Neurocomputing 443 (2021) 369–379

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# ReinforceNet: A reinforcement learning embedded object detection framework with region selection network

Man Zhou [a,b], Rujing Wang [a], Chengjun Xie [a,*], Liu Liu [a,b], Rui Li [a,b], Fangyuan Wang [a,b], Dengshan Li [a,b]

[a] Institute of Intelligent Machines, Hefei Institute of Physical Science, Chinese Academy of Sciences, Hefei 230031 China
[b] University of Science and Technology of China, Hefei 230026 China

ARTICLE INFO

*Article history:*
Received 3 March 2020
Revised 29 April 2020
Accepted 3 February 2021
Available online 10 March 2021
Communicated by Shijian Lu

*Keywords:*
Reinforcement learning
Convolutional neural network
Region selection network
Object detection

ABSTRACT

In recent years, researchers have explored reinforcement learning based object detection methods. However, existing methods always suffer from barely satisfactory performance. The main reasons are that current reinforcement learning based methods generate a sequence of inaccurate regions without a reasonable reward function, and regard the non-optimal one at the final step as the detection result for lack of an effective region selection and refinement strategy. To tackle the above problems, we propose a novel reinforcement learning based object detection framework, namely ReinforceNet, possessing the capability of the region selection and refinement by integrating reinforcement learning agents' action space with Convolutional Neural Network based feature space. In ReinforceNet, we redevelop a reward function that enables the agent to be trained effectively and provide more accurate region proposals. In order to further optimize them, we design Convolutional Neural Network based region selection network (RS-net) and bounding box refinement network (BBR-net). Particularly, the former consists of two sub-networks: Intersection-of-Union network (IoU-net) and Completeness network (CPL-net) which are employed jointly for selecting the optimal region proposal. The latter aims to refine the selected one as the final result. Extensive experimental results on two standard datasets PASCAL VOC 2007 and VOC 2012 demonstrate that ReinforceNet is capable of improving the region selection and learning better agent action representations for reinforcement learning, resulting in the state-of-the-art performance.


## 1. Introduction

Recent years have witnessed some reinforcement learning (RL) based object detection methods [1,2,5–8,12]. These RL based methods always formulate object detection as Markov decision processes (MDP), where RL agent sequentially selects the actions to adjust the aspect ratios of input images within several steps using action-decision strategy until triggering terminal action. An obvious advantage of above RL methods is that only a few region proposals (usually no more than 10 candidates) are required for object detection, while Convolutional Neural Network (CNN) based approaches [3,4,9–11,13] always demand tens of thousands of pre-computed proposals, which makes them difficult to handle the optimal region selection. However, existing RL based object detection methods always suffer from barely satisfactory performance. The main reasons are that RL agent: (i) directly generates a sequence of inaccurate regions without a reasonable reward function, (ii) regards the non-optimal one at final step as detection result without an effective region selection strategy, and (iii) only adopts action space for bounding box regression in RL process.

In this paper, we propose a reinforcement learning embedded object detection framework with region selection and refinement network, a more accurate model integrating RL agents' action space with CNN-based feature space for object detection, as a response to the aforementioned issues. The whole network consists of three main components: (1) RL optimization: a novel reward function for RL agent optimization, (2) RS-net: a region selection network for searching the optimal region proposal, and (3) BBR-net: a bounding box refinement network for further regression.

(1) **RL optimization:** A reasonable reward function is key for RL optimization. However, in previous works [5,6,8], IoU based reward function focuses only on the positive/negative variation of IoU difference between adjacent regions but neglects change magnitude, which does not make RL agent sensitive to small changes. To handle this problem, we simultaneously consider the IoU and Completeness of the change magnitude between adjacent regions into reward function to effectively

* Corresponding author.
*E-mail address:* cjxie@iim.ac.cn (C. Xie).

https://doi.org/10.1016/j.neucom.2021.02.073
0925-2312/

train RL agent. Particularly, Completeness is a newly-defined evaluation metric which could measure the completeness of target object in image. Generally speaking, it is reasonable to assume that the RL agent can gradually enhance learning capacity with running epoch rising. Therefore, we introduce multiple agents rather than single agent to cover and find the optimal region proposal of the detection results.

(2) **Region selection network (RS-net):** In MDP, RL agent [17] sequentially searches for objects by utilizing both the current observation of region image and historical search paths. When RL agent stops search, the final-step region proposal is treated as the detection result. However, Fig. 1 indicates that most of the final-step region proposals are not optimal from observing results of different RL detection methods. To handle this problem, we design a novel network, namely RS-net, to select the optimal region proposal. RS-net consists of two sub-networks: IoU-net and CPL-net, which are responsible for computing IoU and Completeness values of each region proposal respectively. The predicted IoU and Completeness values are used jointly to assess region proposals and select the optimal one, as shown in Fig. 1-(b) and Fig. 4.

(3) **Bounding box refinement network (BBR-net):** Compared with CNN based methods [9,13], standard RL based object detection methods employ action space instead of feature space for bounding box regression. For example, Bueno et al. [8] employs five pre-defined actions to refine the candidate bounding boxes. Nevertheless, the result of region refinement is deeply limited by the parameters of pre-defined action space since the pre-defined actions cannot cover the target size space. Motivated by bounding box fine-tuning strategy adopted in two-stage object detection architectures [9,13], we design a bounding box refinement network (BBR-net) that integrates both action space and feature space for further regression. Specifically, we recurrently exploit CNN backbone for extracting the local feature maps of the proposals selected from RS-net and simultaneously employs them into RL framework for training BBR-net. This strategy provides a more complementary mechanism to harness the inaccurate object location problems.

The major contributions of this paper are as follows:

(1) A novel reinforcement learning based object detection network, namely ReinforceNet, is proposed. ReinforceNet possesses the capability of the region selection and refinement by integrating RL agents' action space with CNN-based feature space.

(2) We redevelop an IoU and Completeness jointly guided reward function, which makes RL agent sensitive to small change magnitudes between adjacent regions. Besides, we replace the single agent with multiple agents, which enrich the expressiveness of our object detection framework.

(3) Extensive experiments on PASCAL VOC 2007 and VOC 2012 object detection benchmarks demonstrate the superior performance of our ReinforceNet compared to state-of-the-art methods.

## 2. Related work

**CNN based Detector** The leading approaches in object detection are currently CNN-based deep detectors, which can be summarized as two-stage [9,23,24] and single-stage detectors [10,11]. For

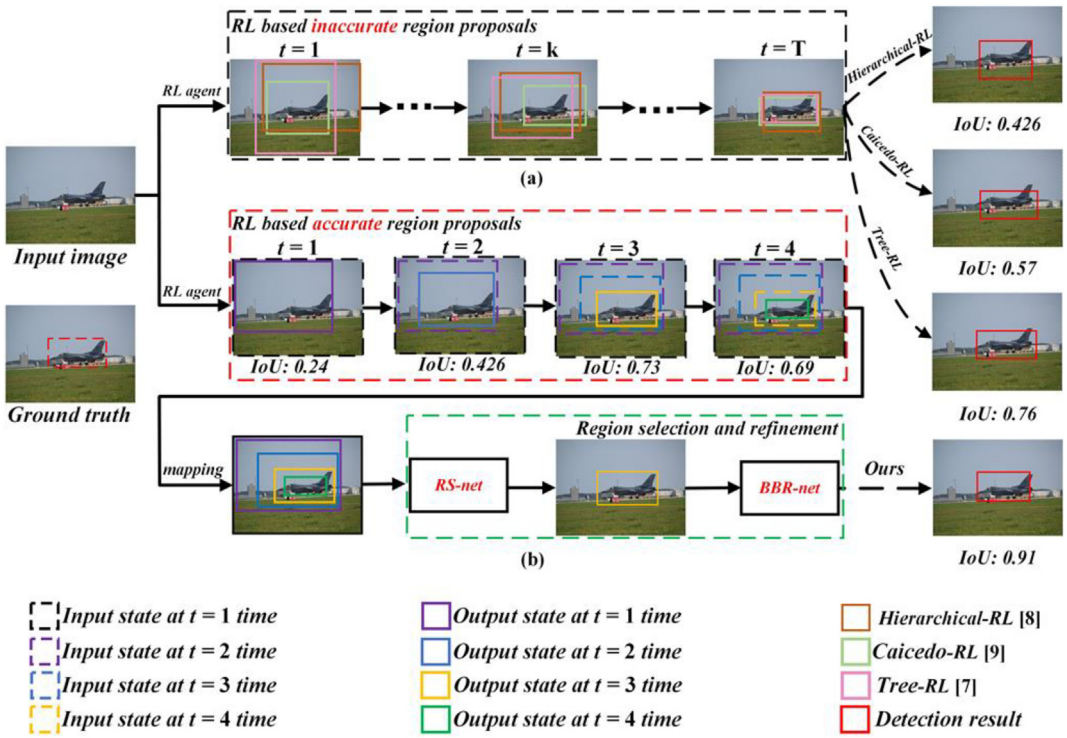



**Fig. 1.** Comparing our work against other RL architectures. Fig. 1-(a) shows quantitative aeroplane detection results with IoU values from different methods: Hierarchical-RL [8], Caicedo-RL [9] and Tree-RL [7], respectively. These RL methods generate a sequence of inaccurate regions and regard the non-optimal one at the final step as the detection result. Fig. 1-(b) illustrates the detection process of our ReinforceNet. Firstly, the well-trained RL agent sequentially takes appropriate actions to transform the input image and provide the accurate region proposals. At time t = 1, 2, 3 and 4, it transforms the previous window (input image, dashed-line violet, blue and yellow boxes) to a new state (violet, blue, yellow and green boxes) respectively. Then, the designed RS-net is responsible for effectively selecting the optimal region (yellow box) from all the region proposals. Finally, the proposed BBR-net further refines the optimal one as the detection result. It can be clearly seen that our ReinforceNet outperforms the state-of-the-art methods.

two-stage detector, the pioneer work R-CNN is reported in [13] by combining external region proposal module and a region-wise classifier to formulate object detection. Although this method appears promising robustness for object detection, its extensive computational cost to obtain region proposals by Selective Search algorithm [14] ensures that the detector is not applicable in reality. To relieve redundant computation of convolutions, the developed SSP-Net [25] and Fast R-CNN [26] extract feature of region proposals from shared full-image feature maps generated by CNN backbone. As Fast R-CNN has enhanced the bottleneck of redundant computation, the detector is still not trained end-to-end. Later, to handle this problem, Faster RCNN [9] introduces a fully convolutional network, Region Proposal Network (RPN), which exploits attention mechanism to tell Fast RCNN where to look. In addition, RPN generates high-quality sparse RoIs and further improves the performance. We note that there is a second kind of detectors, e.g. YOLO, RetinaNet and SSD [10,11,23]. Unlike the first two-stage detector, the second is not based on region proposals. While their performances in runtime have long trailed behind those of two-stage detectors, the two-stage detector can be adapted to a wide range of requirements on accuracy.

**Reinforcement Learning based Detector** It is worth noting that RL based object detection methods are more related to our ReinforceNet. Caicedo et al. [6] design an active object detection model, which applies Deep Q-learning Network [15] to learn action-decision policy to search target until triggering terminal action, and obtains comparative results with RCNN. Then Bueno et al. [8] 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. [12,18] use restricted Edge Boxes [19] to get more appropriate high-quality candidate boxes through the prior knowledge, which achieves high accuracy and recall. Furthermore, aforementioned methods adopt IoR (inhibition-of-return) mechanism to attend multiple object detection problem. In addition, Ba et al. [22] introduce a deep recurrent attention model (RAM) to recognize multiple objects, which trained with RL. Then, the developed clued RAM [25] proposes to add clue or constraint to guide RL agent quickly search the zooms including object, which speeds up detection and achieves better performance. Recently, Chen et al. [21] exploits LSTM network to sequentially capture contextual dependencies for locating attentional regions related to different semantic objects, which obtains off-the-shell results. Recently, there have witnessed several reinforcement learning based works in efficient object detection [32,33,37], weakly supervised object detection [34] and object segmentation [35,36]. [32] proposed low and high spatial resolution dual-road reward to efficiently train RL agent, which adaptively selects the spatial resolution of each image. Compared with previous work that the detector is applied to each part of large image, it greatly increases run-time efficiency. Instead of adopting hand-crafted frame sampling strategies, Wu et al. [33] effectively employed multi-agent to reformulate frame sampling as multiple parallel Markov decision processes. The work in [34] focused on weakly supervised object detection by exploiting image-level annotation to generate pseudo target region, which was further processed as initial location for further detection. [35] firstly introduced reinforcement learning for video object segmentation and adopted RL agent to provide object and context box, which then fed into followed fully convolution network for segmentation. [36] proposed a multi-agent object segmentation method by combining user feedback and gamification strategy to obtain promising performance. [37] used YOLO-V3 to fast provide region proposals and then integrated multi-agent system with Independent Q-Learners (IQL) for object tracker.

## 3. Material and methods

### 3.1. ReinforceNet model

In this section, we will present our ReinforceNet, a novel reinforcement learning embedded object detection framework. In detail, the complete technical pipeline is depicted in Fig. 2 and composed of three main parts: (i) multiple RL agents for jointly generating more accurate region proposals, (ii) RS-net for selecting the optimal region proposal, and (iii) BBR-net for refining the optimal one as the detection result. All the above parts will be illustrated in the following sub-sections.

#### 3.1.1. RL agent

In ReinforceNet, RL agents are responsible for generating region proposals as shown in Fig. 2. Specifically, our RL agents rely on

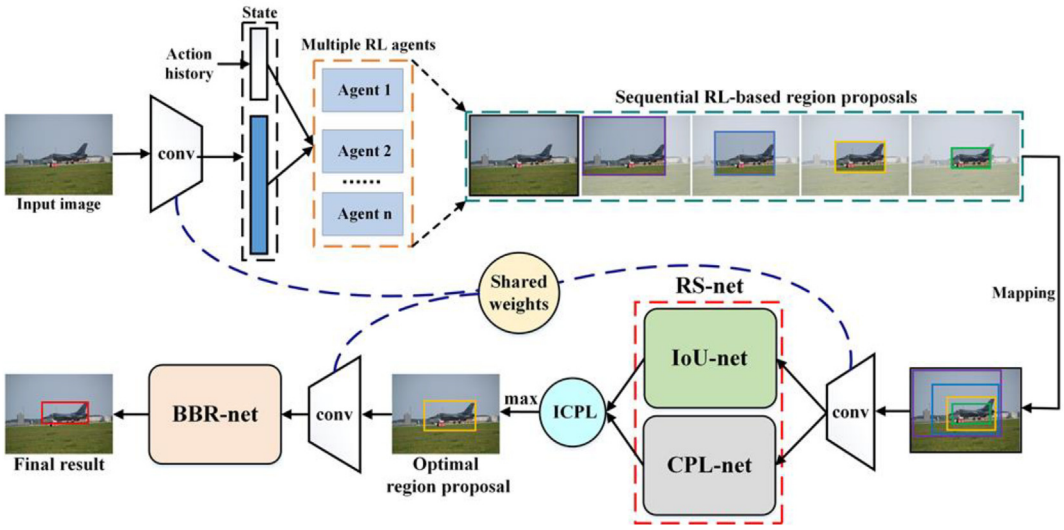



**Fig. 2.** Technical pipeline of our proposed ReinforceNet. Firstly, input images are fed into CNN backbone to obtain feature maps. Secondly, RL agents execute a sequence of actions to locate the object by observing feature maps and simultaneously output a set of region proposals. Subsequently, RS-net including IoU-net and CPL-net is exploited to select the optimal region proposal. Finally, BBR-net is responsible for further refining the optimal one as the result.

pre-trained Vgg16 model [21,24] as backbone architecture, as this kind of model enjoys powerful feature extraction advantage. Given an input image, it can directly provide feature maps linked to our RL agents to generate region proposals at hand. In our work, our multiple RL agents are DQN framework, whose input (namely state in MDP) contains two parts: feature vector of current region image and taken action history vector. Furthermore, the output of DQN is the expected estimation of accumulated reward value of each action for given state. In the test phase, the well-trained RL agent uses greedy policy to select and execute the action with maximal reward and generates region proposals, as shown in Fig. 3.

Specifically, while testing an image, the whole image is fed into vgg16 backbone to obtain feature vector at first. Then feature vector concatenates history action vector as state, where history action vector represents taken action sequence. Next, a well-trained RL agent observes state to samples action space by action-decision policy and then implements actions to transform current image as corresponding aspect ratio sub-region. Finally, obtained sub-region continue to perform above process until terminal action is triggered. All above sub-regions are collected together as region proposals.

### 3.1.2. RS-net

The aim of RS-net is to select the optimal region from the region proposals by two sub-networks: IoU-net and CPL-net (as illustrated in Fig. 4). The two sub-networks jointly model a discriminative region selection and are to be aligned at the corresponding region proposals within the image. For instance, Fig. 4 shows the detailed optimal region selection process of RS-net for airplane image. Given the candidate region proposals provided by RL agents, the IoU-net and CPL-net give out IoU and Completeness values for each region respectively, which are drawn as the green and blue line. Then the red line is obtained by computing ICPL in Eq. (4). The time step region corresponding to the peak of red line is optimal one.

**IoU-net:** In Figs. 2 and 4, RL agents generate various size of region proposals for each image. To match input of IoU-net, each region proposal is firstly resized as 224 × 224 resolutions regardless of its aspect ratio, and then it is fed into Vgg16 backbone to obtain the corresponding feature maps. The feature maps are carried into 4-layer fully connected layers for the IoU prediction.

As for IoU-net training, we collect training samples by employing well-trained RL agents to transform iteratively the initial bounding boxes (which are the whole image) on PASCAL VOC 2007 and 2012 dataset. Meanwhile we introduce ground truth to calculate label for each region proposal, which is IoU in Eq. (1).

Assume that number of collected samples is $N$, the IoU between each region proposal $b_i$, $i \in \{1, \cdots, N\}$ and its ground truth bounding box $g_i$, is defined as:

$$IoU_i = (b_i \cap g_i)/(b_i \cup g_i) \tag{1}$$

where $b_i \cap g_i$ and $b_i \cup g_i$ is intersection and union of $b$ and $g$ at mathematic, respectively. To optimize IoU-net, we adopt smooth-L1 loss:

$$loss_{IoU-net} = -\frac{1}{n}\sum_{i=1}^{n} smooth - L1(IoU(b_i \cap g_i)) \tag{2}$$

where $n$ is the number of batch size. It is not hard to find out that the IoU values of region proposals are unevenly distributed as shown in Fig. 5-(a), thus resulting in a terrible predictor. To solve this problem, we design the following training process: (1) Firstly, we remove from this candidate region set the bounding boxes having an IoU less than 0.3 with ground truth. (2) Secondly, we rank the screened candidate region set by IoU and then spilt them into 7 zooms. Next, we uniformly sample a batch of region proposals from each zoom to train IoU-net. (3) Labels transformation: finally, we transform IoU labels with 2 × IoU-1 operation so as to maintain sample labels for obeying standard Gauss distribution with mean 0. This data generation process empirically brings better performance and robustness to the IoU-net.

**CPL-net:** The second sub-network to compute Completeness jointly considers the change magnitude of adjacent regions in an optimization problem.

In Fig. 5-(b), experiments show that Completeness of each region proposal gradually decreases in chronological order and region proposal set has an uneven distribution, and most of Completeness values of regions cluster around 0.7. For training CPL-net, we manually generate samples by well-trained RL agents and then assign label (which is Completeness in Eq. (3)) to each sample.

Assume that number of collected samples is M, the completeness between each region proposal $b_j$, $j \in \{1, \cdots, N\}$ and its ground truth bounding box $g_j$, is defined as:

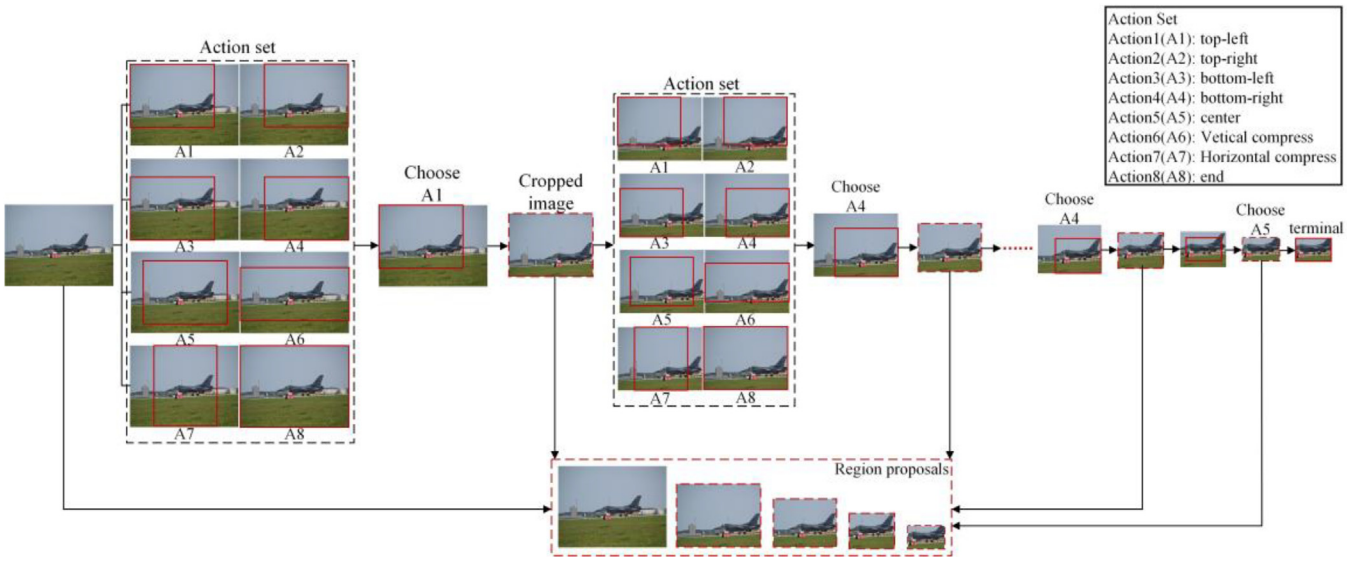



**Fig. 3.** Illustration of the sequential decision-making based region generation process of single reinforcement learning agent.

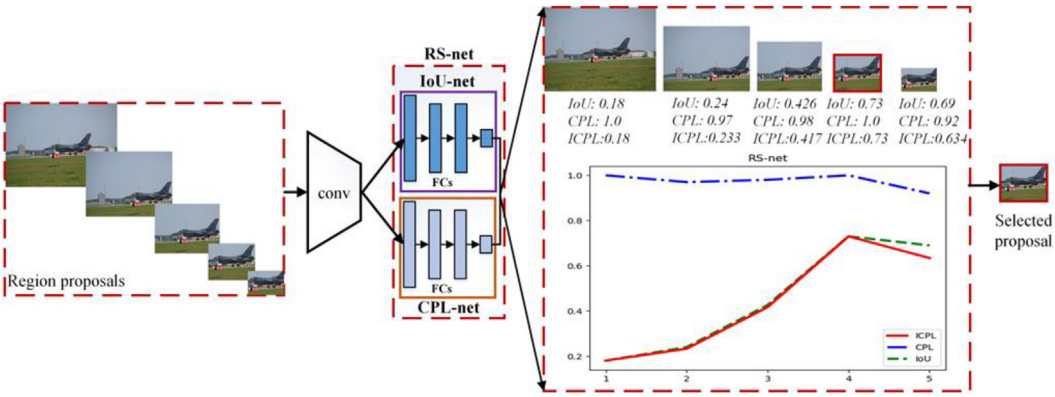



**Fig. 4.** Complementary roles of the IoU-net and CPL-net for searching optimal region proposal.

$$CPL_j = (b_j \cap g_j)/g_j \tag{3}$$

where $b_j \cap g_j$ s intersection of $b$ and $g$ at mathematic. To optimize CPL-net, we also adopt smooth-L1 loss. Specifically, as for CPL-net training, we manually generate training samples by this way: Firstly, we execute the similar operation as IoU-net by using RL agents to generate sequential candidate boxes as part of training set. To handle sample imbalance, we manually construct some regions by random cropping the initial image and then select regions with Completeness less than 0.6 with ground truth. To better train CPL-net, we use the similar sampling policy as IoU-net.

In order to select optimal region, we fuse the output of IoU-net and CPL-net by the following operation:

$$RS-net = max_{\substack{k=1,2,\cdots,C \\ h=1,2,\cdots,D}} \{ICPL|ICPL = IoU^{(k)}(h) \times CPL^{(k)}(h)\} \tag{4}$$

where $C$ is the number of agents and set to 3 in our work, $D$ is the number of region proposals and less than 10, which depends on the number of candidate regions of each image.

### 3.1.3. BBR-net

As clearly declared above, the RS-net, including IoU-net and CPL-net, is designed to select the most appropriate region proposal. Though the RS-net relieves the issue that current RL-based methods regard the non-optimal one at the final step as the detection result for lack of an effective region selection strategy, the output from RS-net is not satisfactory. Consequently, followed by RS-net, we propose an additional BBR-net to further improve the performance. So the agent cannot terminate its sequential decision-making process at the proposal selected by the IoU-net and CPL-net.

BBR-net is proposed to refine the optimal region selected by RS-net. Specifically, BBR-net consists of five fully connected layers [20], where the first four and final layers are used for feature reduction and bounding box regression, respectively. However, there is a conflict between the arbitrary size of region proposals and fixed-dimensional input layer of BBR-net. Different from Faster RCNN that exploits RoI pooling layer to resolve it, our proposed ReinforceNet treats arbitrary size of region proposals by resizing them as common $224 \times 224$ resolutions, which possesses an obvious advantage for cropped small object region to obtain informative representation. As shown in Fig. 2, the optimal yellow box is firstly cropped from the whole image and then resized as $224 \times 224$ resolutions. Next, we feed the resized yellow box into Vgg16 backbone, which pre-trained on ImageNet dataset to extract feature maps. At the last, the more precise red box is predicted by BBR-net from observing the obtained feature maps.

In addition, to relieve the distribution difference between training and testing dataset, as for BBR-net training, we elaborately construct training samples in two ways: firstly, we forward the

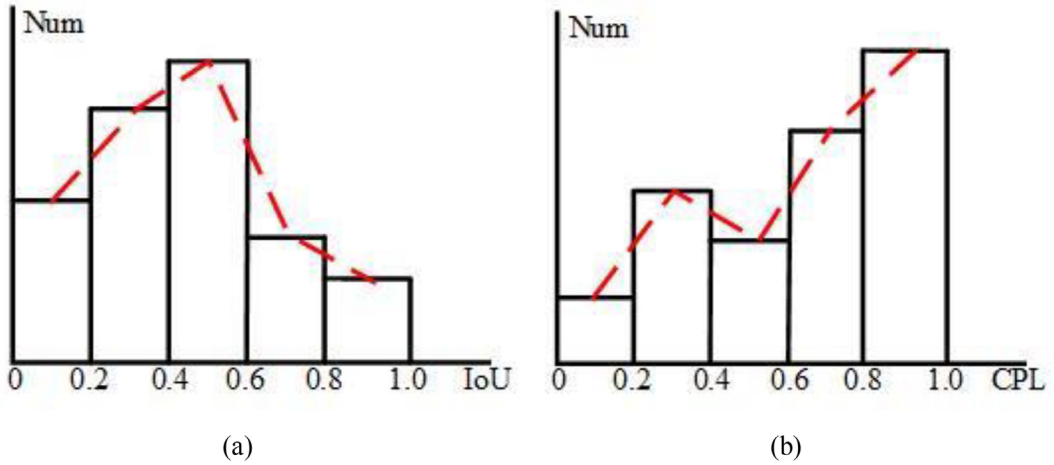



**Fig. 5.** Illustration of IoU and Completeness distribution of region proposals.

PASCAL VOC 2007 and 2012 training set into RL agent to generate region proposals. Secondly, we introduce ground truth to calculate IoU of each region in each image. Finally, we select the region (1) with maximal IoU value and (2) having IoU value more than 0.4 to form training set. To maintain the translation and rotation invariance, we transform the supervised coordinates corresponding to each region in training set as similar to [13]. Moreover, we introduce smooth-L1 loss [9] to optimize BBR-net to obtain better results.

### 3.2. ReinforceNet optimization

Essentially, region proposals have been thought of as a key factor for object detection. To this end, we redevelop a reward function to optimize RL agent for generating more accurate region proposals, which deeply prompts follow-up selection and regression task. Moreover, since using a single agent to provide region proposals may obtain satisfactory results, it only achieves a suboptimal solution. Thus, we integrate multiple RL agents jointly to enrich expressiveness of object detection.

#### 3.2.1. Reward function

Formally, MDP is made up of three key components: pre-defined action set, fixed-dimensional state and reward function. In this work, state space follows similar definition as [8] and action space is expanded with two novel actions of horizontal and vertical suppression by 0.7 times. The two increased actions may relieve the incompleteness problem of detection results. All the actions are descripted in Fig. 3, which are divided into two classes, i.e., transformation action to move image region and terminal action to stop search process.

As for transformation action, this kind of reward function should reflect the performance change of state transition. The reward function is shown as bellow:

$$R_a(s_t, s_{t+1}) = \begin{cases} r + \beta IoU(s_t, s_{t+1}) CPL(s_t, s_{t+1}) - \lambda tIoU(s_t, s_{t+1}) > 0 \\ -r - \beta IoU(s_t, s_{t+1}) CPL(s_t, s_{t+1}) - \lambda tIoU(s_t, s_{t+1}) < 0 \end{cases} \tag{5}$$

Where:

$$IoU(s_t, s_{t+1}) = IoU(s_{t+1}) - IoU(s_t) \tag{6}$$

$$CPL(s_t, s_{t+1}) = CPL(s_{t+1}) - CPL(s_t) \tag{7}$$

and we clearly find out $CPL(s_t, s_{t+1}) \leq 0$ by considering Eq. (3) and pre-defined action space in our work. Where the first term is relevant to IoU and we set r as 1. When the transformation is good (that means the IoU increasing), the obtained value is positive 1, and in contrast negative $-1$. The second term shows the change magnitude of IoU and Completeness. When the change is sharp largely, the term value is bigger. The third term is a punish factor to speed the running process of RL agent. As for the terminal action, the corresponding reward function follows similar setting as [6,7], except that the reward value is set to 5. Once this action is taken, the agent would stop search process.

#### 3.2.2. Multiple agents

It is reasonable that the RL agent can gradually enhance learning capacity with the training epochs rising. In previous works [1,2,5–7], researchers used to fix a single epoch agent for testing. While the previous methods obtain promising performance, the fixed single agent methods to select region proposals may only achieve a suboptimal solution.

As seen in [8], the precision of the 50th epoch is obviously lower than the 45th epoch at recall more than 0.3. To solve the issue, we introduce multiple agents to search optimal region in more extensive space. In detail, we employ multiple agents jointly to generate region proposals and then use RS-net to select the optimal one over all the region proposals.

By conducting numerous experiments, we find out a phenomenon that the capacity of agent is not necessarily increased with the training process proceeding and each epoch agent expresses different sensitivity for different scale object. To this end, we integrate multi-epoch agents to search more appropriate region proposal for different scale of object. Because the multiple agents come from different epochs in the same training process, the agents commonly make different decisions.

In addition, to balance the computation cost and the performance improvement leaded by multiple agents, we implement model-pruning strategy [27] to decide which epoch agents are selected by ranking the accuracy of the last 15 epochs on PASCAL VOC 2007 validation set. Ultimately, from experiments, we find out that the agents of 39th, 45th, 50th epoch are appropriate for jointly generating region proposals in a compromise way. The region generation process of multiple RL agents is described in Eq. (4).

## 4. Results and discussion

### 4.1. Experiment setting

All the experiments in this section are conducted on two widely-used object detection benchmark datasets, i.e., PASCAL VOC 2007 [29] and 2012 [30], which both consist of 20 categories. Because the ground truth annotations of VOC 2012 testing set have been not released publicly, we elaborately design two sets of experiments. 1) ReinforceNet is trained on the union of the 2007 and 2012 training-validation set, and tested on VOC 2007 testing set; 2) ReinforceNet is trained with VOC 2012 training set and tested on VOC 2012 validation set. We extract features to represent all region proposals with the fully convolutional backbone architecture Vgg16 whose model is pre-trained on ImageNet [16,28]. Specifically, we use 50 epochs, as advised by the authors of prior works [8], to train RL agent with $\epsilon$-greedy policy. We also set $\beta = 0.01$ and $\lambda = 0.001$ in Eq. (5) to stabilize and speed up training process. In addition to mAP metric, results are reported with mean IoU metric [31], to thoroughly evaluate the effects of proposed improvements. We use Keras framework with GTX TITAN X GPU for implementation.

### 4.2. Performance comparison

In this section, we present comprehensive experiments to demonstrate the effectiveness of our proposed ReinforceNet by qualitative and quantitative analysis. We test the prototype reinforcement learning based Hierarchical-RL detector [8], Caicedo-RL detector [6], and Tree-RL detector [7], Multitask-RL detector [10], Multistage-RL detector [38], Parameterized-RL detector [39], Stefan-RL detector [40] for comparison in Section 4.2.1. Additionally, we also choose the typical VGG-based Faster R-CNN detector [9], YOLO detector [10] and SSD detector [38] to compare with our model in Section 4.2.2. All the detectors use VGG16 as CNN backbone for fair comparisons. Furthermore, we adopt the best results reported in the original literature for comparison.

#### 4.2.1. Performance comparison with RL-based methods

**Average Precision performance:** Tables 1 and 2 clearly illustrate Average Precision (AP) per category of following methods including our proposed ReinforceNet, Tree-RL, Hierarchical-RL, Caicedo-RL, Multitask-RL, Multistep-RL, Parameterized-RL, Stefan-RL on PASCAL VOC 2007 testing and 2012 validation set. It is inspir-

**Table 1**
Average Precision (AP) per category on the PASCAL VOC 2007 testing set. "M", "C" and "B" represents the component of multiple agents, CPL-net and BBR-net respectively. "ReinforceNet minus X"($X \in \{M, C, B\}$) indicates that the corresponding components are removed from ReinforceNet.

| Method | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Caicedo-RL | 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 | 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 | 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 |
| Multistage-RL | – | –– | –– | –– | –– | –– | –– | –– | –– | –– | –– | –– | –– | –– | –– | –– | –– | –– | –– | –– | 40.7 |
| Tree-RL | 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 |
| Parameterized-RL | 76.7 | 82.1 | 74.3 | 65.1 | 52.4 | 80.4 | 80.9 | 85.3 | 56.5 | 80.2 | 72.5 | 83.1 | 83.5 | 76.1 | 77.4 | 42.6 | 77.4 | 75.0 | 81.8 | 70.1 | 73.6 |
| ReinforceNet minus M/C/B | 51.7 | 45.2 | 57.4 | 52.9 | 40.5 | 53.2 | 63.5 | 55.4 | 48.3 | 55.9 | 61.0 | 62.4 | 66.2 | 57.4 | 55.8 | 34.3 | 50.1 | 45.1 | 57.1 | 44.7 | 53.4 |
| ReinforceNet minus M/C | 73.2 | 78.8 | 71.2 | 62.1 | 50.1 | 77.7 | 79.1 | 83.5 | 54.7 | 77.8 | 70.6 | 78.6 | 81.3 | 71.8 | 73.1 | 39.8 | 68.0 | 67.4 | 79.7 | 67.2 | 70.3 |
| ReinforceNet minus M | 74.9 | 80.3 | 73.0 | 63.7 | 51.8 | 79.5 | 80.3 | 84.2 | 55.3 | 79.5 | 71.2 | 81.1 | 82.0 | 74.7 | 75.5 | 41.1 | 72.1 | 70.8 | 80.6 | 69.4 | 72.0 |
| ReinforceNet | 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 2**
Average Precision (AP) per category on the PASCAL VOC 2012 validation set.

| Method | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Hierarchical–RL | 48.5 | 39.8 | 24.1 | 35.1 | 19.8 | 50.1 | 26.6 | 63.0 | 20.5 | 31.0 | 30.0 | 49.1 | 42.4 | 51.3 | 50.4 | 13.6 | 22.1 | 33.2 | 44.6 | 12.3 | 36.0 |
| Stefan-RL | 47.4 | 31.4 | 21.0 | 9.5 | 2.5 | 44.7 | 19.4 | 50.3 | 6.1 | 18.1 | 21.1 | 46.8 | 35.8 | 40.4 | 18.7 | 8.5 | 17.8 | 18.6 | 41.5 | 38.8 | 27.0 |
| ReinforceNetminus M/C/B | 72.4 | 67.3 | 56.5 | 35.4 | 32.1 | 55.2 | 53.6 | 80.7 | 31.6 | 56.8 | 61.0 | 71.9 | 71.8 | 52.1 | 53.4 | 38.8 | 59.4 | 50.8 | 72.5 | 51.3 | 55.1 |
| ReinforceNet minus M/C | 80.6 | 75.7 | 67.0 | 46.4 | 46.2 | 73.5 | 69.5 | 84.1 | 41.4 | 60.1 | 70.6 | 80.2 | 73.6 | 77.8 | 78.4 | 39.8 | 67.6 | 56.4 | 75.9 | 58.4 | 65.3 |
| ReinforceNetminus M | 81.7 | 78.8 | 70.9 | 47.5 | 48.3 | 76.1 | 68.6 | 84.7 | 43.2 | 60.9 | 71.2 | 83.7 | 75.4 | 80.3 | 78.9 | 41.2 | 68.1 | 60.3 | 76.2 | 65.7 | 67.3 |
| ReinforceNet | 82.1 | 79.7 | 71.2 | 49.2 | 50.7 | 78.7 | 70.4 | 85.0 | 44.3 | 61.5 | 72.4 | 84.4 | 76.1 | 81.2 | 80.1 | 41.4 | 69.2 | 63.2 | 77.4 | 67.2 | 68.4 |

ing to observe that the performance of our ReinforceNet obviously outperforms state-of-the-art methods. Compared with existing RL-based detectors on VOC 2007, our method improves mAP by 45.6 points than Hierarchical-RL, 27.6 points than Caicedo-RL, 0.6 points than Tree-RL, 25.1 points than Multitask-RL, 33 points than Multistep-RL and 0.1 points than Parameterized-RL respectively. Besides, our method outperforms Stefan-RL by 41.4 points on VOC 2012 and Hierarchical–RL by 32.4 points on VOC 2012. Specifically, the large improvement of our method is attributed to the developed reward function for optimizing RL agent better to select more appropriate actions and the additional BBR-net for further regression. In addition, we expand the action space of Hierarchical-RL, thus fitting the aspect ratios of targets better.

**Mean IoU performance:** Tables 3 and 4 shows mean IoU per category of Hierarchical-RL, Faster R-CNN and our proposed ReinforceNet on PASCAL VOC 2007 testing and 2012 validation set, respectively. The mean IoU measures the overall degree of closeness between the ground truth and detection results. It can be seen that ReinforceNet outperforms Hierarchical-RL with a lager margin by 36.5 points and 29.6 points on PASCAL VOC 2007 testing and 2012 validation set, respectively. That is because the developed reward function makes RL agent sensitive to small change and the well-trained RL agent generates more optimal region proposals. Furthermore, BBR-net takes advantage of both action space and feature space for further refining the region proposals, which obtains higher mean IoU. Compared with Faster RCNN, our ReinforceNet obtains 1.6 and 0.2 points improvement on PASCAL VOC 2007 testing and 2012 validation set, respectively.

#### 4.2.2. Performance comparison with CNN-based methods

RL-based object detection methods are different from CNN-based methods. The RL-based methods formulate object detection as Markov decision process and might search the detected objects in several steps, while CNN-based methods often demand to analyzing tens of thousands of pre-computed proposals. In the field of non-dense object detection, the RL-based methods are more effective than CNN-based methods. As for our ReinforceNet, the main contribution is the improved pipeline of Markov process. The thought of selecting optimal region proposal might motivate other RL-based applications, like RL-based image cropping [1], image denoising [41], image Restoration [42]. The comparison performance with CNN based methods are illustrated in Table 5. As can be seen clearly, our ReinforceNet is superior to promising Faster R-CNN based on shallow Vgg16 backbone, which turns out reinforcement learning agent is powerful to generate high-quality object proposals with shallow feature extraction. And our method outperforms YOLO by 10.3 points and obtains comparable performance with SSD. Our RL based method has a good trade-off in detection accuracy (AP) and computational cost (number of region proposals required). Therefore, we believe that our method owns its high research value and advantages to SOTA object detection approaches.

#### 4.2.3. Visualizations comparison

As shown in Figs. 6 and 7, we present some detection results comparison between our ReinforceNet and reinforcement learning based state-of-the-art methods as well as promising Faster RCNN

**Table 3**
Mean IoU per category on the Pascal VOC 2007 testing set.

| Method | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv | mIoU |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Hierarchical–RL | 38.1 | 36.1 | 30.5 | 29.4 | 21.6 | 38.1 | 30.0 | 45.1 | 22.6 | 23.7 | 37.6 | 38.5 | 41.9 | 39.5 | 36.6 | 26.3 | 21.3 | 42.8 | 42.0 | 24.7 | 33.3 |
| Faster RCNN | 65.5 | 68.7 | 59.6 | 55.9 | 50.9 | 71.8 | 68.7 | 74.0 | 51.8 | 66.7 | 62.9 | 71.5 | 72.4 | 68.7 | 67.0 | 54.4 | 63.0 | 69.8 | 71.7 | 68.1 | 65.2 |
| ReinforceNetminus M/C/B | 49.2 | 40.5 | 48.2 | 48.6 | 43.1 | 49.2 | 40.7 | 46.4 | 47.7 | 47.8 | 55.7 | 50.3 | 55.1 | 51.3 | 54.3 | 41.0 | 45.1 | 54.8 | 54.0 | 37.2 | 48.0 |
| ReinforceNet minus M/C | 67.4 | 65.4 | 61.0 | 54.1 | 49.7 | 66.7 | 66.3 | 72.4 | 52.1 | 64.8 | 58.1 | 68.1 | 70.2 | 63.2 | 58.5 | 56.4 | 59.2 | 65.0 | 69.8 | 59.4 | 62.4 |
| ReinforceNetminus M | 70.3 | 67.7 | 62.6 | 54.2 | 50.5 | 71.7 | 66.2 | 74.2 | 53.8 | 65.2 | 62.8 | 68.3 | 71.1 | 65.2 | 62.1 | 61.5 | 62.0 | 69.1 | 70.3 | 62.9 | 64.6 |
| ReinforceNet | 72.1 | 70.1 | 63.3 | 55.3 | 51.3 | 72.8 | 70.2 | 74.8 | 54.5 | 69.8 | 65.9 | 69.4 | 76.8 | 67.4 | 65.3 | 65.6 | 65.1 | 69.8 | 72.0 | 65.1 | 66.8 |

**Table 4**
Mean IoU per category on the PASCAL VOC 2012 validation set.

| Method | aero | bike | bird | boat | bottle | bus | car | cat | chair | Cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv | mIoU |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Hierarchical–RL | 45.3 | 41.1 | 29.1 | 36.3 | 18.7 | 46.2 | 27.2 | 56.2 | 25.4 | 33.7 | 36.5 | 45.7 | 43.6 | 47.1 | 46.8 | 13.3 | 25.3 | 37.1 | 45.6 | 13.2 | 35.7 |
| Faster RCNN | 67.3 | 70.7 | 60.0 | 56.0 | 52.7 | 73.0 | 63.9 | 74.1 | 56.7 | 66.6 | 64.6 | 74.7 | 73.2 | 71.6 | 65.7 | 53.7 | 63.3 | 58.6 | 70.4 | 65.8 | 65.1 |
| ReinforceNetminus M/C/B | 57.7 | 61.5 | 56.3 | 40.3 | 27.4 | 47.2 | 43.3 | 67.1 | 36.7 | 52.1 | 44.2 | 63.4 | 64.7 | 55.2 | 64.1 | 47.9 | 49.7 | 50.1 | 62.8 | 56.7 | 52.4 |
| ReinforceNet minus M/C | 65.1 | 69.8 | 62.3 | 51.7 | 45.6 | 68.8 | 56.2 | 68.9 | 47.6 | 59.3 | 62.0 | 69.7 | 68.5 | 66.6 | 65.5 | 51.2 | 59.6 | 57.3 | 66.7 | 58.6 | 61.0 |
| ReinforceNetminus M | 67.0 | 71.1 | 65.6 | 53.8 | 48.5 | 70.1 | 58.0 | 71.7 | 53.1 | 65.8 | 62.3 | 71.3 | 70.8 | 69.3 | 67.3 | 53.0 | 61.4 | 55.8 | 68.4 | 62.4 | 63.3 |
| ReinforceNet | 67.6 | 73.0 | 67.2 | 54.6 | 49.6 | 72.3 | 63.6 | 76.0 | 54.8 | 66.9 | 63.1 | 72.8 | 72.4 | 72.7 | 71.0 | 53.8 | 62.5 | 57.6 | 69.1 | 65.1 | 65.3 |

**Table 5**
Average Precision (AP) per category on the PASCAL VOC 2007 testing set.

| Method | VOC 2007 | Region proposal number |
|---|---|---|
| Faster RCNN | 73.1 | ~2K |
| YOLO | 63.4 | – |
| SSD | 74.3 | – |
| ReinforceNet | 73.7 | ~50 |

on Pascal VOC 2007 and VOC 2012 respectively. The red boxes represent the detection results of our ReinforceNet. The green boxes indicate the optimal region selected by our RS-net. The yellow and blue box show the detection results of Hierarchical–RL and Faster R-CNN respectively. It can be clearly seen that our proposed ReinforceNet outperforms state-of-the-arts.

### 4.3. Ablation studies

In this section, we conduct a series of ablation studies on the PASCAL VOC 2007 testing and 2012 validation set to analyze the importance of each component.

**Ablation Studies on RS-net:** Ablation studies of RS-net are shown in Table 6 in terms of mAP. RS-net consists of two sub-networks, i.e., IoU-net and CPL-net. The two sub-networks are jointly utilized to select optimal regions, where IoU-net is dominant and CPL-net is auxiliary.

**IoU-net:** In the Table 6, it is clearly observed from the first two rows that IoU-net brings 12.0 and 6.7 points higher AP than the baseline on VOC 2007 and 2012 respectively, thus implying the effectiveness of IoU-net. This is in line with conclusion that the candidate regions in the final step are not necessarily optimal and IoU-net is key for selecting optimal region for object detection.

**CPL-net:** Considering the fourth and fifth rows, CPL-net improves AP from 70.3 to 72.0 and 67.4 to 71.3, which validate the effectiveness of CPL-net. Specifically, the reason is that the more complete region proposals are easily refined to ground truth, as descripted in Fig. 8.

**Ablation Studies on BBR-net:** Considering the effect of BBR-net, we can focus on the third and fourth rows of Table 6. Along with BBR-net, the results are further improved from 53.4 to 70.3 and 55.8 to 67.4 on VOC2007 and 2012 dataset, respectively, which indicates the BBR-net is key for increasing localization performance.

**Ablation Studies on multiple agents:** Ablation Studies on multiple agents are shown in the last row of all the tables. Specifically, in the Table 5, multiple agents bring 1.7 and 1.6 points higher boxes mAP than the single agent baseline on VOC 2007 and 2012 respectively.

## 5. Conclusion

In this work, we have presented a general RL based object detection framework with RS-net and BBR-net. The RS-net allows us to select the optimal region of the target with combination of IoU-net and CPL-net. We state that this framework is applicable for finding more appropriate region proposals since it can compute IoU and Completeness values of each region proposal effectively in practice. For further refining the optimal region from RS-net output, we introduce the BBR-net to converge the optimal region to ground truth. This leads to a significant improvement of the performance for object detection. In addition, an IoU and Completeness integrated reward function is proposed to improve the efficiency

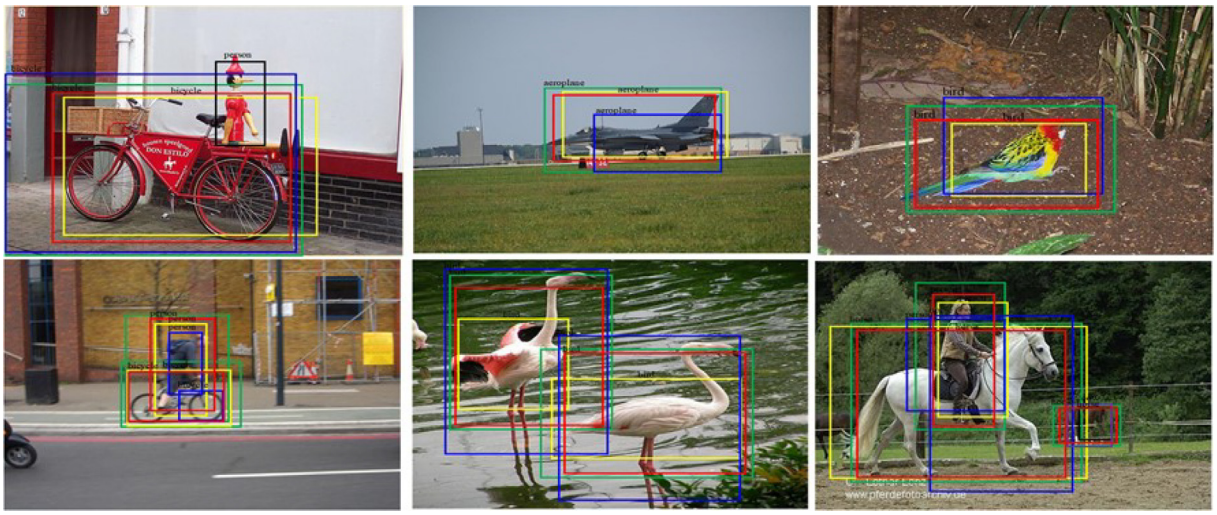


**Fig. 6.** Comparison of detection results between our ReinforceNet and other methods on VOC 2007.

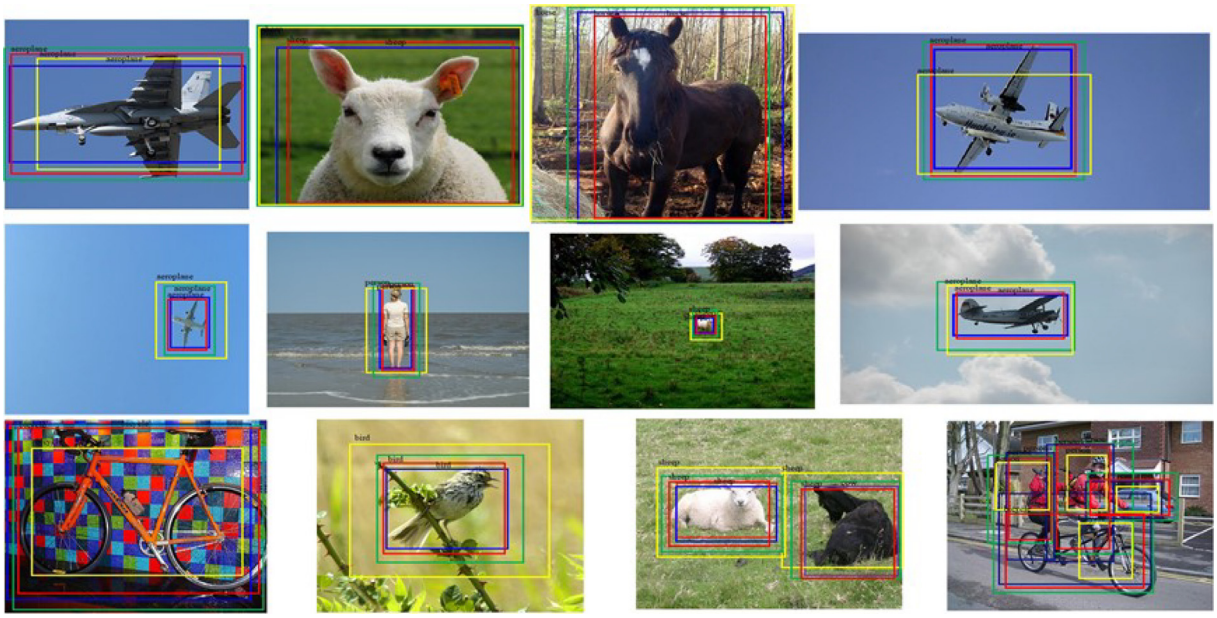


**Fig. 7.** Comparison of detection results between our ReinforceNet and other methods on VOC 2012 validation set.

**Table 6**
Effects of each component on our ReinforceNet. Results are measured with AP.

| Novel reward | IoU-net | CPL-net | BBR-net | Multiple agents | VOC 2007 | VOC 2012 |
|---|---|---|---|---|---|---|
| | | | | | 28.1 | 36.0 |
| | √ | | | | 40.1 | 42.7 |
| √ | √ | | | | 53.4 | 55.1 |
| √ | √ | | √ | | 70.3 | 65.3 |
| √ | √ | √ | √ | | 72.0 | 67.3 |
| √ | √ | √ | √ | √ | 73.7 | 68.4 |

of agent learning online. With this RL strategy, the detection model has a richer expressiveness to tackle inaccurate object location. We implement our ReinforceNet together with the promising methods on widely-used PASCAL VOC 2007 and VOC 2012 object detection benchmarks. The empirical results show that our ReinforceNet achieves state-of-the-art performance.

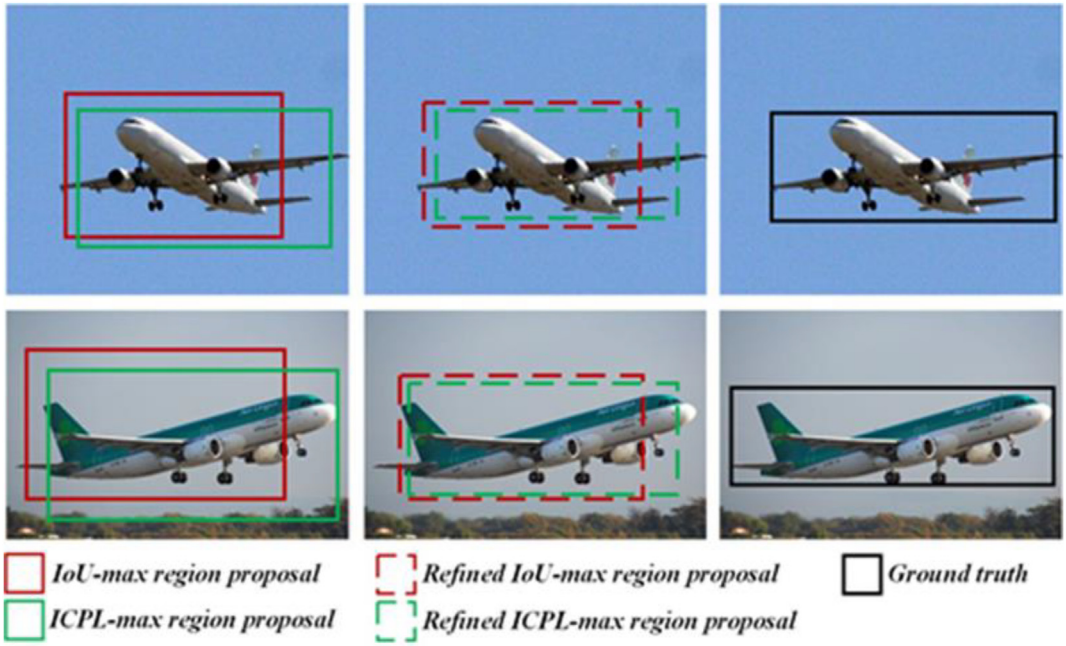



**Fig. 8.** Illustration of the effect of CPL-net.

## CRediT authorship contribution statement

**Man Zhou:** Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing. **Rujing Wang:** Supervision. **Chengjun Xie:** Supervision, Writing - review & editing. **Liu Liu:** Supervision, Writing - review & editing. **Rui Li:** Software, Validation. **Fangyuan Wang:** Software, Validation. **Dengshan Li:** Software, Validation.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgement

This work was supported by National Natural Science Foundation of China under grant 31401293, 31671586, 61773360.

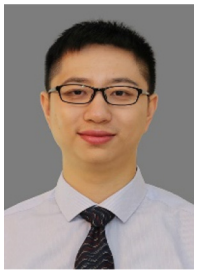
## References

[1] D. Li, H. Wu, J. Zhang, K. Huang, A2-RL: aesthetics aware reinforcement learning for image cropping, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, https://doi.org/10.1109/cvpr.2018.00855.
[2] X. Chang, P. Huang, X. Liang, Y. Yang, A. Hauptmann, RCAA: relational context-aware agents for person search, in: European Conference on Computer Vision, 2018, https://doi.org/10.1007/978-3-030-01240-3_6.
[3] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, D. Lin, Libra R-CNN: towards balanced learning for object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, https://doi.org/10.1109/cvpr.2019.00091.
[4] Z. Cai, N. Vasconcelos, Cascade R-CNN: delving into high quality object detection, in: IEEE International Conference on Computer Vision, 2018, https://doi.org/10.1109/cvpr.2018.00644.
[5] X. Han, H. Liu, F. Sun, X. Zhang, Active object detection with multistep action prediction using deep Q-network, IEEE Trans. Ind. Inf. 15 (2019) 3723–3731, https://doi.org/10.1109/tii.2019.2890849.
[6] J.C. Caicedo, S. Lazebnik, Active object localization with deep reinforcement learning, in: IEEE International Conference on Computer Vision, 2015, https://doi.org/10.1109/iccv.2015.286.
[7] Z. Jie, X. Liang, J. Feng, X. Jin, W. Lu, S. Yan, Tree-structured reinforcement learning for sequential object localization, Neural Inf. Process. Syst. (2016).
[8] M. Bellver, X. Giro-i-Nieto, F. Marques, J. Torres, Hierarchical object detection with deep reinforcement learning, Neural Inf. Process. Syst. (2016).
[9] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, Neural Inf. Process. Syst. (2015), https://doi.org/10.1109/tpami.2016.2577031.
[10] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, https://doi.org/10.1109/cvpr.2016.91.
[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A.C. Berg, SSD: single shot multibox detector, in: European Conference on Computer Vision, 2016, https://doi.org/10.1007/978-3-319-46448-0_2.
[12] P. Abbeel, A.Y. Ng, Apprenticeship Learning via Inverse Reinforcement Learning, in: International Conference on International Conference on Machine Learning, 2004, https://doi.org/10.1145/1015330.1015430.
[13] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, https://doi.org/10.1109/cvpr.2014.81.
[14] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, A.W.M. Smeulders, Selective search for object recognition, Int. J. Comput. Vision 104 (2013) 154–171, https://doi.org/10.1007/s11263-013-0620-5.
[15] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533, https://doi.org/10.1038/nature14236.
[16] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Neural Inf. Process. Syst. (2012), https://doi.org/10.1145/3065386.
[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing Atari with deep reinforcement learning, Neural Inf. Process. Syst. (2013).
[18] Y. Li, K. Fu, H. Sun, X. Sun, An aircraft detection framework based on reinforcement learning and convolutional neural networks in remote sensing images, Remote Sens. 10 (2018) 243, https://doi.org/10.3390/rs10020243.
[19] C.L. Zitnick, P. Dollar, Edge boxes: locating object proposals from edges, in: European Conference on Computer Vision, 2014, https://doi.org/10.1007/978-3-319-10602-1_26.
[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[21] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations, 2015.
[22] J.L. Ba, V. Mnih, K. Kavukcuoglu, Multiple object recognition with visual attention, International Conference on Learning Representations, 2015.
[23] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in: IEEE International Conference on Computer Vision, 2017, https://doi.org/10.1109/iccv.2017.324.
[24] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, https://doi.org/10.1109/cvpr.2016.90.
[25] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: European Conference on Computer Vision, 2014, https://doi.org/10.1007/978-3-319-10578-9_23.
[26] R. Girshick, Fast R-CNN, in: IEEE International Conference on Computer Vision, 2015, https://doi.org/10.1109/iccv.2015.169.
[27] Z. Zhou, M.M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, UNet++: a nested U-Net architecture for medical image segmentation, in: Deep Learning in Medical Image Analysis, 2018, https://doi.org/10.1007/978-3-030-00889-5_1.
[28] J. Deng, W. Dong, R. Socher, L.J. Li, L. Kai, F.F. Li, ImageNet: a large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, https://doi.org/10.1109/cvpr.2009.5206848.
[29] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge2007 (VOC 2007) Results, 2007.
[30] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge2012 (VOC 2012) Results, 2012.
[31] A. Garcia-Garcia, S. Orts-Escolano, S.O. Oprea, V. Villena-Martinez, J. Garcia-Rodriguez, A review on deep learning techniques applied to semantic segmentation, IEEE Trans. Pattern Anal. Mach. Intell. (2017).
[32] B. Uzkent, C. Yeh, S. Ermon, Efficient object detection in large images using deep reinforcement learning, IEEE Winter Conference on Applications of Computer Vision, 2020.
[33] W. Wu, D. He, X. Tan, S. Chen, S. Wen, Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition, IEEE International Conference on Computer Vision, 2019.
[34] D. Zhang, J. Han, L. Zhao, T. Zhao, From discriminant to complete: reinforcement searching-agent learning for weakly supervised object detection, IEEE Trans. Neural Networks Learn. Syst. (2020).
[35] J. Han, L. Yang, D. Zhang, X. Chang, X. Liang, Reinforcement cutting-agent learning for video object segmentation, IEEE Conference on Computer Vision and Pattern Recognition, 2018.
[36] G. Vecchio, S. Palazzo, D. Giordano, F. Rundo, C. Spampinato, MASK-RL: multi-agent video object segmentation framework through reinforcement learning, IEEE Tran. Neural Networks Learn. Syst. (2020).
[37] M. Jiang, T. Hai, Z. Pan, H. Wang, Y. Jia, C. Deng, Multi-agent deep reinforcement learning for multi-object tracker, IEEE Access (2019).
[38] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A.C. Berg, SSD: single shot multi-box detector, European Conference on Computer Vision, 2016.
[39] Y. Wang, L. Zhang, L. Wang, Z. Wang, Multitask learning for object localization with deep reinforcement learning, IEEE Trans. Cogn. Dev. Syst. (2019).
[40] J. König, S. Malberg, M. Martens, S. Niehaus, A. Krohn-Grimberghe, A. Ramaswamy, Multi-stage reinforcement learning for object detection, in: Computer Vision Conference, 2019, https://doi.org/10.1007/978-3-030-17795-9 _ 13.
[41] J. Ke, Z. Zhang, Y. Yingze, Image registration optimization mechanism based on reinforcement learning and real time denoising, Multimedia Tools Appl. (2019).
[42] K. Yu, C. Dong, L. Lin, C., Change Loy, Crafting a toolchain for image restoration by deep reinforcement learning, IEEE Conference on Computer Vision and Pattern Recognition, 2018.

**Man Zhou** received the B.E degree in automation from Wuhan University of Science and Technology, Wuhan, China in 2013. He is currently pursuing the Ph.D. degree in Pattern Recognition and Intelligent System from University of Science and Technology of China, Hefei, China. His current research interests include deep reinforcement learning and computer vision.

**Rujing Wang** received the B.E. degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 1987, and M.S. degree in electronic engineering from Dalian University of Technology, Dalian, China, in 1990, and Ph.D. degree in pattern recognition and intelligent system from University of Science and Technology of China, Hefei, China, in 2005. He is currently with the Institute of Intelligent Machinery of the Chinese Academy of Sciences as Professor and Researcher. His main research interests include intelligent agriculture, agricultural internet of things, Agricultural knowledge engineering.

**Chengjun Xie** received the M.S. degree in software engineering from the Hefei University of Technology, Hefei, China, in 2008, and Ph.D. degree in image processing from in the Hefei University of Technology, Anhui, China, in 2011. He is currently working in the Institute of Intelligent Machinery of the Chinese Academy of Sciences as Associate Researcher. His research interests include crop disease and pest image recognition, agricultural big data, agricultural Internet of Things.



**Liu Liu** received the B.E. degree in information engineering aerospace information applications from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2015, and M.S. degree in advanced computer science from University of Manchester, Manchester, United Kingdom, in 2016. He is currently pursuing the Ph.D. degree in pattern recognition and intelligent system with University of Science and Technology of China, Hefei. His current interest is deep learning and computer vision.

**Fangyuan Wang** received the B.E. and M.S. degrees in electrical engineering and automation from Hefei University of Technology, Hefei, China, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree in pattern recognition and intelligent system with University of science and technology of China, Hefei. His current interest is deep learning and computer vision.

**Rui Li** Received the B.E. degree in Hebei University of Architecture, China, in 2010, and M.S. degree in computer applied technology from Hefei University of Technology, Hefei, China, in 2013 respectively. He is currently pursuing the Ph.D. degree in electronic information with University of science and technology of China, Hefei. His current interest is deep learning and computer vision.

**Dengshan Li** receieved the B.E. degree in electronic science and technology from Hefei University of Technology, Hefei, China, in 2008, and M.S. degree in computer science and technology from Anhui University of Technology, Ma'anshan, China, in 2017 respectively. He is currently pursuing the Ph.D. degree in computer application technology with University of Science and Technology of China, Hefei. His current interest is deep learning and computer vision.