Learning to Rank Proposals for Object Detection

Zhiyu Tan[1] Xuecheng Nie[2] Qi Qian[1] Nan Li[1] Hao Li[1]

[1]Alibaba Group, Beijing, China

[2]Department of Electrical and Computer Engineering, National University of Singapore, Singapore

{zhiyu.tzy, qi.qian, nanli.ln, lihao.lh}@alibaba-inc.com niexuecheng@u.nus.edu

Abstract

Non-Maximum Suppression (NMS) is an essential step of modern object detection models for removing duplicated candidates. The efficacy of NMS heavily affects the final detection results. Prior works exploit suppression criterions relying on either the objectiveness derived from classification or the localization produced by regression, both of which are heuristically designed and fail to explicitly link with the suppression rank. To address this issue, in this paper, we propose a novel Learning-to-Rank (LTR) model to produce the suppression rank via a learning procedure, thus facilitating the candidate generation and lifting the detection performance. In particular, we define a ranking score based on IoU to indicate the ranks of candidates during the NMS step, where candidates with high ranking score will be reserved and the ones with low ranking score will be eliminated. We design a lightweight network to predict the ranking score. We introduce a ranking loss to supervise the generation of these ranking scores, which encourages candidates with IoU to the ground-truth to rank higher. To facilitate the training procedure, we design a novel sampling strategy via dividing candidates into different levels and select hard pairs to adopt in the training. During the inference phase, this module can be exploited as a plugin to current object detector. The training and inference of the overall framework is end-to-end. Comprehensive experiments on benchmarks PASCAL VOC and MS COCO demonstrate the generality and effectiveness of our model for facilitating existing object detectors to state-of-the-art accuracy.

1. Introduction

Object detection is a fundamental yet challenging task in computer vision, It is extensively applied in video/image indexing [29] [24], face recognition [31], autonomous driving cars [7] [19] and human pose estimation [32].

Existing object detection models heavily rely on the Non-Maximum Suppression (NMS) algorithm to remove duplicated bounding boxes via a suppression criterion, which is defined either from the objectness derived from classification or localization produced by regression. However, these existing suppression criterions fail to explicitly link with the candidate rank in the elimination procedure, as shown in Figure 1. Inaccurate ranks of candidates will cause error eliminations and degrades the performance of object detectors. The suppression criterion still needs to be improved to facilitate the performance of object detectors.

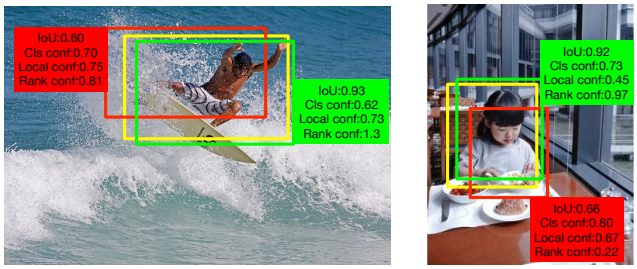

Figure 1. Motivations for the proposed Learning-to-Rank model for oject detection. Yellow boxes represent the groundtruth. Read and green boxes represent the predicted candidates. Existing works fail to reserve the more accurate green candidates due to the heuristically designed suppression criterions based on classification or localization scores. The proposed LTR model addresses this problem via producing the suppression rank with a learning procedure. See text for details.

In this paper, we propose to enhance the suppression criterion in both the training and inference phases of object detectors. We observe existing criterions are either designed in a heuristic manner or produced in an implicit way. An explicit ranking score definition and generation can successfully remove the gap between the suppression criterion and candidate preservation. Motivated by this, we propose a Learning-To-Rank (LTR) model to predict the ranking scores of candidates in a learning procedure, thus overcoming the limitations of existing methods and improving the object detection.

In particular, we define a ranking score based on IoU to indicate the ranks of candidates during the NMS step, where candidates with high ranking score will be reserved and the ones with low ranking score will be eliminated. We

design a lightweight network to predict the ranking score. We introduce a ranking loss to supervise the generation of these ranking scores, which encourages candidates with IoU to the ground-truth to rank higher. To facilitate the training procudure, we design a novel sampling strategy via dividing candidates into different levels and select hard pairs to adopt in the training. During the inference phase, this module can be exploited as a plugin to current object detector.

We implement this module with a small convolutional neural network. The training and inference of the overall framework is end-to-end. Comprehensive experiments on PASCAL VOC [8] and MSCOCO [22] show that our proposed LTR model achieves outperforming accuracy with multiple detection framework, including Faster R-CNN [11], Mask RCNN [12] and Cascade rcnn [2], without any bells and whistles. Our contributions can be summarized into three folds: (1) We propose a novel Learning-To-Rank model to improve the NMS algorithm in both training and inference phases of object detection. (2) We propose a novel pair sampling strategy to improve the learning speed. (3) Our LTR model generally improves the performance of multiple object detector and sets new state-of-the-art accuracy on multiple benchmarks.

2. Related work

Object detection has been extensively studied in the literature. Recently, deep learning techniques significantly boost the performance of object detection algorithms over traditional methods based on hand-crafted ones (*e.g.,* Haar [35], SIFT [9], HOG [6], etc), due to its strong capabilities to extract power features with Convolution Neural Networks (CNNs). Existing CNN based methods can be divided into two categories: one-stage based detectors and two-stage based ones.

One-stage based detectors, such as SSD [23], Retinanet [21] and YOLO [25], are mainly focused on computation efficiency, which enables fast object detections. In term of accuracy, two-stage based detectors still dominate the community and usually outperform the single-stage based ones. In this paper, we aim to improve the two-stage based detectors, thus further pushing forward the frontier of object detection. Current two-stage based object detectors involves two steps: (1) object proposal generation and (2) object classification and bounding box refinement. In this paradigm, Non-Maximum Suppression (NMS) plays a key role to produce high quality candidates, which significantly affecting the final detection results. However, most of existing works ignore this important post-processing step. Here, we propose to improve the NMS, thus lifting the performance of two-stage based detectors. In the following, we mainly review existing NMS algorithms.

In order to remove massive duplicated bounding boxes, NMS is applied as the post-process procedure in mainstream detection pipeline [9] [11] [27] [4]. NMS selects the bounding box with the maximum classification confidence and eliminates its nearby boxes using a predefined IoU threshold iteratively. In computer vision, it has been almost 50 years to take the NMS algorithm as the post-processing process for most of object detection framework. NMS is still crucial to CNN-based detectors, which improves performance by removing duplicated results. Although detectors can generate many candidate bounding boxes with accurate location before NMS, these bounding boxes will probably be removed because of lower predicted confidence during NMS. Recently, many effective techniques are proposed to improve NMS. Soft-NMS [1] is a parameter-free algorithm, where duplicate bounding boxes removal is replaced by decaying the bounding box scores with a continuous function. A set of learning-based algorithms have been proposed as alternatives to the parameter-free NMS and Soft-NMS. Softer NMS [14] averages the selected boxes in a softer way. Learning NMS [15] uses a complex neural network to perform NMS using only boxes and their scores. Fitness NMS [34] introduces the localization information of bounding box into ranking confidence. Relation network [16] uses a sub-network to learn NMS by mining the visual information of object-object interactions. However, the confidence training of prior works are all based on classification task or regression task, which are not suitable for NMS algorithm.

Different from existing NMS algorithms, we propose a novel learning-to-rank model to produce the suppression score via a learning procedure and explicitly build link to the suppression rank, thus facilitating the candidate generation and lifting the detection performance, which is elaborated bellow.

3. Proposed Approach

In this section, we will illustrate the proposed Learning-to-Rank network model for object detection. An overview of the LTR model is given in Figure 2. The core of LTR is the Rank-NMS subnetwork, which is pluggable to the conventional two-stage object detectors. In particular, the Rank-NMS subnetwork takes the ROI-Aligned features of positive bbox candidates from the object detector as input and predicts their ranking scores based on Intersection-over-Union (IoU) values to the ground-truth. Then, LTR fuses the ranking scores with the classification scores to produce the final ranking confidences, which are used as suppression criterion for the NMS algorithm. In this way, LTR overcomes limitations of existing heuristically designed criterions for NMS and encourages to reserve bbox candidates with higher IoU with the ground-truth, thus improving the object detection accuracy. To train the Rank-NMS subnetwork, we design a ranking loss as supervision and propose a sample-pair selection strategy to speed up convergence,

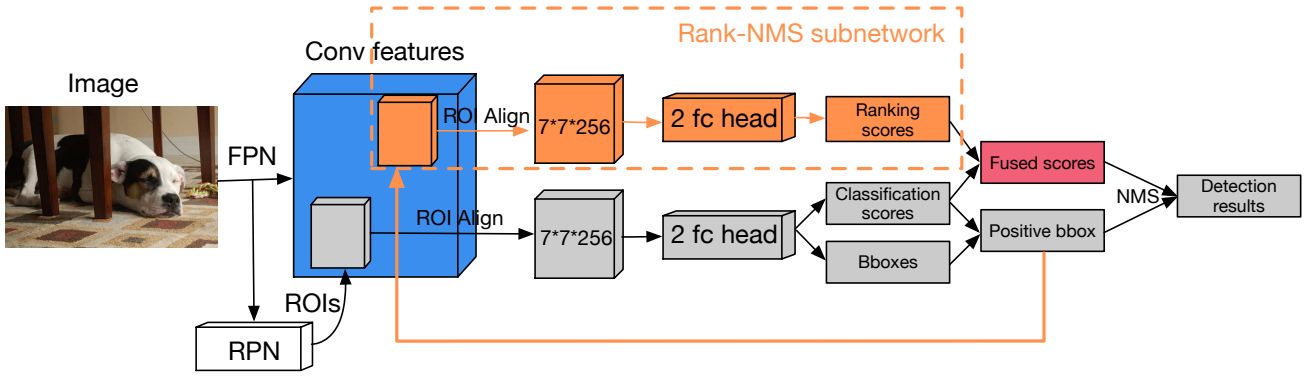

Figure 2. Overview of the proposed Learnin-to-Rank model for object detections.

details of which are illustrated in the following.

3.1. Ranking Loss

In general, the higher IoU values the bboxes to the groundtruth are, the better the localization is. Therefore, the Rank-NMS subnetwork, denoted as $\mathrm{R}(\cdot)$, aims to predict higher ranking scores for bboxes with higher IoU values and vice versa. To achieve this goal, we define the IoU based ranking as following

$$\mathrm{R}(f(b_i)) + \alpha < \mathrm{R}(f(b_j)), \text{s.t. } \rho_i < \rho_j, \tag{1}$$

where b_i and b_j denote bbox candidates, and ρ_i and ρ_j their IoU values with groudtruth, respectively. $f(\cdot)$ denotes the ROI-Aligned operation to extract features for a bbox. α is a constant to control the ranking margin. Then, we define the ranking loss by

$$L = \frac{1}{N} \sum_{(b_i,b_j),\ \rho_i \le \rho_j} \max\left(0, \mathrm{R}(f(b_i)) - \mathrm{R}(f(b_j)) + \alpha\right), \tag{2}$$

where N is the total number of bbox pairs that satisfy the ranking condition defined in Eqn. (1). We train the Rank-NMS subnetwork by minimizing the loss defined in Eqn. (2). In practice, N is always very large and the possible bbox pairs contains many easily-ranked ones. To accelerate the training, we propose an effective sample-pair selection strategy, explained in the next subsection.

3.2. Pair Selection

The selection strategy of training pairs is crucial to learn the proposed Rank-NMS network model. To reduce the redundant and eliminate the uninformative bbox paris from Eqn. (1), we propose a sampling-after-splitting strategy to effectively select valuable training pairs. In particular, we first divide all bboxes into subsets according to their q-values, which are calculated by the quantization function:

$$q(\rho_i) = \left\lceil \frac{\max(0, \rho_i - 0.5)}{0.05} \right\rceil. \tag{3}$$

Algorithm 1 Pair sampling algorithm

Input: The set of ranking scores for n bbox candidates $\mathcal{R}=\{r_1, ..., r_n\}$ and the set of k quantized subsets $\mathcal{S}=\{S_0, ..., S_k\}$;

Output: The set of selected pairs $\mathcal{P}$

$\mathcal{P} \leftarrow \emptyset$;
for $i = 1$ to k **do**
  $\mathrm{Pos}_i \leftarrow S_i$; // take all positive samples
  $U_i \leftarrow S_0 \cup ... \cup S_{i-1}$; // take all negative samples
  Generate a ranked list L_i upon U_i in descending order of the predicted ranking score;
  $\mathrm{Neg}_i \leftarrow$ get top h elements of L_i;
  Generate the pair set $\mathcal{C}_i$ by the Cartesian product of Pos_i and Neg_i;
end for
$\mathcal{P} \leftarrow C_i \cup \cdots \cup \mathcal{C}_k$;
return $\mathcal{P}$;

Eqn. (3) quantizes bbox candidates based on their IoU values with groundtruth and enforces larger IoU values to produce larger quantization results. Based on Eqn. (3), we will generate 11 subsets with their indices ranging from 0 to 10. Then, we sample the positive and negative samples individually for each of quantized subsets following the same rules—bounding boxes with higher IoU values to the ground-truth are defined as positive samples while bounding boxes with lower IoU values as negative ones. For the ith quantization subset, its positive samples are the ones in itself. While its negative samples are the top-h elements in the list that ranks the bounding boxes from union of the quantized subsets with the quantization value smaller than i in the descending order based on the ranking score. Then, the training pair set for this quantiation subset is derived by Cartersian product between its positive and negative samples. We repeat the above procedure to generate the training pair set for all the quantization subsets. Algorithm 1 illustrates the pair selection strategy.

3.3. Score Fusion

Prior works always exploits classification scores of bbox candidates as the suppression criterion of NMS algorithm, which considers the objectiveness and well filters out negative candidates. Differently, the ranking score accounts for the overlapness between bbox candidates and groudtruth and is good at reserving accurate positive candidate. To derive more reliable suppression criterion, we propose to fuse the classification and ranking scores via their weighted sum:

$$s = \beta s_r + (1 - \beta) s_c, \tag{4}$$

where s_r and s_c denote the ranking and classification scores, respectively. β is the balance factor, set as 0.15. s is the final score used as the suppression criterion of the NMS algorithm in both training and inference phases.

4. Experiments

4.1. Experiment setup

Datasets We conduct experiments on two widely used benchmarks for the object detection task: MSCOCO [22] dataset and PASCAL VOC [8] dataset. In particular, MSCOCO is a large scale dataset with 80 categories of object annotations in total. In this paper, we use the 2017 split, containing 118,000 samples, to train our models. We use the standard validation split (5,000 samples) and test-dev split (20,000 samples) for evaluating our proposed method. We exploit the official Average Precision (AP) as metric. PASCAL VOC is another dataset for extensively evaluating the object detection algorithms. It provides annotations for 20 object classes and provides two different splits: VOC 2017 and VOC 2012. Here, we follow the conventions to use the combination of training and validation sets of these two splits, for model training, including 16,000 images in total. We evaluate our models on the VOC 2007 test set also with AP as the metric.

Data augmentation Follow conventions, we only adopt horizontal flipping in data augmentation to our models for both MSCOCO and PASCAL VOC datasets.

Training and inference We exploit ResNet [13] as the network backbone. In the training phase, we exploit the ImageNet pre-trained models to initialize our network for both MSCOCO and PASCAL VOC datasets. The new adding layers are randomly initialized with normal distributions, where the mean is set to 0 and standard deviation 0.01 or 0.001. We set the dimension of features as 1024 for all the classification, regression and ranking tasks. We extract 512 proposals per image from the region proposal network. We make all the regressors in our model class-agnostic for simplicity. We use a batch size of 16 for training all the models. We implement the proposed method with MMDetection [3].

Table 1. Ablation analysis on the ranking margin α in the ranking loss, with β fixed as 0.15 on PASCAL VOC 2007VAL dataset.

α	AP	AP^{50}	AP^{60}	AP^{70}	AP^{80}	AP^{90}
0.0	54.45	79.91	74.76	61.66	39.08	10.73
0.25	57.92	79.87	75.65	65.39	46.12	16.29
0.5	58.75	79.98	75.50	65.43	48.00	18.34
0.75	58.32	78.39	74.63	65.25	48.30	18.47
1.0	57.22	76.48	73.02	63.89	47.75	18.24

We use the SGD as the optimizer for model learning. We train all models for 12 epochs in total. We set the initial learning rate as 0.02 and 0.01 for MSCOCO and PASCAL VOC, respectively. In addition, we drop the learning rate by a factor of 10 at 8th and 11 epochs, while for PASCAL VOC, we only drop the learning rate at 9th epoch with the same factor. We use the Cross-Entropy loss for object classification and Smooth L1 loss [10] for bounding box regression, which are summed with the ranking loss for supervising the model training. We also adopt linear warming up strategy to begin the training of our model. For MSCOCO (PASCAL VOC), We resize the image to make its short edge as 800 (600) pixels while keeping the long edge no longer than 1,333 (1,000) pixels, as input to the network for both training and inference. We perform single-scale testing to generate all the results.

4.2. Results on PASCAL VOC

4.2.1 Ablation study

We first conduct ablation studies on the validation set PASCAL VOC 2007VAL to analyze the effects of some important hyper-parameters to the proposed model, including the ranking margin α in Eqn. 1 and the balance weight β in Eqn. 4. Results are shown in Table 1 and 2, respectively.

From Table 1, we can see when setting α as 0, which indicates learning the rank loss without margin, our model achieves 54.45 AP. While exploiting soft margin ($\alpha > 0$) brings obvious performance improvements, *e.g.,* $\alpha = 0.5$ achieves 7% accuracy gain over $\alpha = 0$, demonstrating that soft margin can enlarge the rank gap for proposal pairs and leading to more discriminative ranking scores. We can also see that increasing α from 0 to 0.5, the accuracy is consistently raised from 54.45% AP to 58.75% AP, especially for high IoU metrics, *e.g.,* AP^{90}, due to the improvement to the error tolerance. However, further increasing α to 1.0 degrades the performance, caused by introducing much noise with too large margin during the ranking process. Therefore, we set α as 0.5 in our experiments.

Table 2 shows the effects of the balancing weight β on the detection results, while varying α from 0.25 to 0.75. We can see that with small ranking margin $\alpha = 0.25$, properly increasing β can bring performance improvement, implying that correct ranking score is complementary to the

Table 2. Ablation analysis on the balancing weight β for score fusion with different α on PASCAL VOC 2007VAL dataset.

β	α	AP	AP^{50}	AP^{60}	AP^{70}	AP^{80}	AP^{90}
0.05	0.25	57.28	79.94	75.38	64.64	44.55	15.49
0.05	0.5	58.26	80.29	75.64	65.51	46.56	16.93
0.05	0.75	58.44	79.56	75.27	65.56	47.60	17.63
0.15	0.5	**58.75**	79.98	75.50	65.43	48.00	18.34
0.25	0.25	58.09	79.65	75.52	65.29	46.56	16.57
0.25	0.5	58.18	78.74	74.74	65.20	47.55	17.66
0.25	0.75	56.89	75.78	72.31	63.28	47.75	18.62
0.35	0.25	57.99	79.26	75.13	65.13	47.14	16.69
0.35	0.5	57.19	76.93	72.97	63.96	47.29	17.96
0.35	0.75	55.29	73.22	70. 00	61.24	46.57	18.81
0.45	0.25	57.56	78.44	74 3.3	64 5.8	46.82	16.93
0.45	0.5	55.96	74.99	71.29	62 37.	46.28	17.81
0.45	0.75	54.08	71.31	68. 32	59 9.1	45.70	18.49

Table 3. Ablation analysis on the positive samples ratio on PASCAL VOC 2007VAL dataset.

pos ratio	AP	AP^{50}	AP^{60}	AP^{70}	AP^{80}	AP^{90}
0.25	58.75	79.98	75.50	65.43	48.00	18.34
0.5	58.26	79.50	75.40	65.19	47.09	17.36
0.75	58.60	79.70	75.24	65.11	47.64	18.48

classification confidence and helps to select more suitable object proposals. However, when using large ranking margin, larger β causes accuracy drop, due to the introducing of ranking noise that brings negative effects for generating proposals. In addition, we find that $\beta = 0.15$ with $\alpha = 0.5$ gives the best detection performance, which is utilized default parameter setting in our experiments.

Next, we conduct experiments to analyze the effects of the ratio of positive samples in the learning phase with the proposed ranking loss. Results are shown in Table 3. We experiment with three different ratios ranging from 0.25 to 0.75. We can see decreasing the positive sample ratio always improves the performance, indicating less positive samples can facilitate the model to differentiate correct proposals with erroneous ones, which thereby enhances the learning of the proposed ranking loss.

Then, we compare the proposed hard-pair sampling strategy with the full-pair one and show the results in Table 4. We can see that the proposed hard-pair sampling strategy achieves superior performance over the full-pair one (58.75% AP vs 56.54% AP), demonstrating the effectiveness of mining hard pairs of proposals to facilitate the learning of the ranking loss. The superiority of the proposed hard-pair sample strategy can be further observed when using high IoU metrics, *e.g.,* at AP^{90}, the hard-pair sampling strategy lifts the performance from 13.31% AP to 18.34% AP, which further verifying its efficacy to improve proposal selection.

Table 4. Comparison between the proposed hard-pair sampling strategy and the full-pair one on PASCAL VOC 2007VAL dataset.

Sampling Strategy	AP	AP^{50}	AP^{60}	AP^{70}	AP^{80}	AP^{90}
Hard-Pair	58.75	79.98	75.50	65.43	48.00	18.34
Full-pair	56.54	79.49	75.09	63.75	43.84	13.31

4.2.2 Comparison with the state-of-the-arts

Table 5 shows the comparisons between the proposed approach with state-of-the-arts on the full testing dataset of PASCAL VOC 2017. We evaluate the efficacy of the proposed Rank-NMS on two state-of-the-art object detectors: Faster R-CNN and Cascade R-CNN. We adopt the ResNet-FPN as the backbone and vary its depth in 50 and 101.

From Table 5, we can see the proposed Rank-NMS consistently improves the detection accuracy for both Faster and Cascade R-CNNs. The performance improvement on Faster R-CNN is obvious, 4.59% AP and 3.35% AP with the backbone of 50 and 101 layers ResNet-FPN, respectively. These results validate the effectiveness of the proposed Rank-NMS for generating high quality proposals. We can also find that even the baseline Cascade R-CNN already achieves very high detection accuracy, the proposed Rank-NMS still raises their performance, which can be more obviously observed at AP^{90} (from 21.53% AP to 24.68% AP and from 25.75% AP to 28.90% AP with ResNet-50/101-FPN as backbone, respectively.). These results further demonstrate the effectiveness of the proposed Rank-NMS to improve the proposal generation for current state-of-the-art object detectors.

4.3. Results on MSCOCO

In this section, we conduct experiments on MSCOCO dataset to further evaluate the efficacy of the proposed Rank-NMS. Details are explained bellow.

4.3.1 Compare with other NMS method

We first compare the proposed Rank-NMS with tradition NMS algorithms: Soft-NMS [1] and IoU-NMS (the standalone version) [18], on MSCOCO validation set. In particular, the Soft-NMS is based on the confidence score from the classification while the IoU-NMS the overlap score from the regression. We conduct experiments with different objectors, including Faster R-CNN, Cascade R-CNN and Mask R-CNN, to comprehensively analyzing the effectiveness of different NMS algorithms. Here, we utilize ResNet-50 as the backbone for all models. In addition, we also experiment with the combination of the proposed Rank-NMS and Soft-NMS to verify the compatibility of the proposed method. Results are shown in Table 6.

We can see, with a fixed object detector, the proposed Rank-NMS consistently outperforms IoU-NMS and Soft-

Table 5. Comparison with state-of-the-arts on the full testing dataset of PASCAL VOC 2007 .

Backbone	Method	Rank-NMS	AP	AP^{50}	AP^{60}	AP^{70}	AP^{80}	AP^{90}
ResNet-50-FPN	Faster R-CNN	✗	54.16	79.79	75.02	61.74	39.02	8.84
	Faster R-CNN	✓	58.75	79.98	75.50	65.43	48.00	18.34
	Cascade R-CNN	✗	60.09	79.98	74.66	65.84	50.52	21.53
	Cascade R-CNN	✓	61.32	79.71	75.74	67.13	52.31	24.68
ResNet-101-FPN	Faster R-CNN	✗	58.22	82.1	77.49	66.56	45.70	12.50
	Faster R-CNN	✓	61.57	80.58	76.90	67.70	52.40	22.67
	Cascade R-CNN	✗	63.37	81.69	77.30	69.26	55.01	25.75
	Cascade R-CNN	✓	63.90	81.14	77.17	69.81	56.08	28.90

Table 6. Comparisons between the proposed Rank-NMS with tradition NMS algorithms on MSCOCO validation set.

Method	+IoU-NMS [18]	+Soft-NMS	+Rank-NMS	AP	AP^{50}	AP^{75}	AP^S	AP^M	AP^L
Faster R-CNN				36.4	58.4	39.1	21.6	40.1	46.6
	✓			37.3	56 0	-	-	-	-
		✓		36.9	58.4	40.1	21.9	40.7	47.1
			✓	38.6	58.2	41.7	22.4	42.4	50.9
		✓	✓	38.9	58.1	42.4	22.5	42.8	51.2
Cascade R-CNN				40.3	58.6	43.9	22.9	43.8	53.2
	✓			40.9	58.2	-	-	-	-
		✓		41.0	58.8	45.2	23.2	44.6	54.0
			✓	41.0	59.1	44.5	23.0	44.3	54.8
		✓	✓	41.4	58.8	45.5	23.2	44.9	55.3
Mask R-CNN				37.3	59.1	40.3	22.0	40.9	48.2
	✓			38.1	56.4	-	-	-	-
		✓		37.8	59.1	41.3	22.2	41 6.	48.7
			✓	39.3	58.8	42.3	22.8	42.7	52.2
		✓	✓	39.6	58.7	43.1	23.0	43.1	52.5

NMS. This demonstrates the advantage of the proposed Rank-NMS over existing classification or regression based NMSs for selecting high quality proposals with the learned ranking score. We can also see that the performance improvement on large objects is more obvious, *e.g.,* with Mask R-CNN detector, Rank-NMS improves the accuracy from 48.7% AP^L to 52.2% AP^L. In addition, we can find the proposed Rank-NMS improves all the object detectors, demonstrating its generality. Moreover, we can observe, combining the Rank-NMS with Soft-NMS can further achieve performance gain. This result shows that the proposed Rank-NMS is compatible with existing NMS algorithms and provides valuable complementary proposals to improve object detection. For the comparison with Softer-NMS [14], its map result achieves 39.2 AP on COCO val2017 by lengthening the learning period. Our Rank-NMS achieves 38.9 AP with only half of the learning period as Softer-NMS and has the potential to further improve performance by lengthening the learning period.

For further analyzing the advantages of the proposed Rank-NMS, we plot the recall curve for different NMS algorithms in Figure 3, with the matching IoU ranging from 0.5 to 1. We can find that the proposed Rank-NMS consistently achieves better recall over the traditional NMS algorithms when varying the matching IoU, indicating that it produces the improved ranking list for proposals. In addition, we can see that combining Rank-NMS with Soft-NMS 34.5% recall at matching IoU 0.9, where the recall upperbound is 41.4%. This results further validate the effectiveness of the proposed Rank-NMS for preserving higher IoU proposals.

Similar to PASCAL VOC dataset, we also conduct experiments on MSCOCO validation set to analyze the effects of the proposed Rank-NMS on different object detector. In addition, we also report the running speed to analyze the efficiency of Rank-NMS. Results are shown in Table 9. We can see that the proposed Rank-NMS can consistently improve the object detection models, including Faster R-CNN, Cascade R-CNN and Mask R-CNN. We can also see that the proposed Rank-NMS can improve the object detectors with large network backbone, *e.g.,* Cascade R-CNN with ResNet-101-FPN. These results further validate the ef-

Table 7. Studies on the effects of classification and ranking scores to the NMS algorithm for object detection task on MSCOCO val5k dataset.

NMS Score	AP	AP^{50}	AP^{75}	AP^S	AP^L	AP^M
NMS-Cls	36.4	58.4	39.1	21.6	40.1	46.6
NMS-LTR	35.4	51.6	38.7	19.0	39.2	48.4
NMS-Cls+LTR	38.6	58.2	41.7	22.4	42.4	50.9

fectiveness and generality of the proposed Rank-NMS for improving the proposal selection. In addition, we can see that using Rank-NMS only brings slight times cost, which demonstrates the efficiency of the proposed Rank-NMS for object detection.

4.3.2 Performance of individual LTR

We conduct experiments on COCO dataset for Faster FPN model with ResNet50 to study the effects of the classification and ranking scores to the NMS algorithm for object detection task. Results are shown in Table 7. We use NMS-LTR to denote the NMS algorithm with only the ranking score for suppressing bounding boxes while NMS-Cls the one with only the classification score. We use NMS-Cls+LTR to denote performing NMS algorithm with the fusion of classification and ranking scores. We use Faster R-CNN as the object detector and the other settings are the same for the experiments. We can see the baseline NMS-Cls achieves 36.4 AP. While NMS-LTR slightly decreases the performance to 35.4 AP, since the ranking score is better at ranking bounding box candidates for the true positives but it is weaker to distinguish the true negatives than the classification score. After fusing the ranking score with classification score, we can see NMS-Cls+LTR achieves 38.6 AP, outperforming both the NMS algorithm with only a single kind of score. This result verifies that the proposed ranking score is complementary to the classification score, which can help produce more accurate ranking of bounding box candidates after filtering the false alarms with the classification score. We will add the above experiments and illustrations in revision.

4.3.3 Comparison with the state-of-the-arts

Next, we compare the proposed Rank-NMS with state-of-the-art object detection models on MSCOCO test-dev. We evaluate the proposed method with three object detectors: Faster R-CNN, Mask R-CNN and Cascade R-CNN[1]. We use ResNet-101-FPN as the backbone. We report the performance of all methods with single-model inference for fair comparison. Results are shown in Table 8.

We can see that with Cascade R-CNN as the object detector, our Rank-NMS sets new state-of-the-art 43.7% AP on MSCOCO test-dev, demonstrating the superior performance of the proposed methods. We can also find that when combining with Soft-NMS, the performance can be further improved to 43.7% AP, in addition improving all object detectors with Rank-NMS only. These results further verify the compatibility of the proposed Rank-NMS method.

[1]We set positive samples ratio of each stage of Cascade R-CNN as 0.5.

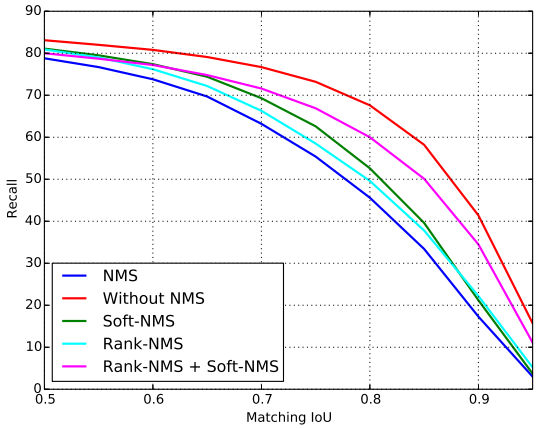

Figure 3. Comparison among the recall curves of different NMS algorithms with the matching IoU ranging from 0.5 to 1.

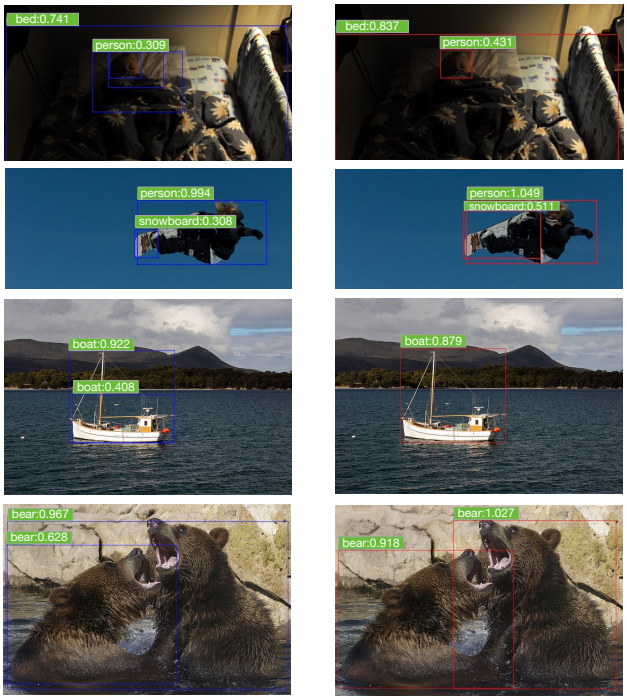

Figure 4. Qualitative comparison between the proposed Rank-NMS and the traditional NMS on MSCOCO dataset. .

4.3.4 Qualitative results

Qualitative results for comparison between the proposed Rank-NMS with traditional NMS algorithms are provided

Table 8. Comparison with the state-of-the-art single-model detectors on MSCOCO test-dev. * denotes using bells and whistles at inference.

Method	Backbone	AP	AP^{50}	AP^{75}	AP^S	AP^M	AP^L
YOLOv2 [25]	DarkNet-19	21.6	44.0	19.2	5.0	22.4	35.5
YOLOv3 [26]	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9
SSD513 [23]	ResNet-101	31.2	50.4	33.3	10.2	34.5	49.8
RetinaNet [21]	ResNet-101	39.1	59.1	42.3	21.8	42. 7	50.2
Faster R-CNN	ResNet-101-FPN [20]	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [30]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN	Inception-ResNet-v2-TDM [28]	36.8	57.7	39.2	16.2	39.8	52.1
Mask R-CNN [12]	ResNet-101-FPN	38.2	60.3	41.7	20.1	41.1	50.2
Cascade R-CNN [2]	ResNet-101-FPN	42.8	62.1	46.3	23.7	45.5	55.2
Fitness NMS [34]	DeNet-101 [33]	41.8	60.9	44.9	21.5	45.0	57.5
Deformable R-FCN [5] *	Aligned-Inception-ResNet	37.5	58.0	40.8	19.4	40.1	52.5
Deformable R-FCN* + Soft-NMS [1]	Aligned-Inception-ResNet	40.9	62.8	-	23.3	43.6	53.3
IoU-Net [18]	ResNet-101-FPN	40.6	59.0	-	-	-	-
Faster R-CNN +Rank-NMS	ResNet-101-FPN	41.0	60.8	44.5	23.2	44.5	52.5
Faster R-CNN +Rank-NMS +Soft-NMS	ResNet-101-FPN	41.3	60.7	45.3	23.5	44.9	52.9
Mask R-CNN +Rank-NMS	ResNet-101-FPN	41.6	61.3	45.4	23.7	45.1	53.5
Mask R-CNN +Rank-NMS +Soft-NMS	ResNet-101-FPN	42.0	61.1	46.2	23.9	45.5	53.9
Cascade R-CNN +Rank-NMS	ResNet-101-FPN	43.2	61.8	47.0	24.6	46.2	55.4
Cascade R-CNN +Rank-NMS +Soft-NMS	ResNet-101-FPN	43.7	61.6	48.1	24.9	46.8	56.1

Table 9. Analysis for the effects of the proposed Rank-NMS on different object detectors on MSCOCO 2017 validation set. We also report the running speed, counted on a single Titan P100 GPU, to analyze the efficiency of Rank-NMS.

Backbone	Method	+Rank-NMS	Inf Time(fps)	AP	AP^{50}	AP^{75}	AP^S	AP^M	AP^L
ResNet-50-FPN	Faster R-CNN	✗	9.3	36.4	58.4	39.1	21.6	40.1	46.6
	Faster R-CNN	✓	8.3	38.6	58.2	41.7	22.4	42.4	50.9
	Cascade R-CNN	✗	6.7	40.3	58.6	43. 9	22.9	43 8.	53.2
	Cascade R-CNN	✓	5.8	41.0	59.1	44.5	23.0	44 3.	54.8
	Mask R-CNN	✗	7.3	37.3	59.1	40.3	22.0	40.9	48.2
	Mask R-CNN	✓	6.2	39.3	58.8	42.3	22.8	42.7	52.2
ResNet-101-FPN	Faster R-CNN	✗	7.6	38.6	60.4	41.8	22.3	43.2	49.8
	Faster R-CNN	✓	7.0	40.3	60.1	43.5	23.5	44.4	53.7
	Cascade R-CNN	✗	5.8	42.7	61.6	46.6	23.8	46.2	57.4
	Cascade R-CNN	✓	5.5	42.8	60.9	46.7	24.9	46.3	57.2
	Mask R-CNN	✗	6.2	39.4	61.0	43.3	23.1	43.7	51.3
	Mask R-CNN	✓	5.6	41.1	60.6	44.7	23.5	45.1	55.1

in Figure 4. We can see that the proposed Rank-NMS can help generate more accurate detection results over traditional NMS under challenging scenarios, *e.g.,* low-light conditions (1st row), viewpoint variation (3rd row) and object overlapping (5th row). These results further demonstrate the effectiveness of the proposed Rank-NMS model for proposal selection.

5. Conclusion

In this paper, we propose a novel Learning-to-Rank (LTR) model to improve the proposal selection in the NSM procedure. In particular, LTR introduces the ranking loss to learn to predict the ranking score for the generated proposals from the region proposal networks, which provides more reliable criterion for ranking the proposals. To facilitate the training phase, we also propose a novel hard-pair sampling strategy to select the discriminative proposal pairs for learning the ranking score. We implement the LTR model with a small CNN, which can be inserted into current object detectors for end-to-end training and inference. Comprehensive experiments on multiple benchmarks demonstrate the outperforming accuracy of the proposed Rank-NMS, together with its generality to various object detectors.

References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-nms – improving object detection with one line of code. In *ICCV*, 2017.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *CVPR*, 2018.

[3] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. mmdetection. https://github.com/open-mmlab/mmdetection, 2018.

[4] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, 2016.

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.

[6] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[7] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *CVPR*, 2009.

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[9] Pedro Felzenszwalb, Ross Girshick, and David McAllester. Cascade object detection with deformable part models. In *CVPR*, 2010.

[10] Ross Girshick. Fast R-CNN. In *ICCV*, 2015.

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016.

[14] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *CVPR*, 2019.

[15] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *CVPR*, 2017.

[16] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *CVPR*, 2018.

[17] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *CVPR*, 2017.

[18] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *ECCV*, 2018.

[19] Xiaodan Liang, Tairui Wang, Luona Yang, and Eric Xing. CIRL: controllable imitative reinforcement learning for vision-based self-driving. In *ECCV*, 2018.

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

[24] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[25] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[26] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[28] Abhinav Shrivastava, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta. Beyond skip connections: Top-down modulation for object detection. *arXiv preprint arXiv:1612.06851*, 2016.

[29] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[30] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *ICLR Workshop*, 2016.

[31] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[32] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014.

[33] Lachlan Tychsen-Smith and Lars Petersson. Denet: Scalable real-time object detection with directed sparse sampling. In *ICCV*, 2017.

[34] Lachlan Tychsen-Smith and Lars Petersson. Improving object localization with fitness nms and bounded iou loss. In *CVPR*, 2018.

[35] Paul Viola, Michael Jones, et al. Rapid object detection using a boosted cascade of simple features. *CVPR*, 2001.