Parallel Search by Reinforcement Learning for Object Detection

Ye Huang, Chaochen Gu(✉), Kaijie Wu, and Xinping Guan

Key Laboratory of System Control and Information Processing, MOE of China,
Shanghai Jiao Tong University,
Shanghai 200240, China
{lutein,jacygu,kaijiewu,xpguan}@sjtu.edu.cn

Abstract. Object detection algorithms generally search through extensive potential areas without considering spatial correlations. To fully utilize rich information contained in high-level image features, a hierarchical object detection method with parallel search formulated as Markov Decision Process, is presented. Starting from independent initial locations, our model generates adequate region proposals by Reinforcement Learning (RL) method for subsequent refinement of bounding boxes. An attention-based state initialization algorithm combined with a novel reward function for RL training are proposed to facilitate the agent's control over window transformations. Following a coarse-to-fine detection strategy, we adopt adjustable action parameters and perform profound refinement for the generated proposals. Compared with existing detection algorithms, experiments on PASCAL VOC 2007 & 2012 dataset indicate the proposed model achieves encouraging object detection performance with fewer proposals generated.

Keywords: Object detection · Reinforcement learning
Attention mechanism · Neural network

1 Introduction

Existing object detection methods typically adopt a region proposal strategy combined with a classifier to predict the objectness scores of attended areas. Region proposal algorithms always generate excessive windows to capture multiple objects in all scales first (using image segmentation, sliding windows or fixed grids), then utilize image features to reduce the number of potential regions. Low-level features such as color and texture (Selective Search), edge (Edge Boxes [23]), gradient (HoG features used by DPM [5]), high-level features such as Fully Convolutional Network feature maps prove effective in providing adequate positive regions for subsequent classifiers. R-CNN [7] generates about 2,000 windows on raw images through Selective Search method. Fast R-CNN [6] improves the speed by applying traditional image segmentation algorithms on feature maps, decreasing overlapping windows and repetitive computation of features. Faster

© Springer Nature Switzerland AG 2018
J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11259, pp. 272–283, 2018.
https://doi.org/10.1007/978-3-030-03341-5_23

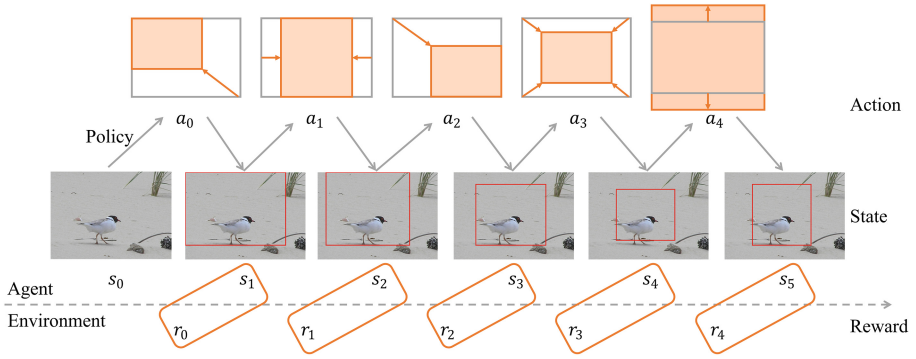


Fig. 1. At each step, the agent chooses one specific action from available predefined transformations based on current state, and gathers the next state deterministically after action taken.

R-CNN [18] and a similar approach, SSD [13] utilize sliding windows with several scales named "anchors" on different layers of feature maps. YOLO [17] segments images into grids of fixed size and detects objects on those grid segments.

Both traditional and deep learning based region proposal methods are capable of generating massive candidate areas. Nevertheless, these predefined sizes, fixed grids or time-consuming sliding windows bring in unnecessary regions that become a burden to consecutive classifiers. The independent generation operation ignores the sequential location correlations contained in extracted areas, which is incompatible with human perception procedure. Humans normally recognize multiple objects across complicated backgrounds quickly and precisely. Research on biological visual attention suggests that the visual system in humans uses attention mechanisms to focus on specific areas of the visual input [3]. Further studies [15] suggest that optimal eye movement strategy integrates the information of the whole visibility map and successively searches for fixation locations. Similar to this attention-based search procedure, the proposed method utilizes an attention map to locate salient areas coarsely, and formulates the following search movement as an agent transforming detection windows according to a RL-optimized policy.

In this paper, an effective object detection pipeline that directs a reinforcement learning agent to explore potential target areas on feature maps from fine designed initial locations, is disclosed. With less candidate regions proposed, this hierarchical detection method combines sequential object bounding box search with attention-based model to detect objects in multiple scales accurately. Under the guidance of an optimal policy learned in RL, the model utilizes a parallel candidate region generation method. This means the entire search procedure starts at differing initial locations and adopts adaptive exploration strategies. Objectiveness scores of all attended regions are then predicted to generate adequate positive proposals, which will be fed into classification and regression layers to be

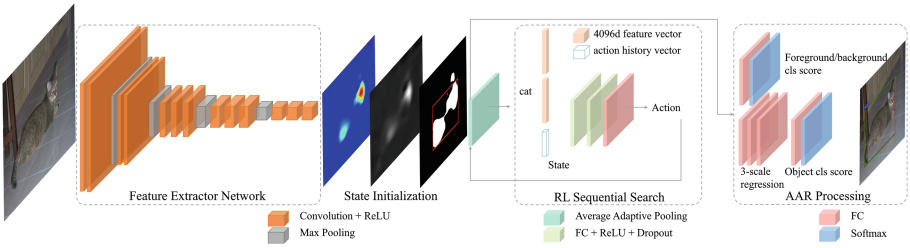


Fig. 2. Architecture of proposed pipeline. Including three main components: attention-based state initialization, parallel search with reinforcement learning and all attended regions (AAR) processing.

better evaluated and refined. The final detection results including the bounding boxes and classes of target objects are given by the refinement network.

Similar works [1,2,10] that use reinforcement learning to optimize detection process all adopt fixed search step size and predefined initial location, resulting in the lack of transformation flexibility. While in our method, an adaptive step size related to current window scales and attention-based initial search position improve the performance of an agent. Also, differing from previous method [10], the proposed reward function reflects the value of an agent's actions in a more reasonable way, driving the agent to make more informed judgements on actions selected. [2] and [1] mask searched regions with settled shapes such as a black cross, which impedes the agent detecting potentially smaller objects contained within masked areas. In contrast, by not limiting the accessibility of an agent but allowing it to exploit all attended regions by feeding them all into subsequent refinement networks.

Our pipeline is explained in detail in Sect. 2. Experiments on PASCAL VOC [4] dataset demonstrate that our method achieves competitive performance compared with similar RL methods. Comparisons with other region generation algorithms demonstrate there are fewer regions involved in the proposed model. A more detailed ablation study and an analysis of experimental results are presented in Sect. 3.

2 Approach

As shown in Fig. 2, the proposed approach includes three main components; (1) attention-based state initialization, (2) parallel search with reinforcement learning, and (3) all attended regions (AAR) processing. The method starts from the global image and local salient areas simultaneously, then performs zoom in/out exploration to locate large scale targets and local exploitation to find small objects. Based on image features and previous search path, the agent of RL balances the exploration of uncovered areas and exploitation among the discriminative regions. Thus the semantic correlation and relevant spatial information contained in feature maps may be fully utilized. All attended regions

are then classified into positive or negative samples according to their foreground/background scores, with processed positive regions selected as proposals for subsequent object classification and bounding box regression to refine the results.

2.1 Attention-Based Initialization Strategy

Two individual schemes are implemented to approximate the initial state; a predetermined region and an attention-based location. These two initialization strategies are followed by different action groups described in Sect. 2.2.

In the experiment, this predetermined region in the first stream is designed as the whole image for better analysis of global information. Corresponding transformation groups are mainly composed of large scale zoom in/out actions (Fig. 1). The fixed location is computationally efficient, while it generally takes more steps to reach targets.

In order to reduce the number of steps needed for reaching ideal destination, we adopt Grad-CAM [16] method for the second stream to coarsely extract an attention-based location, which directs the agent to focus on discriminative regions. Based on CAM [22], Grad-CAM works on the explanation and visualization of deep neural networks, producing heat-map that illustrates salient areas of an image. Gradients computed, with respect to feature maps, are then forwarded through Global Average Pooling (GAP) layer to obtain the weights for the heat-map. A binary map processed from the heat-map is then used to compute the minimal circumscribed rectangles as initial locations (Fig. 3). The subsequent action group consists of translations with relatively small step size.

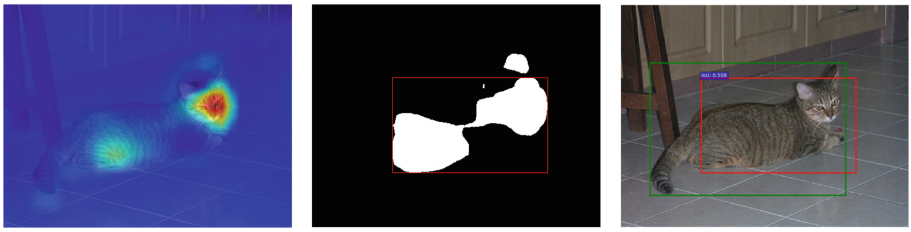


Fig. 3. Preprocessing of attention-based initialization. The image is first fed into a pre-trained feature extraction network to obtain the heat-map according to Grad-CAM method. Then the minimal circumscribed rectangles in the binary map of attained heat-map, are computed.

A more effective localization for Grad-CAM state initialization, involves training a feature extraction network from scratch based on VGG [19] model. This is also used to obtain state vectors of RL and feature vectors for later region proposal classification task. More basic models like ResNet [8] and AlexNet [11]

are also used in the experiment section to test performance. This extraction network is optimized using a criterion that optimizes a multi-label one-versus-all loss based on Binary Cross Entropy between the target and the output:

$$\mathcal{L}_{cls} = -\sum_{i}(x[i] * \log(y[i]) + (1 - x[i]) * \log(1 - y[i])) \tag{1}$$

where $x[i], y[i]$ denote the i^{th} element of the model output and one-hot-encoded target.

2.2 Parallel Search with Reinforcement Learning

The process of starting from an attention-based initial location to generally zoomed-in ROIs could be interpreted as discrete stochastic control task. Figure 1 illustrates a basic RL search path. With decision making involved, the recurrent searching is modelled as MDP optimized by Deep Q-Network [14] (DQN). The search procedure directly exploits feature vectors extracted from feature map to avoid forwarding images repeatedly, (Fig. 2), with the two agents trained to perform parallel search.

In the hierarchical detecting procedure, series of bounding boxes with adjustable scales and aspect ratios are generated, which shares the spirits of Faster R-CNN and SSD. While both of them utilize exhaustive windows fixed to predefined scales, in our method, these scales are flexible and optimized during training, with fewer candidate regions produced and time consumed. The parallel detection procedure is depicted in Fig. 4. **Action**, **State**, **Reward** and **Q-learning algorithm** are illustrated in details as follow:

Action: Similar to human perception procedure, applicable bounding box transformation contains two main branches: translation to move the window horizontally or vertically, scaling to change aspect ratio. In addition, a special action that indicates the termination of search is defined to prevent the agent from being trapped in endless detection.

The location of a bounding box is illustrated by the coordinates of its two diagonal vertices $[x_1, y_1, x_2, y_2]$, all the translation moves then could be described as increasing or decreasing corresponding values without changing aspect ratio, and scaling moves are formulated as modifying the width or height individually with centric coordinate fixed to $(\frac{x_2-x_1}{2}, \frac{y_2-y_1}{2})$ and update:

$$w = (x_2 - x_1) * \alpha_w, h = (y_2 - y_1) * \alpha_h \tag{2}$$

α_w, α_h are related to current width and height, $\alpha_w = \beta * w_c, \alpha_h = \beta * h_c$.

Larger box moves with bigger scale factor could quickly localize objects in uncovered areas, then these potential regions are explored more carefully using smaller scale factors to refine. This zoom-in-out strategy facilitates the search for objects in various scales.

State: The state is designed to cover local feature vector, global feature vector and history of actions gathered from the beginning of the search sequence. Global

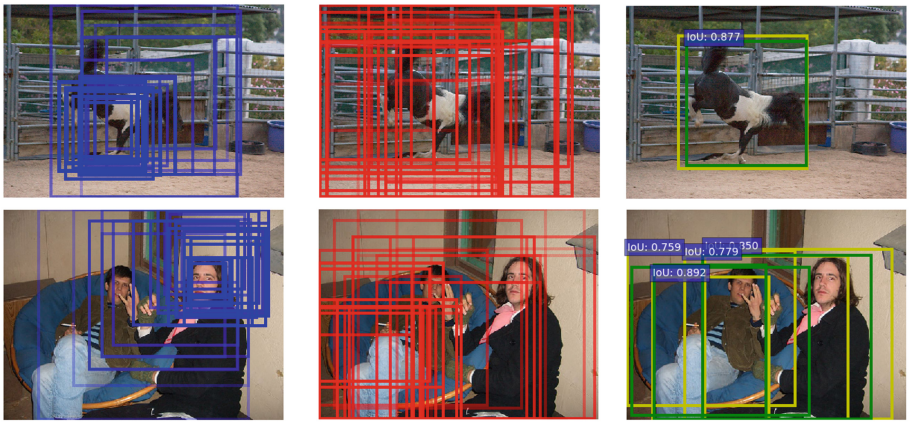


Fig. 4. Two searching paths are utilized to focus on both global and discriminative regions. Blue windows are produced by global search with mainly rescaling actions, and red windows start from salient areas. Deeper color represents later steps of the whole search path. The result shows that the jointly search could detect multiple objects with high precision. (Color figure online)

features related to the whole image provide the agent with rich visual content of potential areas that may serve as essential guidance for later exploration. Similarly, local features corresponding to current observed region utilize high-level image features to sufficiently exploit the context and spatial correlation information. State contains all relevant information from history [20], thus history vector is consist of one-hot action vectors that have been taken. As depicted in Fig. 2, feature vectors are directly obtained from an adaptive average pooling layer that follows the last convolutional layer. Before calculating state vector, boxes are mapped back to image coordinate to remove tiny and cross-boundary ones using method proposed by [12].

Reward: Focusing on goal-directed learning, we adopt proportion of intersection area and union area (IoU) to measure the value of actions and design an intuitive reward function that directs the agent to pay more attention to those actions that move current window closer to targets.

Work [1,2,10] both use a binary reward function that gives $+1$ for actions improving IoU and -1 for those decreasing it. To quantify the value of actions more distinctly, the newly developed reward function is proportional to the improvement of IoU rather than the sign of it. Actions that move current window towards targets closer are given higher rewards corresponding to the amount of change. For translation and scaling actions, the value of reward signal $R(s, a)$ after taking action a under state s can be computed by:

$$R = \begin{cases} k \cdot (\text{IoU}(b', g) - \text{IoU}(b, g)), \text{if IoU } (b', g) > \text{IoU}(b, g) \\ \text{sign}(\text{IoU}(b', g) - \text{IoU}(b, g)), \text{else} \end{cases} \tag{3}$$

where g, b, b' denote the ground truth bounding box, current window, next window after action taken and k represents the scale factor for positive actions.

Terminal action does not transform current window but records its location and restarts new refinement, which is beneficial to limit the sequential search within an adaptive number of steps. Reward for terminal action is:

$$R = \begin{cases} +\eta, \text{if IoU}(b, g) > \text{threshold} \\ -\eta, \text{else} \end{cases} \tag{4}$$

Q-Learning: Q-learning is an off-policy Temporal Difference control algorithm that learns to optimize the long-term cumulative reward by searching for an optimal policy function. Q function is trained to predict the value given current state s and action a. We use a deep neural network to approximate the Q function:

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a') \tag{5}$$

The loss function of Deep Q-network is Mean Squared Error Loss (MSE Loss) of target Q and the output of Q-network. To guarantee the convergence, experience replay buffer and asynchronous update with target network are also adopted in our experiment. Since we use two initialization schemes, there are two corresponding policy models to be trained independently.

2.3 All Attended Regions Processing

The successively explored regions via sequential attention patterns are categorized as positive and negative samples according to their IoU with ground-truth boxes, the assignment criterion is the same with Faster R-CNN. To avoid degeneration of model due to the imbalance between positive and negative samples, we keep all the positive samples and randomly sample negative regions to restrain the ratio to 1:3. In case there are insufficient positive samples, additional windows generated by random combination of current regions are also fed into classification network.

All attended regions in various scales are labelled and then scored. The classification network utilizes Smooth L1 Loss to evaluate the possibility of a region being foreground. To restrain windows with high overlap, we use nonmaximum suppression (NMS) based on the first classification score.

Three regressors are trained for regions with different aspect ratios: $1:1$, $1:a$ and $a:1 (a > 1)$. A multi-task loss function for both object classification and bounding box regression is used at the end of Fig. 2.

3 Experiment

Extensive experiments with different neural networks are performed on PASCAL VOC 2012 dataset which contains about 20k images (train + validation + test) of 20 object categories. The detection results are evaluated using mean

Average Precision (mAP) as this metric reflects both accuracy and generalization of model, which is widely applied in object detection task. As a region proposal method, our model is compared with existing region generation algorithms. Comparative experiments to other RL-based object detection methods are conducted by analyzing the detection results. To validate the effectiveness of our parallel strategy, ablation experiments are performed carefully: we analyze the two branch of initialization and searching policy individually, in addition, the consumption of action steps to convergence is also evaluated.

Experimental Details: The output size of adaptive average pooling layer is 2×2 to obtain 2048d feature vector. We use 5 composite scaling actions for global search and 8 translating actions for saliency search agent. 20 steps of actions are encoded as history vector to be concatenated with two 2048d feature vectors, accounting for the full state. The maximum step length is set to be 36, with 3 additional window scales $h : w, w : w, h : h$ adopted as alternative regions at each step of parallel search path. Threshold for terminal action's reward is set to be 0.6, and we choose $\beta = 5/6, \eta = 5.0, k = 10.0$.

Policy training strategy is $\epsilon - greedy$ with ϵ decreasing from 1.0 to 0.1 linearly during the first 9 epochs, and then fixed to 0.1 in last 41 epochs to ensure that the agent keeps balance between exploration and exploitation. Experience replay buffer has the size of 10,000 and the batch size of 64, discount factor γ for Q function is 0.9. SGD back-propagation method is utilized with momentum value fixed to 0.9 during training.

Threshold of NMS in background/foreground classification is 0.7, then we change it to 0.3 when dealing with the final output. Since in attention-based state initialization, the classification mean Average Precision (mAP) of feature extractor network has reached 83.7, we directly use the parameters of its classifier as pre-trained model for object classification.

Detection Results: mAP results evaluated on PASCAL VOC 2007 dataset are shown in Table 1. Given results by [9], Fast R-CNN models that utilizes BING [12], EdgeBoxes [23], Selective Search [21] as region proposal methods respectively, R-CNN, Faster R-CNN, YOLO and SSD models, as well as other RL-based detection models are presented in the table. We adopt VGG16 with RoI pooling as basic extraction structure. Our model has achieved relatively higher mAP score with much more fewer candidate regions generated compared with other algorithms.

Comparison of the recall rates between other region proposal generation methods is shown in Fig. 5(a). It's worth mentioning that though these approaches are traditional, they are combined with Fast R-CNN structure to evaluate the results. This experiment demonstrates that our model can achieve a similar recall rate as current proposal generation algorithms while we exploit a significantly smaller number of candidate regions. When IoU threshold varies within [0.5,1], the recall of our model performs relatively promising among other methods.

Table 1. Detection mAP of existing methods on VOC07 test set. All methods use VGG16 structure, and are trained on VOC07 except work [10], it's trained on VOC07+12 trainval set and adopts Fast R-CNN(ResNet101) as subsequent network.

Method	Data	mAP (%)	Proposals
Ours(VGG16)	07	69.4	200
FRCNN+Bing [12]	07	49.0	1k
FRCNN+EdgeBoxes [23]	07	60.4	1k
FRCNN+SelectiveSearch [21]	07	59.5	2k
R-CNN [7]	07	54.2	2k
RPN+VGG [18]	07	69.9	300
SSD300 [13]	07+12	68.0	8k
YOLO [17]	07	66.4	49
RL-based [10]	07+12	76.6	Not provided
RL-based [2]	07	46.1	4k

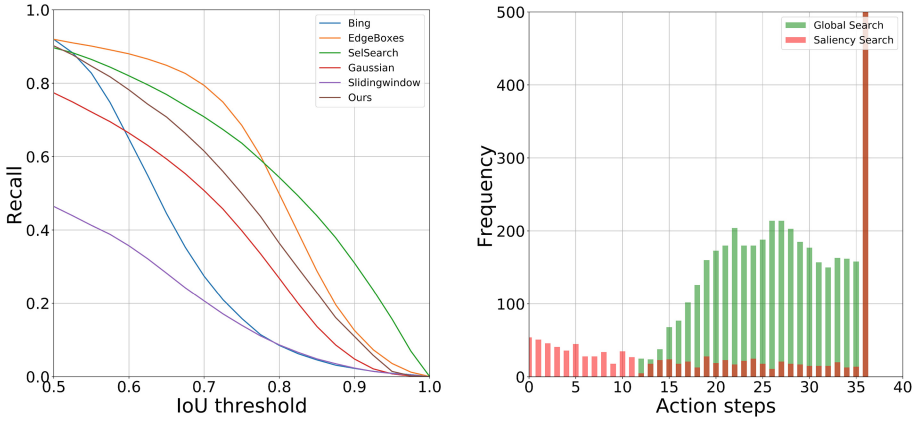
Analysis of Action Steps: As depicted in Fig. 5(b), the saliency search agent normally requires fewer action steps to detect possible regions. In most cases, both agents consume all 36 steps, generating about 100 proposals. Though R-CNN family remains state-of-the-art method, Faster R-CNN produces about 2k proposals generally. We found that the selection of action groups is flexible because it has a negligible effect on the final result. Specifically, appropriate combination of these two basic branches could strengthen the agent's ability of exploration, and composite actions generally reduce the number of requisite steps to locate targets.

Ablation Experiment: To investigate the performance of our parallel strategy, we conduct several ablation experiments including different extraction network shown in Table 2 and individual test of saliency search and global search.

Table 2. Classification score and detection mAP of different basic extraction networks on PASCAL VOC 2012 data set.

Method	Mean accuracy (%)	Val score (%)	Test score (%)	mAP (%)
VGG16+RoI pooling	69.71	75.1	82.1	69.4
ResNet18 [12]+resize	58.20	54.9	45.0	58.6
AlexNet [23]+resize	39.55	44.36	35.80	46.1
VGG16 [21]+resize	65.55	72.50	71.75	65.5

We adopt different basic feature extraction network to evaluate their performance. This basic network is applied to extract feature vectors that can be used



(a) (b)

Fig. 5. (a) is the comparison of recall with other region proposal generation methods evaluated on PASCAL VOC 2007 test data. (b) shows the distribution of the number of action steps used in parallel search procedure.

to calculate state and generate grad-cam initial areas, thus we treat it as a typical classification task using multi-label classification loss function and stochastic gradient descent to optimize. Different deep neural networks are trained from scratch under similar training process.

Table 2 shows the classification scores and corresponding detection results evaluated on both validation and test set, with test scores provided by official PASCAL VOC evaluation server. The input image of three basic models are resized to 224×224 and augmented with randomly horizontal flip. We further analyze the performance of model VGG16 by alternating image resize to feature map adaptive pooling (RoI pooling). Model that adopts adaptive pooling achieves better classification and detection results than those use image resize, since image resize would lost more information than adaptive pooling.

Table 3 demonstrates the detection results of the two parts of parallel strategy separately. Each search is capable of detecting some objects while the combined parallel search achieves better performance. Given global information at the beginning, global search can locate more objects than saliency search does since the latter focuses on relatively smaller areas.

Table 3. Detection mAP of individual saliency search and global search strategies.

Method	Data	mAP (%)	Average steps
Parallel	07	69.4	31
Saliency search	07	30.9	30
Global search	07	48.2	33

4 Conclusion

A parallel search pipeline using reinforcement learning to generate distinctive and accurate region proposals for object detection has been disclosed. Under the framework of parallel policies including global search and saliency search, the trained RL agents can perform adaptive transformations within limited number of steps to generate adequate and high-quality candidate regions. The attention-mechanism-guided search empowers our method to locate salient objects and explore all possible objects across relatively larger areas. Compared with existing RL-based detection algorithms and region proposal generation methods, the method implemented with a refinement network is more effective in locating and classifying target objects. This is evidenced by the promising detection results on PASCAL VOC dataset with much fewer generated region proposals.

References

1. Bellver, M., Giró-i-Nieto, X., Marqués, F., Torres, J.: Hierarchical object detection with deep reinforcement learning (2016)
2. Caicedo, J.C., Lazebnik, S.: Active object localization with deep reinforcement learning. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), pp. 2488–2496. IEEE (2015)
3. Carrasco, M.: Visual attention: the past 25 years. Vis. Res. **51**, 1484–1525 (2011)
4. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**, 303–338 (2010)
5. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8. IEEE (2008)
6. Girshick, R.: Fast R-CNN (2015)
7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
9. Hosang, J., Benenson, R., Dollr, P., Schiele, B.: What makes for effective detection proposals? IEEE Trans. Pattern Anal. Mach. Intell. **38**, 814–830 (2016)
10. Jie, Z., Liang, X., Feng, J., Jin, X., Lu, W., Yan, S.: Tree-structured reinforcement learning for sequential object localization. In: Advances in Neural Information Processing Systems, pp. 127–135 (2016)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
12. Lenc, K., Vedaldi, A.: R-CNN minus R, vol. abs/1506.06981 (2015). http://arxiv.org/abs/1506.06981
13. Liu, W., et al.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part I. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2

14. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**, 529–533 (2015)
15. Najemnik, J., Geisler, W.S.: Optimal eye movement strategies in visual search. Nature **434**, 387–391 (2005)
16. Ramprasaath, R., Abhishek, D., Ramakrishna, V., Michael, C., Devi, P., Dhruv, B.: Grad-CAM: why did you say that? Visual explanations from deep networks via gradient-based localization (2016)
17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
18. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014)
20. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, vol. 1. MIT Press, Cambridge (1998)
21. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. Int. J. Comput. Vis. **104**, 154–171 (2013)
22. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2921–2929. IEEE (2016)
23. Zitnick, C.L., Dollár, P.: Edge boxes: locating object proposals from edges. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 391–405. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_26