RefinedBox: Refining for Fewer and High-quality Object Proposals

Yun Liu, Shi-Jie Li, Ming-Ming Cheng*

CCS, Nankai University, Tianjin, P.R.China, 300350

Abstract

Recently, object proposal generation has shown value for various vision tasks, such as object detection, semantic instance segmentation, multi-label image classification, and weakly supervised learning, by hypothesizing object locations. We are motivated by the fact that many traditional proposal methods generate dense proposals to cover as many objects as possible but that i) they usually fail to rank these proposals properly and ii) the number of proposals is very large. For example, the well-known object proposal generation methods, Edge Boxes and Selective Search, can achieve high detection recall with thousands of proposals per image. But the large number of generated proposals makes subsequent analyses difficult due to the large number of false alarms and heavy computation load. To significantly reduce the number of proposals, we design a computationally lightweight neural network to refine the initial object proposals. The refinement consists of two parallel processes, re-ranking and box regression. The proposed network can share convolutional features with other high-level tasks by joint training, so the proposal refinement can be very fast. We show a joint training example of object detection in this paper. Extensive experiments demonstrate that our method can achieve state-of-the-art performance with a few proposals compared with some well-known proposal generation methods.

Keywords: Object proposals; Fewer proposals; Mining proposals

1. Introduction

Generating *a small number of* object proposals while covering as many objects in an image as possible is crucial for the efficiency and accuracy of subsequent high-level applications, such as object detection [1, 2], instance semantic segmentation [3, 4], multi-label classification [5], video summarisation [6], and deep multiple instance learning [7], by reducing the search space and the false alarms. In the past decade, many bottom-up object proposal methods have been developed to generate dense proposals to cover as many objects as possible, such as Selective Search [8], Edge Boxes [9] and MCG [10]. Since it is difficult to represent high-level semantic information using traditional hand-crafted features, these bottom-up methods usually i) fail to rank the generated proposals properly and ii) have to use large number of proposals to ensure detection recall. Although these existing bottom-up algorithms can achieve high detection recall with thousands of proposals per image, the large number of generated proposals makes subsequent analyses difficult due to the large number of false alarms and heavy computation load [5, 7, 11, 12]. Recently, some deep learning based proposal methods have attracted a lot of attention in this field, including RPN [13], DeepMask [14], and SharpMask [15]. With the powerful representation capability of convolutional neural networks (CNNs), these methods can provide high detection recall with fewer candidate boxes than traditional bottom-up algorithms. However, RPN [13] generates proposals by sampling anchors from downsampled convolutional feature maps (1/16 scale), and DeepMask [14] and SharpMask [15] discover objects by scanning image patches. Hence the sub-optimal proposal sampling strategies make them difficult to fully leverage the powerful capability of CNNs. As a result, the number of true objects (*e.g.* usually less than 10) in an image is still much smaller than the number of proposals generated by these deep-based methods (*e.g.* usually a few hundred).

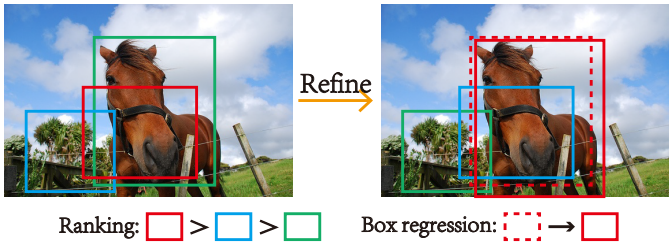


Figure 1: Overview of object proposal refinement. The left image shows the original proposals, and the right image shows the results after refinement. We first re-rank the proposals by computing new objectness scores, after which a box regression procedure is applied to each proposal box for accurate location.

Can we significantly reduce the number of proposals while maintaining the high recall? This is crucial for a much wider range of applications, *e.g.* mining knowledge from huge amounts of unlabeled/weakly-labeled data [5, 7], for which the large number of false positives will pose significant challenges not only for computational efficiency but also for system stability. Some research towards reducing the number of proposals for specific vision tasks has been proposed. For example, Wei *et al.* [5] adopted normalized cut [16] to cluster bounding boxes generated by BING algorithm [17] and picked out the top 1 hypothesis with the highest objectness score in each cluster. They applied the selected proposals to multi-label image classification and achieved the state-of-the-art performance. Qi *et al.* [11] introduced an aggregation score at each pixel by calculating the sum of all objectness scores whose correspond-

*Corresponding author: MM Cheng (cmm@nankai.edu.cn).

ing proposal boxes cover this pixel. The resulting aggregation score maps are used to estimate object locations. Li *et al.* [12] adopted a mask-out strategy to collect proposals with higher quality for each object category. A proposal is collected for one class if the mask-out image by this proposal box has a significant drop in classification score of this class.

In this paper, we focus on *mining the number of proposals* while obtaining high detection recall. We observe that some traditional proposal generation methods can achieve high detection recall when the number of candidate boxes is sufficiently large, because traditional methods usually design clever strategies to search all possible positions for objects, unlike the simple proposal sampling strategies in deep learning [13, 14, 15]. Of course, the large number of candidates causes many false positives in the subsequent applications and thus affects the final performances. However, if we can select the good ones from the large set of candidates, it will benefit a series of vision tasks. Several algorithms have been proposed to refine object proposals, including DeepBox [18] and MTSE [19]. DeepBox builds a neural network to recompute the objectness scores of the initial boxes and then re-rank them. MTSE tries to refine each box using superpixels by making each box tightly cover some inner superpixels. However, the proposal quality of DeepBox is worse than RPN [13], and thus the number of proposals can not be reduced. Moreover, the performance of MTSE depends on the quality of superpixels, and the image segmentation within MTSE causes a significant increase in computational load.

To combine the superiority of traditional proposal methods and the powerful representation capability of CNNs [13, 14, 15, 20, 21], we propose a novel method to re-rank and align existing proposal boxes in a single inference of a neural network. An overview of our approach is shown in Figure 1. Our refinement of candidate boxes includes two steps: re-ranking and box regression. The re-ranking step tries to re-rank the proposals according to the tightness of their coverage with complete objects. The box regression step attempts to fine tune the shapes and locations of boxes in order to make them cover real objects more tightly. To achieve this goal, our refinement network is designed to learn new objectness scores and perform box regression simultaneously. The proposed network is also computationally lightweight, so it can be applied to applications with little extra time consumption. The training process of refinement can be performed in an end-to-end manner. For the sake of brevity, we call our proposed method **RefinedBox** in the remainder of this paper. Since RefinedBox is lightweight and easily optimized, it has the potential to share convolutional features with high-level applications by joint training. To show a joint training example, we unify RefinedBox and the well-known detection framework of Fast R-CNN [2] by connecting our refinement layers after the last convolutional layer of the base network such as VGG16 [22], and then introduce an alternating fine-tuning strategy. As a result, our refinement network can share the base convolutional layers with the subsequent object detection network, making the refinement procedure very efficient.

Using the proposal boxes produced by various traditional methods as input, we evaluate the proposed method on the PASCAL VOC2007 [23] and MS COCO [24] datasets. For object proposal generation on the VOC2007 dataset, our method achieves the detection recall of 80.4% and 67.9% for intersection-over-union (IoU) 0.5 and 0.7, respectively, using only 10 refined boxes per image. Using only 10 boxes for object detection, our method achieves a mean average precision (mAP) of 65.4% compared with the mAP of 54.1% for RPN [13]. The experiments demonstrate that the proposed RefinedBox method can generate high-quality object proposals when the number of proposals is limited.

2. Related Work

Since this paper targets object proposal refinement, we first briefly describe recent developments in object proposal generation. We then go on to discuss the refinement techniques of bounding boxes. We broadly divide the related research into four parts: segmentation-based proposal generation methods, edge-based methods, CNN-based methods, and proposal post-processing methods.

Segmentation-based object proposal generation methods use the image segmentation as input and try to find the proper combinations of these image segments to cover all complete objects. These methods usually combine some low-level features (such as saliency, color, SIFT [25], *etc*.) to score the bounding boxes and then select boxes with high scores. Selective Search [8], one of the most popular object proposal methods, uses the strength of exhaustive search and segmentation to obtain high-quality proposals by a hierarchical merging of superpixels. MCG [10] introduces a high-performance image segmentation algorithm that makes effective use of multiscale information. The produced multiscale hierarchies of regions are combined into object proposals by exploring the combinatorial space. Manen *et al.* [26] built a connectivity graph of an image's superpixels, and generated spanning trees with large expected sum of edge weights using a randomized version of Prim's algorithm. The bounding boxes of these spanning trees are final object proposals. Rantalankila *et al.* [27] performed local search on superpixels to form a segmentation hierarchy. Then global search is applied to obtain graph cut segmentations of the intermediate hierarchy. Many other proposal generation methods [28, 29, 30] also fall into this category.

Edge-based proposal methods exploit the observation that complete objects in natural images usually have well-defined closed boundaries [31]. In recent years, several efficient algorithms have been proposed using the edge feature. Zhang *et al.* [32] designed a cascaded ranking SVM (CSVM) method to obtain proposals using gradient features. Cheng *et al.* [17] proposed a very efficient algorithm, BING, which runs at 300fps by quantizing CSVM [32] into some binary operations. Lu *et al.* [33] proposed a new closed contour measure based on the closed path integral. Edge Boxes [9] computes the objectness scores according to the number of contours that are wholly contained in each bounding box.

CNN-based proposal methods generate object proposals from CNNs directly, such as RPN [13], DeepMask [14], and Sharp-

Mask [15], inspired by the fact that CNNs have powerful capability in learning feature representations [20, 21]. RPN [13] simultaneously predicts object bounds and objectness scores at each position of full-image convolutional features. DeepMask [14] is trained jointly with two objectives: given an image patch, the system first outputs a class-agnostic segmentation mask and then outputs the likelihood of the patch being centered on a full object. SharpMask [15] propose to augment feedforward nets for object segmentation with a novel top-down refinement approach. The resulting bottom-up/top-down architecture is capable of efficiently generating high-fidelity object masks. However, the number of proposals generated by these CNN-based methods is still too many (*e.g.* usually a few hundred) for natural images.

Proposal post-processing aims to refine the object proposals in order to accurately locate objects in an image. Kuo *et al.* [18] proposed a small neural network called DeepBox to recompute the objectness scores of the existing boxes and then re-rank these boxes according to the new objectness scores. Chen *et al.* [19] tried to align the proposal boxes with the superpixels. Zhang *et al.* [34] further discussed the optimization of object proposal generation. They first used edges and then superpixels to optimize the proposal boxes. Their segmentation based optimization accelerates the superpixel generation in MTSE [19], thus the resulting system can be run at a very fast speed. He *et al.* [35] proposed oriented object proposals that have different orientations, not only the vertical boxes used in regular methods. In this paper, we build a refinement network to refine existing bounding boxes. The refined boxes produced by our method achieve the state-of-the-art performance both for object proposal generation evaluation and object detection evaluation.

3. RefinedBox

3.1. Network Architecture

Our method takes the object proposals produced by other proposal generation methods as input and then tries to refine them. The refinement is twofold: re-ranking and box regression. To re-rank the existing boxes, we recompute the objectness score for each box using the semantic information in the deep neural network. To obtain the box regression, the network is designed to learn the regressions of the center coordinates, width, and height for each box.

VGG16 [22] is a widely used base network architecture in deep learning research. It is composed of 13 convolutional layers and 3 fully connected layers. Inspired by previous literature [2, 13], we build our network based on VGG16 to showcase our refinement method. Our network architecture is shown in Figure 2. Our network takes a natural image and corresponding initial boxes as input. The initial boxes are produced by other object proposal generation methods. In this paper, we use some well-known proposal generation methods as examples, including Edge Boxes [9], MCG [10], Selective Search [8], and RPN [13]. The input image first undergoes a forward pass through some convolutional layers, *e.g.* the 13 convolutional layers in VGG16. In order to reduce the time consumption of box refinement, we design a computationally lightweight neural network. Thus, we first connect a convolutional layer with kernel size 3×3 after the 13-th convolutional layer to reduce the number of channels from 512 to 128. Then, a *ROI Pooling* layer is followed to down-sample each initial box region into a fixed feature map size, *i.e.* 7×7. *ROI Pooling* divides an input feature map into grids with the same width and height and perform max pooling in each grid. Next, a fully connected layer with only 512 output neurons is connected. A ReLU layer is followed after the added convolutional layer and fully connected layer, respectively. At last, two branches of ranking and box regression are used to recompute the objectness score and obtain the location offsets of each initial box. The ranking branch is a fully connected layer with two output neurons representing the probabilities of being an object or not. The box regression branch predicts the box regression values which will be described below.

In the training of RefinedBox, each initial box is assigned a binary class label of being an object or not. The loss function can be written as

$$L_{obj}(p, u) = -[1_{\{u=1\}} log\ p_1 + 1_{\{u\neq 1\}} log\ p_0], \tag{1}$$

where p is computed by a softmax over the two outputs of a fully connected layer and u is the label of this box (1 or 0). The box regression layer is a fully connected layer which is designed to learn the coordinate offsets. We perform the parameterizations of four coordinates as following:

$$\begin{aligned} t_x &= (x - x_{in})/w_{in}, & t_y &= (y - y_{in})/h_{in}, \\ t_w &= log(w/w_{in}), & t_h &= log(h/h_{in}), \\ v_x &= (x^* - x_{in})/w_{in}, & v_y &= (y^* - y_{in})/h_{in}, \\ v_w &= log(w^*/w_{in}), & v_h &= log(h^*/h_{in}), \end{aligned} \tag{2}$$

where x, y, w, and h represent the coordinates of the box center, width, and height, respectively. Variables x, x_{in}, and x^* are for the predicted box, input box, and ground truth box, respectively; similar definitions hold for y, w, and h. Hence variables v is the regression target and t is the predicted tuple. The box regression loss is defined as

$$\begin{aligned} L_{reg} &= \sum_{i\in\{x,y,w,h\}} \text{smooth}_{L_1}(t_i - v_i), \\ \text{smooth}_{L_1}(x) &= \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \end{aligned} \tag{3}$$

where $\text{smooth}_{L_1}(x)$ is a well-known regression loss function [2]. Thus the joint loss function can be written as

$$L(p, u, t, v) = L_{obj}(p, u) + \lambda \cdot 1_{\{u=1\}} L_{reg}(t, v), \tag{4}$$

in which the parameter λ is a balance parameter, and we set it as 1 in this paper.

3.2. Joint Training with Object Detection

So far we have described how to train the proposal refinement network. Since the proposed network is very lightweight, it has

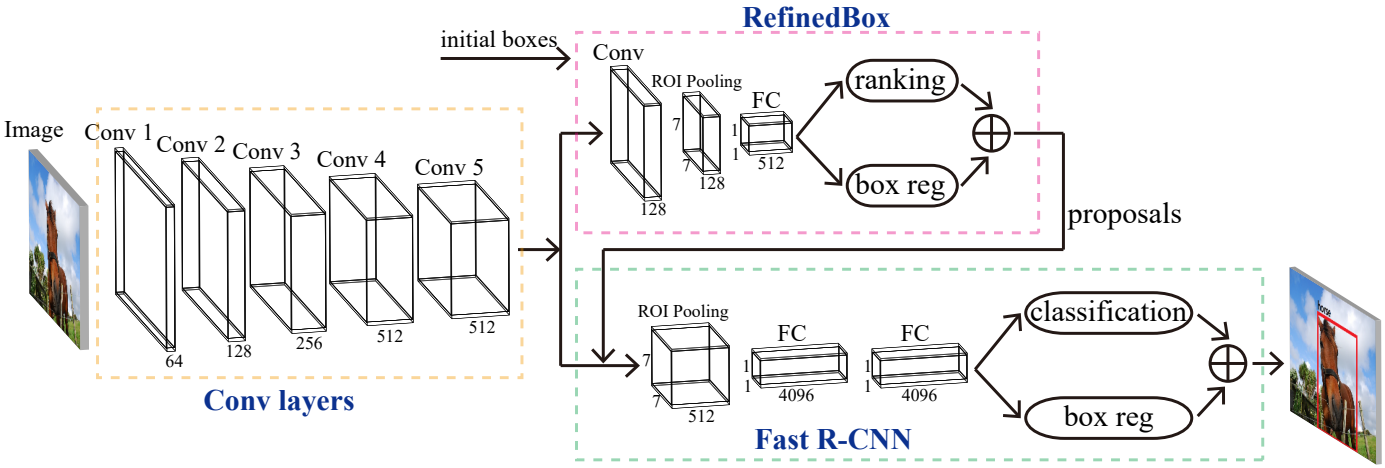


Figure 2: Overview of our network architecture. We display object detection as an example of joint training. The proposed network takes a nature image and corresponding initial boxes produced by other object proposal generation methods such as Edge Boxes as input. The branch of RefinedBox is designed to refine the initial boxes, then the refined boxes are inputted into the branch of Fast R-CNN for classification. Note that the refinement of boxes and consequent object detection can share the convolutional features.

Algorithm 1 Alternating training process of RefinedBox.

Input: the proposed network with the backbone net (W_{VGG}), RefinedBox module (W_{RB}), and detection module (W_{Det}); the initial proposals B_{in}; the backbone model (W^{pre}_{VGG}) pre-trained on ImageNet
Output: the unified network of W_{VGG}, W_{RB}, and W_{Det}
Step 1: $W_{VGG} \leftarrow W^{pre}_{VGG}$; $W_{RB} \leftarrow random()$
Step 2: $W_{VGG}, W_{RB} \leftarrow finetune(W_{VGG}, W_{RB}; B_{in})$
Step 3: $B' \leftarrow rerank(B_{in}; W_{VGG}, W_{RB})$
Step 4: $W_{VGG} \leftarrow W^{pre}_{VGG}$; $W_{Det} \leftarrow random()$
Step 5: $W_{VGG}, W_{Det} \leftarrow finetune(W_{VGG}, W_{Det}; B')$
Step 6: $W_{RB} \leftarrow random()$
Step 7: $W_{RB} \leftarrow finetune(W_{RB}; W_{VGG}, B_{in})$
Step 8: $B' \leftarrow rerank(B_{in}; W_{VGG}, W_{RB})$
Step 9: $W_{Det} \leftarrow random()$
Step 10: $W_{Det} \leftarrow finetune(W_{Det}; W_{VGG}, B')$

the potential to share convolutional features with high-level applications. Here, we use object detection as an example to show the joint training process of RefinedBox and consequent applications. In order to test the ability of RefineBox to generate a few proposals with high quality, we only use the top 10 proposals per image of RefinedBox to perform object detection.

As shown in Figure 2, we connect the well-known detection framework, Fast R-CNN [2], after the convolutional layers as a parallel branch to RefinedBox. The refined proposals produced by the RefinedBox branch are inputted into Fast R-CNN. In order to make the RefinedBox and Fast R-CNN share the same convolutional features, we apply an alternating fine-tuning process. The algorithm is presented in Algorithm 1. Object detection depends on the re-ranked proposals generated by the preceding step for training. Before step 6, object proposal and detection networks are trained separately. Then, the backbone network is fixed, and only the unique layers for RefinedBox and detection are fine-tuned. After the alternating training, both networks form a unified network.

For other high-level applications, the joint training is in the similar way. In other words, Algorithm 1 is also applicable to other tasks by replacing W_{Det} with the module for other tasks. The key of Algorithm 1 is to make the high-level task and RefinedBox share the same backbone network using an alternate training between the higher-level task and RefinedBox modules, so that an input image only needs to pass through the backbone network once.

The number of floating-point operations (FLOPs) is often used to measure the computational cost of a network, where a floating-point operation means a multiply-add operation. For each proposal box, there are 120.0 million FLOPs for the fully connected layers of the Fast R-CNN branch, while only 3.2 million FLOPs for the fully connected layers of the RefinedBox branch. Therefore, the RefinedBox branch only incurs a little extra computational load.

3.3. Implementation Details

For the training of RefinedBox, each stochastic gradient descent (SGD) mini-batch is constructed from an image in which 256 boxes are selected as training samples. In each batch, half of the sampling boxes are positive samples and the other half are negative. The intersection-over-union (IoU) means the ratio of the intersection area of two boxes over the union area. The positive sampling boxes have IoU overlaps of at least 0.7 with ground truth boxes, while the negative samples are boxes whose max IoU overlaps with ground truth are in the interval [0.1, 0.5). The initial learning rate is set to 1e-3 and will be divided by 10 after 12 epochs. We run SGD for 16 epochs in total.

For the training of the detection module, each mini-batch has 256 object proposals that are from the same image. As in Fast R-CNN [2], 25% of these proposals have IoU overlap with a ground truth of at least 0.5, and they are viewed as positive samples. The remaining negative samples have max IoU overlap with ground truth in the interval [0.1, 0.5). The top 1000

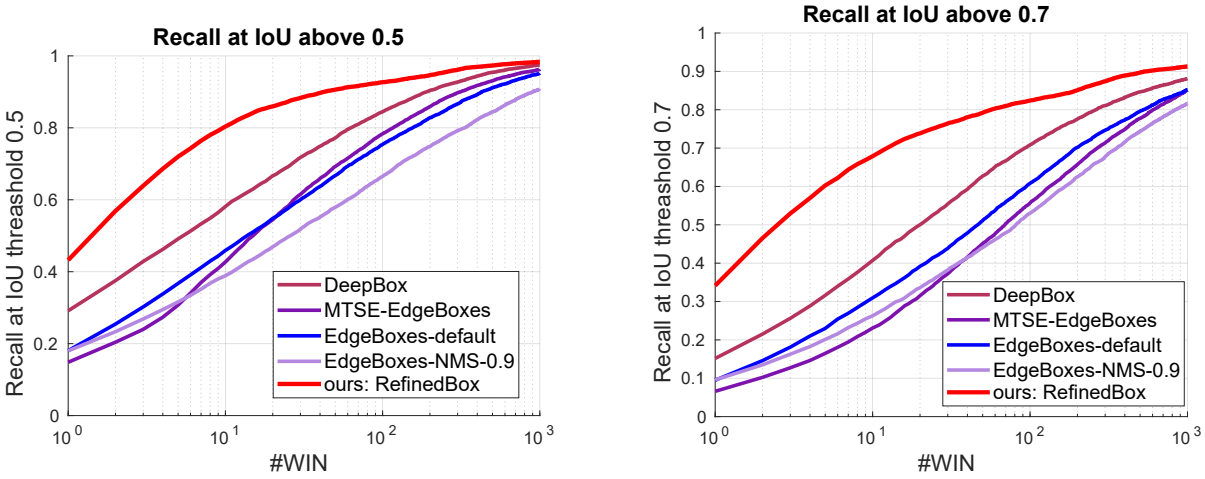


Figure 3: Evaluation of different refinement algorithms. These two subfigures show object detection recall *vs.* the number of proposals (#WIN) at IoU threshold 0.5 (left) and 0.7 (right), respectively. The method of EdgeBoxes-default is Edge Boxes [9] with default parameters, and EdgeBoxes-NMS-0.9 changes the parameter of non-maximum suppression (NMS) to 0.9.

proposals generated by RefinedBox are used in training. The learning rate is 1e-3 for the first 12 epochs, and then the learning rate is divided by 10 for another 4 epochs. For test, only the top 10 proposals (per image) of RefinedBox are used. In contrast, the traditional proposal methods, such as Edge Boxes and Selective Search, usually need thousands of proposals. We implement the proposed method based on the publicly available code[1]. The training and testing are conducted on a GTX TITAN X GPU.

4. Experiments

4.1. Experimental Setup

Datasets: We evaluate the proposed method on two widely used object detection datasets, including PASCAL VOC2007 [23] and MS COCO [24]. PASCAL VOC2007 dataset [23] is composed of 2501 training, 2510 validation, and 4952 test images with corresponding annotations across 20 object categories. We train the models on the VOC2007 *trainval* set and test on the VOC2007 *test* set. MS COCO dataset [24] consists of 82783 training images and 40504 validation images. We adopt its training set for training and its validation set for proposal evaluation.

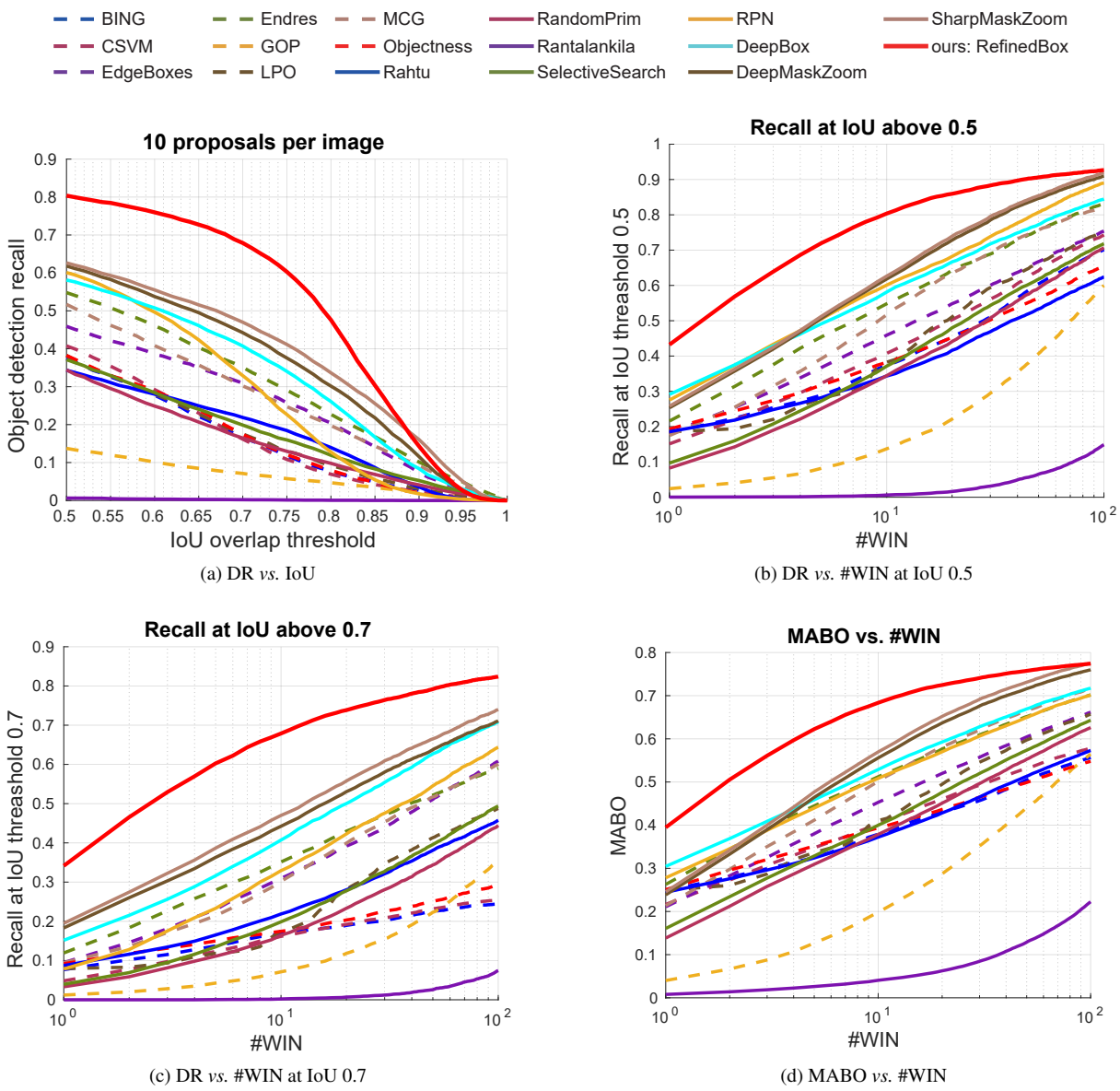
Competitors: To demonstrate the effectiveness of the proposed proposal refinement method, we compare our method with the existing mainstream proposal methods, including non-deep methods, including BING [17], CSVM [32], Edge Boxes [9], Endres [29], GoP [36], LPO [30], MCG [10], Objectness [31], Rahtu [28], RandomPrim [26], Rantalankila [27], and Selective Search [8], and recent deep learning based methods, including RPN [13], DeepBox [18], DeepMaskZoom [14], and SharpMaskZoom [15]. DeepMaskZoom and SharpMaskZoom are the best version of DeepMask [14] and SharpMask [15], respectively. We first compare with these methods for proposal evaluation. Then, for the PASCAL VOC2007 dataset [23], we feed the proposals produced by these methods into a region-based object detection framework, Fast R-CNN [2], to evaluate the quality of proposals in object detection. Our experiments demonstrate that our method can generate high-quality proposals for object detection with good efficiency.

Table 1: Evaluation results (%) in terms of DR on the PASCAL VOC2007 *test* set. RefinedBox[1], RefinedBox[2], RefinedBox[3] and RefinedBox[4] mean RefinedBox with Edge Boxes, MCG, Selective Search, and RPN respectively.

#WIN	DR (IoU=0.5)				DR (IoU=0.7)				Time
	10	30	50	100	10	30	50	100	(s)
BING	37.5	51.0	60.4	70.1	16.9	20.2	22.5	24.4	**0.003**
CSVM	40.8	56.1	64.2	74.3	16.2	20.9	23.1	25.5	0.33
EdgeBoxes	45.9	60.0	66.7	75.4	31.0	43.8	51.1	60.8	0.25
Endres	54.8	68.9	75.6	83.3	35.1	47.1	52.2	59.0	19.94
GOP	13.7	29.5	40.7	60.0	0.7	15.6	22.3	35.6	0.29
LPO	38.2	59.4	66.4	75.3	17.5	34.8	41.3	48.8	0.46
MCG	51.7	69.3	75.8	82.1	30.2	45.4	51.7	60.1	17.46
Objectness	38.2	50.2	56.4	65.4	17.4	22.6	25.0	29.3	0.91
Rahtu	34.3	46.9	53.3	62.3	21.9	32.1	38.1	45.8	0.67
RandomPrim	34.4	50.7	59.2	70.7	16.4	28.1	34.4	44.5	0.12
Rantalankila	0.6	3.1	6.5	14.9	0.2	1.2	2.6	7.4	3.57
SelectiveSearch	37.1	54.3	61.8	71.8	19.9	32.7	39.6	49.4	1.60
RPN	60.1	73.8	80.7	89.0	32.9	47.6	54.5	64.4	0.10
DeepBox	58.1	71.8	77.2	84.5	40.7	55.4	62.7	70.9	0.45
DeepMaskZoom	61.8	78.5	84.7	91.0	44.2	58.1	63.8	71.1	1.20
SharpMaskZoom	62.6	79.5	85.4	91.9	47.0	60.9	66.5	74.0	0.57
RefinedBox[1]	80.4	88.3	90.6	**92.7**	67.9	**76.4**	**79.2**	**82.4**	0.31
RefinedBox[2]	**80.5**	87.6	88.8	89.6	68.2	75.2	76.4	77.1	17.52
RefinedBox[3]	79.2	86.4	88.2	89.7	**68.6**	76.1	78.0	79.6	1.66
RefinedBox[4]	79.5	**88.6**	**90.8**	92.4	65.3	75.2	77.6	79.5	0.16

Metrics: To evaluate the proposals, we adopt the metrics of object detection recall (DR), mean average best overlap (MABO), and average recall (AR). Detection recall considers a ground truth object to be found when the IoU overlap of this ground truth object and a proposal is larger than a threshold. To calculate the average best overlap (ABO) for a specific class, we calculate the best IoU overlap between each ground truth annotation (belonging to this class) and proposals generated for

[1]https://github.com/rbgirshick/py-faster-rcnn



(a) DR *vs.* IoU

(b) DR *vs.* #WIN at IoU 0.5

(c) DR *vs.* #WIN at IoU 0.7

(d) MABO *vs.* #WIN

Figure 4: Evaluation results on the PASCAL VOC2007 test set. (a) shows object detection recall *vs.* IoU overlap threshold using 10 proposals per image. (b) and (c) display object detection recall *vs.* the number of proposals (#WIN) at IoU threshold 0.5 and 0.7, respectively. (d) shows MABO *vs.* the number of candidates using at most 100 proposals per image.

the corresponding image, and average over all ground truth objects in this class. MABO is defined as the mean ABO over all classes [8]. AR is introduced by Hosang *et al.* [37] to calculate the average recall when IoU thresholds are [0.5:0.05:0.95] for a fixed number of proposals.

4.2. Object Proposal Evaluation On the VOC2007 Dataset

Here, we first compare the proposed RefinedBox with other proposal refinement approaches, including DeepBox [18] and MTSE [19]. The comparison between different proposal refinement approaches is shown in Figure 3. We choose Edge Boxes [9] to produce the initial proposals which are inputted into these refinement algorithms, but we change the default parameter of non-maximum suppression from 0.75 to 0.9 in order to obtain more boxes. We find that our method achieves much higher object detection recall than other competitors at both IoU thresholds 0.5 and 0.7. The gap between our RefinedBox and other competitors is very large. Using only one proposal per image, RefinedBox achieves a detection recall of 43.2% and 34.2% at IoU 0.5 and IoU 0.7, respectively, while the recall of the original Edge Boxes are 29.1% and 15.2%, respectively. In addition, RefinedBox can share the convolutional layer with subsequent object detection, and the additional layers of RefinedBox are computationally lightweight, so RefinedBox is an efficient detection framework. In fact, the total time consumption of RefinedBox and subsequent object detection is similar to the Faster R-CNN [13] at about 0.13 second per image. DeepBox builds a separate network to re-rank boxes, while MTSE segments an image first and then uses superpixels to refine boxes;

Table 2: Evaluation results (%) in terms of AR, MABO, and mAP (object detection performance using 10 proposals per image) on the PASCAL VOC2007 *test* set. RefinedBox[1], RefinedBox[2], RefinedBox[3] and RefinedBox[4] mean RefinedBox with Edge Boxes, MCG, Selective Search and RPN, respectively.

#WIN	AR				MABO				mAP
	10	30	50	100	10	30	50	100	
BING	16.5	21.3	24.6	27.9	37.9	45.7	50.5	55.7	34.4
CSVM	17.0	22.7	25.5	29.1	40.3	49.2	53.1	57.9	35.7
EdgeBoxes	26.3	36.3	41.3	48.0	45.3	55.7	60.4	66.2	39.1
Endres	31.1	40.5	44.8	50.6	51.2	60.9	65.1	70.2	42.8
GOP	6.8	14.6	20.7	31.9	19.8	35.2	44.0	56.4	13.3
LPO	17.2	31.1	36.7	43.2	41.1	54.6	59.7	65.7	34.5
MCG	27.6	40.5	45.9	52.9	50.1	62.1	66.5	71.6	41.2
Objectness	16.8	22.0	24.6	28.8	39.4	46.5	49.9	54.8	34.9
Rahtu	18.5	26.5	30.8	36.8	37.2	46.5	51.3	57.3	32.4
RandomPrim	16.1	25.8	31.3	39.6	37.9	49.6	55.3	62.6	31.9
Rantalankila	0.2	1.2	2.7	7.0	4.1	8.5	12.9	22.3	2.4
SelectiveSearch	18.6	29.8	35.5	43.6	40.0	52.0	57.4	64.3	34.1
RPN	28.4	38.1	42.7	48.9	50.8	60.6	65.0	70.1	54.1
DeepBox	33.9	44.5	49.2	54.9	52.9	62.8	66.9	71.8	50.9
DeepMaskZoom	37.1	48.5	53.2	59.1	55.6	67.6	71.6	76.0	52.7
SharpMaskZoom	39.7	51.5	56.1	62.0	57.0	69.2	73.1	77.3	53.5
RefinedBox[1]	53.0	**58.7**	**60.6**	**62.4**	68.4	**74.1**	**75.8**	**77.4**	65.4
RefinedBox[2]	**53.7**	58.4	59.3	59.8	**68.9**	73.8	74.7	75.3	65.2
RefinedBox[3]	53.5	**58.7**	60.0	61.1	67.9	73.2	74.6	75.8	**65.5**
RefinedBox[4]	49.8	56.1	57.7	59.0	66.6	72.9	74.3	75.4	65.0

Table 3: Evaluation results (%) in terms of DR on the MS COCO validation set. RefinedBox[1], RefinedBox[2], RefinedBox[3] and RefinedBox[4] mean RefinedBox with Edge Boxes, MCG, Selective Search, and RPN respectively.

#WIN	DR (IoU=0.5)				DR (IoU=0.7)			
	10	30	50	100	10	30	50	100
BING	11.8	17.3	22.4	28.8	2.1	2.8	3.5	4.2
EdgeBoxes	17.7	26.2	30.7	37.7	11.4	18.1	21.8	27.5
GOP	11.3	22.7	30.0	41.1	7.3	13.8	18.1	25.1
LPO	15.1	26.6	32.2	42.1	7.0	14.4	18.4	24.7
MCG	24.5	36.7	42.5	50.6	14.7	23.5	28.1	34.6
Objectness	13.9	20.9	25.0	31.6	5.8	8.3	9.7	11.8
Rahtu	12.2	19.7	24.1	30.1	7.4	12.6	15.9	20.6
RandomPrim	12.9	22.4	28.2	37.2	6.2	11.7	15.3	21.4
SelectiveSearch	12.2	20.1	24.6	31.6	4.5	8.7	11.5	16.0
RPN	30.6	46.2	55.1	65.0	19.8	31.6	38.4	46.6
DeepBox	21.9	32.3	38.4	47.5	14.8	23.0	27.8	34.7
DeepMaskZoom	37.4	52.6	59.1	66.4	28.4	40.3	45.6	52.2
SharpMaskZoom	37.6	52.9	59.4	66.6	29.3	41.5	46.7	53.2
RefinedBox[1]	44.7	57.1	61.8	67.3	37.9	48.0	51.8	56.2
RefinedBox[2]	**45.4**	56.9	61.2	65.9	38.3	47.3	50.5	53.6
RefinedBox[3]	44.4	56.5	61.3	66.8	**38.5**	**48.9**	**53.1**	**57.6**
RefinedBox[4]	44.6	**57.3**	**62.4**	**68.1**	38.3	48.6	52.6	56.7

however, the image segmentation step is a time-consuming operation. Thus, RefinedBox is more suitable to be used in many applications.

Now, we compare with state-of-the-art object proposal generation methods. Extensive comparisons are shown in Figure 4. RefinedBox also uses Edge Boxes as input, and we apply the default parameters for the evaluation of Edge Boxes. Our method achieves the state-of-the-art performance across all cases. For object detection recall *vs.* the number of proposals at IoU 0.7, the performance improvements between RefinedBox and other competitors are also very large. The higher detection recall and fewer proposals will benefit the subsequent high-level applications a lot. RPN has recently become popular for object detection, but our proposed RefinedBox is much more accurate than it. The object detection recall of RefinedBox with only 10 proposals per image is similar to RPN using 100 proposals per image. The improvement from RPN to RefinedBox demonstrates the effectiveness of our method. With only a small number of proposals, RefinedBox can achieve much better performance than other competitors, including recent state-of-the-art deep learning based DeepMask [14] and SharpMask [15]. Using only 30 proposals, RefinedBox can achieve detection recall of 88.3 and 76.4 for IoU overlap 0.5 and 0.7, respectively. This will meet the requirements of many applications for a small amount of but high-quality object proposals.

To quantify these plots, we list the corresponding numbers in Table 1. RefinedBox achieves much better performance than various initial input methods. With Edge Boxes and an IoU threshold of 0.5, the detection recall of RefinedBox is 17.8%, 8.8%, 5.2%, and 0.8% higher than the second best method (SharpMaskZoom [15]) when using 10, 30, 50, and 100 proposals per image, respectively. At an IoU threshold of 0.7, the detection recall of RefinedBox with EdgeBoxes is 20.9%, 15.5%, 12.7%, and 8.4% higher than SharpMaskZoom when 10, 30, 50, and 100 proposals are used per image respectively. Since our goal is to significantly reduce the number of proposals, the evaluation results suggest that we have achieved it. We also notice that RPN [13] is much better than traditional non-deep approaches. This is the key reason why Faster R-CNN can achieve better detection performance than Fast R-CNN. Since RefinedBox aims at selecting and refining the good proposals from all proposals generated by previous method, the most influential factor is the upper bound of the input proposals, *i.e.* the largest detection recall of previous methods with enough proposals, not the performance with a limited number of proposals. On the VOC2007 dataset, Edge Boxes can achieve high detection recall with enough proposals, which is the reason why RefinedBox with Edge Boxes performs best. The runtime of RefinedBox for each image is about 0.06 second, which is very fast when compared with these traditional proposal generation methods. We report the AR and MABO of various competitors in Table 2. As expected, RefinedBox achieves the best performance again.

4.3. Object Detection On the VOC2007 dataset

Since object detection is an important application of object proposals, we test the quality of different proposal algorithms according to their performance in object detection. We feed the proposals produced by the aforementioned methods into a well-known region-based object detection framework, Fast R-CNN [2]. We optimize RefinedBox using the joint training algorithm described above. We follow the settings in [34]. The top 1000 proposals per image are used to retrain the Fast R-CNN network. All of these methods are trained on the VOC2007 *trainval* set and tested on the *test* set. Note that only the top 10 proposals per image are used to evaluate the ability of generating a small amount of proposals for different methods.

The results are summarized in Table 2. In terms of mAP, RefinedBox is 26.3%, 24.0%, 31.4% and 10.9% higher than the

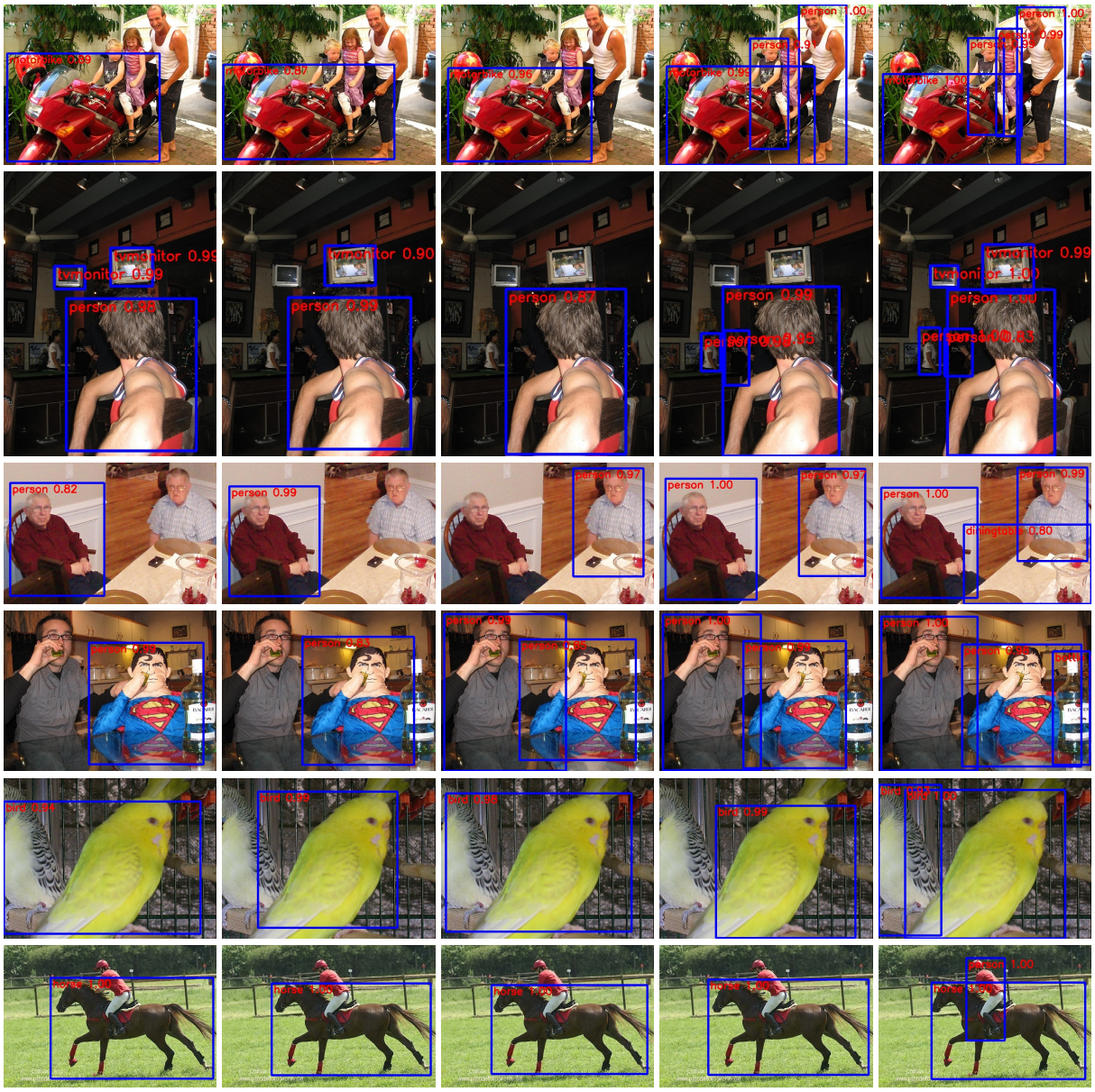
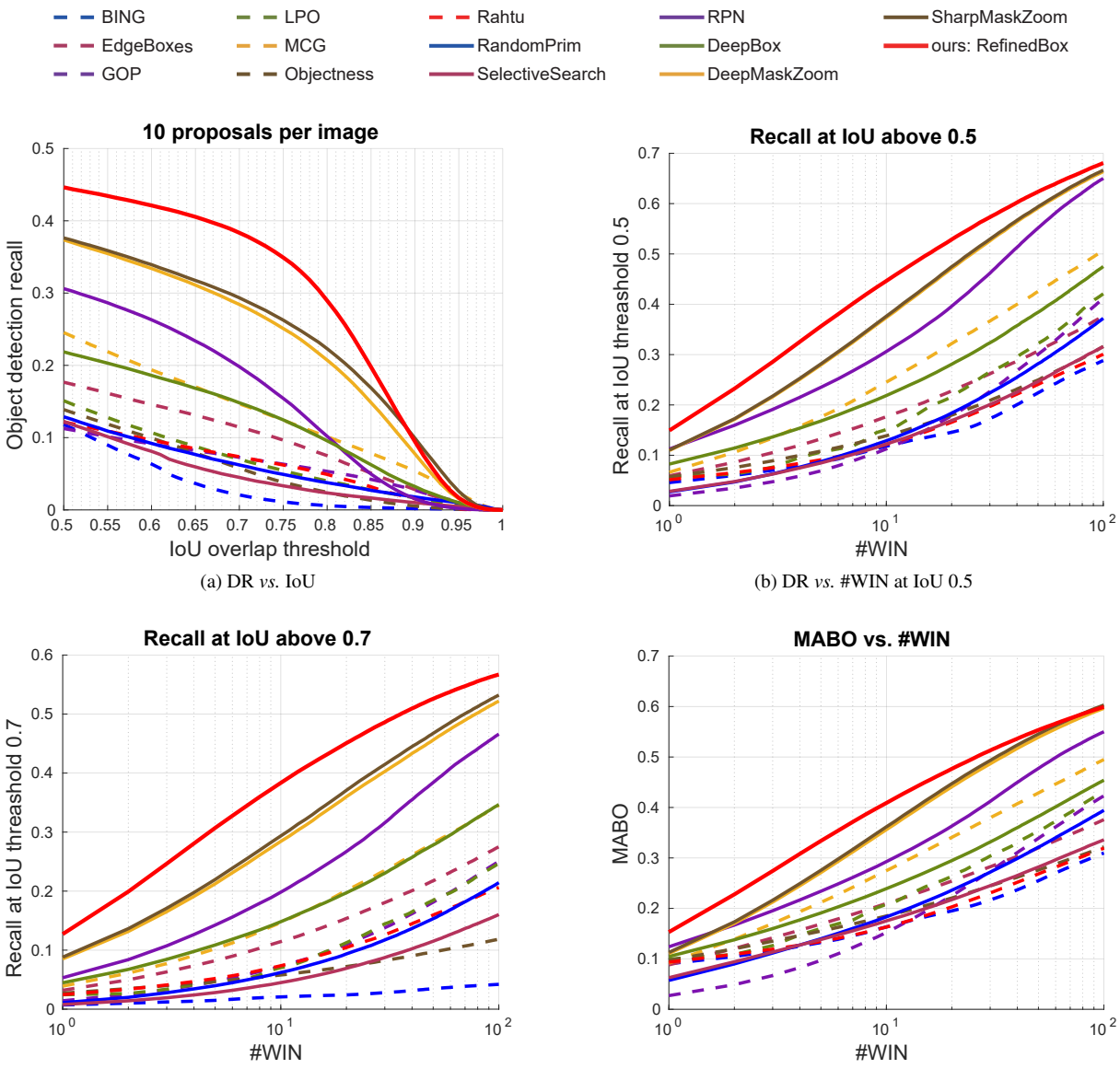


Figure 5: Qualitative comparison for object detection using only top 10 proposals. Here, RefinedBox uses Edge Boxes [9] as the input. All images are from VOC2007 *test* set.

original proposal methods, *i.e.* Edge Boxes, MCG, Selective Search and RPN, respectively. Compared with other proposal generation methods, RefinedBox can also achieve much higher detection performance. These evaluation results demonstrate that RefinedBox can generate a small amount of proposals with significantly high quality. It is intersting to observe that RPN [13] performs slightly better than DeepBox [18], DeepMask [14] and SharpMask [15] for object detection, while RPN performs worse for object proposal evaluation. Maybe this is because RPN is carefully designed for object detection in Faster R-CNN framework [13]. We provide qualitative comparison between RefinedBox and baselines for object detection in Figure 5. We can see that RefinedBox significantly improves the detection performance of baseline methods.

4.4. Object Proposal Evaluation On the COCO Dataset

In this part, we evaluate the proposed method and competitors on the COCO dataset. The visualization of DR and MABO is displayed in Figure 6. In each figure, there is a large gap between RefinedBox and other approaches, which demonstrates the effectiveness of RefinedBox in generating a small amount of proposals. The numeric comparison of DR is summarized in Table 3. The AR and MABO of various methods are shown in Table 4. RefinedBox performs significantly better than various competitors in terms of all metrics. SharpMask [15] achieves the second place and is slightly better than DeepMask [14]. Note that SharpMask and DeepMask uses mask annotations for training, while RefinedBox only uses box annotations for training. This further demonstrates the importance of a proper box



(a) DR *vs.* IoU

(b) DR *vs.* #WIN at IoU 0.5

(c) DR *vs.* #WIN at IoU 0.7

(d) MABO *vs.* #WIN

Figure 6: Evaluation results on the MS COCO validation set. RefinedBox uses RPN [13] as inputs. (a) shows object detection recall *vs.* IoU overlap threshold using 10 proposals per image. (b) and (c) display object detection recall *vs.* the number of proposals (#WIN) at IoU threshold 0.5 and 0.7, respectively. (d) shows MABO *vs.* the number of candidates using at most 100 proposals per image.

refinement to generate high-quality object proposals.

5. Conclusion

In this paper, we present a proposal refinement method using re-ranking and box regression. It is very efficient because the added layers are designed to be computationally lightweight. Extensive experiments demonstrate that RefinedBox can significantly reduce the number of proposals generated by previous algorithms. Since the refinement network can be easily optimized, we find we can perform joint training of it with consequent applications. The evaluation on object detection demonstrates the effectiveness of RefinedBox.

Limitations. Since the efficiency of the RefinedBox module is proportional to the number of initial proposals, RefinedBox may be less efficient for complex images that may have too many initial proposals. Since RefinedBox performs on the small feature maps caused by the the downsampling in the backbone network, the images with many small objects will affect its performance, as object detection methods [1, 13, 38].

Future work. A small amount of high-quality object proposals meet the requirements of many high-level applications, including multi-label image classification, [5], pedestrian detection [39], deep multiple instance learning [7], *etc*. With fewer but more accurate proposals, these tasks are expected to achieve better performance. In the future, we plan to apply our refinement method to other high-level applications, *e.g.* mining knowledge from huge amounts of unlabeled data.

Table 4: Evaluation results (%) in terms of AR and MABO on the MS COCO validation set. RefinedBox[1], RefinedBox[2], RefinedBox[3] and RefinedBox[4] mean RefinedBox with Edge Boxes, MCG, Selective Search and RPN, respectively.

#WIN	AR				MABO			
	10	30	50	100	10	30	50	100
BING	3.5	5.0	6.4	8.0	16.3	21.7	25.5	31.0
EdgeBoxes	9.9	15.1	17.9	22.3	21.0	28.2	32.0	37.6
GOP	6.6	12.5	16.4	22.6	15.2	27.4	33.9	42.3
LPO	6.9	13.4	16.9	22.6	20.8	30.4	35.3	43.1
MCG	13.6	21.3	25.3	30.9	27.5	37.9	42.9	49.5
Objectness	5.8	8.5	10.1	12.7	18.5	24.5	27.6	32.2
Rahtu	6.5	10.7	13.3	17.0	16.3	23.1	26.8	31.9
RandomPrim	6.1	11.1	14.4	19.7	18.3	27.2	32.1	39.4
SelectiveSearch	5.0	8.9	11.4	15.4	17.5	24.5	28.2	33.6
RPN	16.1	25.0	30.2	36.1	29.3	41.2	47.7	55.0
DeepBox	12.5	18.9	22.5	27.8	23.9	33.2	38.2	45.4
DeepMaskZoom	23.6	33.5	38.0	43.4	35.6	48.6	53.9	59.6
SharpMaskZoom	24.6	34.8	39.3	44.7	36.2	49.3	54.6	**60.3**
RefinedBox[1]	30.3	37.9	40.7	43.9	41.0	51.0	54.8	59.1
RefinedBox[2]	**31.3**	38.4	40.9	43.4	**42.1**	**51.8**	55.3	59.2
RefinedBox[3]	30.9	**38.8**	**41.8**	**45.2**	41.0	51.1	55.1	59.6
RefinedBox[4]	30.4	38.2	41.1	44.3	40.9	51.3	**55.4**	59.9

Acknoledgment. This research was supported by Major Project for New Generation of AI under Grant No. 2018AAA0100400, NSFC (61620106008), and Tianjin Natural Science Foundation (17JCJQJC43700).

References

[1] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conf. Comput. Vis. Pattern Recog., 2014, pp. 580–587.

[2] R. Girshick, Fast R-CNN, in: Int. Conf. Comput. Vis., 2015, pp. 1440–1448.

[3] K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, in: Int. Conf. Comput. Vis., 2017, pp. 2961–2969.

[4] A. Arnab, P. H. Torr, Pixelwise instance segmentation with a dynamically instantiated network, in: IEEE Conf. Comput. Vis. Pattern Recog., 2017, pp. 441–450.

[5] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, S. Yan, HCP: A flexible CNN framework for multi-label image classification, IEEE Trans. Pattern Anal. Mach. Intell. 38 (9) (2016) 1901–1907.

[6] Y. J. Lee, K. Grauman, Predicting important objects for egocentric video summarization, Int. J. Comput. Vis. 114 (1) (2015) 38–55.

[7] J. Wu, Y. Yu, C. Huang, K. Yu, Deep multiple instance learning for image classification and auto-annotation, in: IEEE Conf. Comput. Vis. Pattern Recog., 2015, pp. 3460–3469.

[8] J. R. Uijlings, K. E. Van De Sande, T. Gevers, A. W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.

[9] C. L. Zitnick, P. Dollár, Edge Boxes: Locating object proposals from edges, in: Eur. Conf. Comput. Vis., 2014, pp. 391–405.

[10] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, J. Malik, Multiscale combinatorial grouping, in: IEEE Conf. Comput. Vis. Pattern Recog., 2014, pp. 328–335.

[11] X. Qi, Z. Liu, J. Shi, H. Zhao, J. Jia, Augmented feedback in semantic segmentation under image level supervision, in: Eur. Conf. Comput. Vis., 2016, pp. 90–105.

[12] D. Li, J.-B. Huang, Y. Li, S. Wang, M.-H. Yang, Weakly supervised object localization with progressive domain adaptation, in: IEEE Conf. Comput. Vis. Pattern Recog., 2016, pp. 3512–3520.

[13] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Adv. Neural Inform. Process. Syst., 2015, pp. 91–99.

[14] P. O. Pinheiro, R. Collobert, P. Dollár, Learning to segment object candidates, in: Adv. Neural Inform. Process. Syst., 2015, pp. 1990–1998.

[15] P. O. Pinheiro, T.-Y. Lin, R. Collobert, P. Dollár, Learning to refine object segments, in: Eur. Conf. Comput. Vis., 2016, pp. 75–91.

[16] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.

[17] M.-M. Cheng, Y. Liu, W.-Y. Lin, Z. Zhang, P. L. Rosin, P. H. Torr, BING: Binarized normed gradients for objectness estimation at 300fps, Computational Visual Media 5 (1) (2019) 3–20.

[18] W. Kuo, B. Hariharan, J. Malik, DeepBox: Learning objectness with convolutional networks, in: Int. Conf. Comput. Vis., 2015, pp. 2479–2487.

[19] X. Chen, H. Ma, X. Wang, Z. Zhao, Improving object proposals with multi-thresholding straddling expansion, in: IEEE Conf. Comput. Vis. Pattern Recog., 2015, pp. 2587–2595.

[20] Tao, Dapeng and Guo, Yanan and Yu, Baosheng and Pang, Jianxin and Yu, Zhengtao, Deep multi-view feature learning for person re-identification, IEEE Trans. Circ. Syst. Video Technol. 28 (10) (2017) 2657–2666.

[21] Han, Junwei and Zhang, Dingwen and Cheng, Gong and Liu, Nian and Xu, Dong, Advanced deep-learning techniques for salient and category-specific object detection: A survey, IEEE Signal Process. Mag. 35 (1) (2018) 84–100.

[22] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Int. Conf. Learn. Represent., 2015.

[23] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The PASCAL visual object classes challenge 2007 (VOC2007) results, http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html (2007).

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: Eur. Conf. Comput. Vis., 2014, pp. 740–755.

[25] D. G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.

[26] S. Manen, M. Guillaumin, L. Van Gool, Prime object proposals with randomized prim's algorithm, in: Int. Conf. Comput. Vis., 2013, pp. 2536–2543.

[27] P. Rantalankila, J. Kannala, E. Rahtu, Generating object segmentation proposals using global and local search, in: IEEE Conf. Comput. Vis. Pattern Recog., 2014, pp. 2417–2424.

[28] E. Rahtu, J. Kannala, M. Blaschko, Learning a category independent object detection cascade, in: Int. Conf. Comput. Vis., 2011, pp. 1052–1059.

[29] I. Endres, D. Hoiem, Category-independent object proposals with diverse ranking, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2) (2014) 222–234.

[30] P. Krahenbuhl, V. Koltun, Learning to propose objects, in: IEEE Conf. Comput. Vis. Pattern Recog., 2015, pp. 1574–1582.

[31] B. Alexe, T. Deselaers, V. Ferrari, Measuring the objectness of image windows, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2189–2202.

[32] Z. Zhang, P. H. Torr, Object proposal generation using two-stage cascade SVMs, IEEE Trans. Pattern Anal. Mach. Intell. 38 (1) (2016) 102–115.

[33] C. Lu, S. Liu, J. Jia, C.-K. Tang, Contour box: Rejecting object proposals without explicit closed contours, in: IEEE Conf. Comput. Vis. Pattern Recog., 2015, pp. 2021–2029.

[34] Z. Zhang, Y. Liu, X. Chen, Y. Zhu, M.-M. Cheng, V. Saligrama, P. H. Torr, Sequential optimization for efficient high-quality object proposal generation, IEEE Trans. Pattern Anal. Mach. Intell. 40 (5) (2017) 1209–1223.

[35] S. He, R. W. Lau, Oriented object proposals, in: Int. Conf. Comput. Vis., 2015, pp. 280–288.

[36] P. Krähenbühl, V. Koltun, Geodesic object proposals, in: Eur. Conf. Comput. Vis., 2014, pp. 725–739.

[37] J. Hosang, R. Benenson, P. Dollár, B. Schiele, What makes for effective detection proposals?, IEEE Trans. Pattern Anal. Mach. Intell. 38 (4) (2015) 814–830.

[38] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection (2017) 2117–2125.

[39] S. Paisitkriangkrai, C. Shen, A. van den Hengel, Pedestrian detection with spatially pooled features and structured ensemble learning, IEEE Trans. Pattern Anal. Mach. Intell. 38 (6) (2016) 1243–1257.