**	56.6	**80.3**	**72.4**	**83.2**	**83.4**	76.3	77.1	42.2	**77.3**	74.8	**82.4**	70.2	**73.7**

Table 1: Comparison with state-of-the-arts on PASCAL VOC 2007 [18] test set.

Method	Dataset	
	VOC2007 test	VOC2012 val
Hierarchical-RL[13]	28.1	36.0
Stefan-RL[19]	26.5	27.0
Caicedo-RL[7]	46.1	51.8
Multitask-RL[16]	48.6	52.1
Multistage-RL[20]	40.7	46.3
Tree-RL[17]	73.1	67.8
ReinforceDet	**73.6**	**68.4**

Table 2: mean Average Precision (AP) on PASCAL VOC 2007 test and 2012 validation set [18]

Method	Number of Region Proposals	mAP
Fast RCNN[21]	$\sim$ 1.5k	70.0
Faster RCNN[2]	$\sim$ 0.3k	73.2
ReinforceDet	$\sim$ 50	**73.6**

Table 3: Region Proposal comparison with current object detection methods on PASCAL VOC 2007 test set [18]

are reported with mAP metric to thoroughly evaluate the effects. We use Keras framework with GTX TITAN X GPU for implementation.

4.2. Comparison with State-of-the-arts

We report the Average Precision (AP) on PASCAL VOC 2007 test set and 2012 val set [18]. The detailed results are shown in Table 1 and Table 2. It is inspiring to observe that the performance of our ReinforceDet obviously outperforms state-of-the-art methods. To be specific, our method outperforms Hierarchical-RL with **45.5** and **32.4** points, Stefan-RL with **47.1** and **41.4** points, Caicedo-RL with **27.5** and **16.6** points, Multitask-RL with **25** and **16.3** points, Multistage-RL with **22.9** and **22.1** points, Tree-RL with **0.5** and **0.6** points on Pascal VOC 2007 testing and 2012 Validation dataset respectively.

Compared with classic region-based object detection methods, the Table 3 clearly demonstrates that our proposed method achieves comparable performance within less resource consumption. Our RL based method is capable of screening out redundant region proposals. However, the region-based methods provide tens of thousands of region proposals without taking account of image context (like object number). In our work, we decouple the two-branch CNN network as two cascaded subnetworks (viewed as IoU-net and bounding box net), where the former is responsible for selecting the optimal one and the latter refine it. Using the paradigm of reinforcement learning and the decoupled mechanism, our proposed method achieves a good trade-off between precision and resource consumption.

Novel Reward	Apprenticeship Learning	Decoupled Two-branch Network	Dataset	
			VOC2007 test	VOC2012 val
			36.1	39.5
✓			40.1	42.7
✓	✓		69.4	63.1
✓		✓	71.1	66.7
✓	✓	✓	**73.7**	**68.4**

Table 4: Ablation studies of each component in ReinforceDet. Results are measured with mAP.

4.3. Ablation Study

The Table 4 shows the effects of each component on our method. It is clearly shown that each proposed technique is beneficial to improve the performance. The reward function and apprenticeship learning optimize the reinforcement learning agent to provide much accurate region proposal set. The two-branch network exploit the powerful feature representation from CNN to enhance the observation (also as state) and regress the output from RL network.

5. CONCLUSION

In this work, we propose a reinforcement learning based object detection method ReinforceDet, which integrates both the advantage of reinforcement learning and CNN to achieve promising performance. In our method, we propose a novel reward function and apprenticeship learning used for region proposal task. In addition, to further refine the region proposals, we present a two-branch CNN network for IoU prediction and bounding box regression. To save the computational cost in inference, we also propose a decoupled pipeline during testing. Extensive experiments demonstrate that our ReinforceDet achieves promising results within low-memory resource consumption.

6. REFERENCES

[1] Zhaowei Cai and Nuno Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[2] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[4] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun, "Megdet: A large mini-batch object detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6181–6189.

[5] Debang Li, Huikai Wu, Junge Zhang, and Kaiqi Huang, "A2-rl: Aesthetics aware reinforcement learning for image cropping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8193–8201.

[6] Xiaojun Chang, Po-Yao Huang, Yi-Dong Shen, Xiaodan Liang, Yi Yang, and Alexander G Hauptmann, "Rcaa: Relational context-aware agents for person search," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 84–100.

[7] Juan C Caicedo and Svetlana Lazebnik, "Active object localization with deep reinforcement learning," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2488–2496.

[8] Pieter Abbeel and Andrew Y Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 1.

[9] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[10] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin, "Reppoints: Point set representation for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9657–9666.

[11] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9627–9636.

[12] Xiaoning Han, Huaping Liu, Fuchun Sun, and Xinyu Zhang, "Active object detection with multistep action prediction using deep q-network," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3723–3731, 2019.

[13] Miriam Bellver, Xavier Giró-i Nieto, Ferran Marqués, and Jordi Torres, "Hierarchical object detection with deep reinforcement learning," *arXiv preprint arXiv:1611.03718*, 2016.

[14] Yang Li, Kun Fu, Hao Sun, and Xian Sun, "An aircraft detection framework based on reinforcement learning and convolutional neural networks in remote sensing images," *Remote Sensing*, vol. 10, no. 2, pp. 243, 2018.

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

[16] Yan Wang, Lei Zhang, Lituan Wang, and Zizhou Wang, "Multitask learning for object localization with deep reinforcement learning," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 4, pp. 573–580, 2018.

[17] Zequn Jie, Xiaodan Liang, Jiashi Feng, Xiaojie Jin, Wen Feng Lu, and Shuicheng Yan, "Tree-structured reinforcement learning for sequential object localization," *arXiv preprint arXiv:1703.02710*, 2017.

[18] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[19] Stefan Mathe, Aleksis Pirinen, and Cristian Sminchisescu, "Reinforcement learning for visual object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2894–2902.

[20] Jonas König, Simon Malberg, Martin Martens, Sebastian Niehaus, Artus Krohn-Grimberghe, and Arunselvan Ramaswamy, "Multi-stage reinforcement learning for object detection," in *Science and Information Conference*. Springer, 2019, pp. 178–191.

[21] Ross Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.