Revisiting RCNN: On Awakening the Classification Power of Faster RCNN

Bowen Cheng[1], Yunchao Wei[1]*, Honghui Shi[2],
Rogerio Feris[2], Jinjun Xiong[2], and Thomas Huang[1]

[1] University of Illinois at Urbana-Champaign, IL, USA
{bcheng9, yunchao, t-huang1}@illinois.edu
[2] IBM T.J. Watson Research Center, NY, USA
Honghui.Shi@ibm.com {rsferis, jinjun}@us.ibm.com

Abstract. Recent region-based object detectors are usually built with separate classification and localization branches on top of shared feature extraction networks. In this paper, we analyze failure cases of state-of-the-art detectors and observe that most *hard false positives* result from classification instead of localization. We conjecture that: (1) Shared feature representation is not optimal due to the mismatched goals of feature learning for classification and localization; (2) multi-task learning helps, yet optimization of the multi-task loss may result in sub-optimal for individual tasks; (3) large receptive field for different scales leads to redundant context information for small objects. We demonstrate the potential of detector classification power by a simple, effective, and widely-applicable *Decoupled Classification Refinement* (DCR) network. DCR samples hard false positives from the base classifier in Faster RCNN and trains a RCNN-styled strong classifier. Experiments show new state-of-the-art results on PASCAL VOC and COCO without any bells and whistles.

Keywords: Object Detection

1 Introduction

Region-based approaches with convolutional neural networks (CNNs) [2, 10, 11, 17–20, 27, 31, 33] have achieved great success in object detection. Such detectors are usually built with separate classification and localization branches on top of shared feature extraction networks, and trained with multi-task loss. In particular, Faster RCNN [27] learns one of the first end-to-end two-stage detector with remarkable efficiency and accuracy. Many follow-up works, such as R-FCN [3], Feature Pyramid Networks (FPN) [21], Deformable ConvNets (DCN) [4], have been leading popular detection benchmark in PASCAL VOC [6] and COCO [23] datasets in terms of accuracy. Yet, few work has been proposed to study what is the full potential of the classification power in Faster RCNN styled detectors.

---

[*] corresponding author

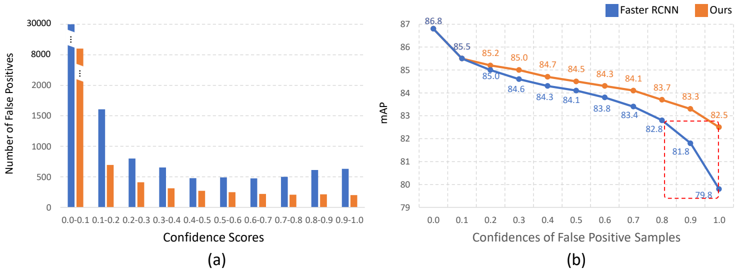


Fig. 1. (a) Comparison of the number of false positives in different ranges. (b) Comparison of the mAP gains by progressively removing false positives; from right to left, the detector is performing better as false positives are removed according to their confidence scores.

To answer this question, in this paper, we begin with investigating the key factors affecting the performance of Faster RCNN. As shown in Figure 1 (a), we conduct object detection on PASCAL VOC 2007 using Faster RCNN and count the number of false positive detections in different confidence score intervals (blue). Although only a small percentage of all false positives are predicted with high confidence scores, these samples lead to a significant performance drop in mean average precision (mAP). In particular, we perform an analysis of potential gains in mAP using Faster RCNN: As illustrated in Figure 1 (b), given the detection results from Faster RCNN and a confidence score threshold, we assume that all false positives with predicted confidence score above that threshold were classified correctly and we report the correspondent hypothesized mAP. It is evident that by correcting all false positives, Faster RCNN could, hypothetically, have achieved 86.8% in mAP instead of 79.8%. Moreover, even if we only eliminate false positives with high confidences, as indicated in the red box, we can still improve the detection performance significantly by 3.0% mAP, which is a desired yet hard-to-obtain boost for modern object detection systems.

The above observation motivates our work to alleviate the burden of false positives and improve the classification power of Faster RCNN based detectors. By scrutinizing the false positives produced by Faster RCNN, we conjecture that such errors are mainly due to three reasons: (1) Shared feature representation for both classification and localization may not be optimal for region proposal classification, the mismatched goals in feature learning lead to the reduced classification power of Faster RCNN; (2) Multi-task learning in general helps to improve the performance of object detectors as shown in Fast RCNN [10] and Faster RCNN, but the joint optimization also leads to possible sub-optimal to balance the goals of multiple tasks and could not directly utilize the full potential on individual tasks; (3) Receptive fields in deep CNNs such as ResNet-101 [15] are large, the whole image are usually fully covered for any given region proposals. Such large receptive fields could lead to inferior classification capacity by introducing redundant context information for small objects.

Following the above argument, we propose a simple yet effective approach, named Decoupled Classification Refinement (DCR), to eliminate high-scored false positives and improve the region proposal classification results. DCR decouples the classification and localization tasks in Faster RCNN styled detectors. It takes input from a base classifier, *e.g.* the Faster RCNN, and refine the classification results using a RCNN-styled network. DCR samples *hard false positives*, namely the false positives with high confidence scores, from the base classifier, and then trains a stronger correctional classifier for the classification refinement. Designedly, we do not share any parameters between the Faster RCNN and our DCR module, so that the DCR module can not only utilize the multi-task learning improved results from region proposal networks (RPN) and bounding box regression tasks, but also better optimize the newly introduced module to address the challenging classification cases.

We conduct extensive experiments based on different Faster RCNN styled detectors (*i.e.* Faster RCNN, Deformable ConvNets, FPN) and benchmarks (*i.e.* PASCAL VOC 2007 & 2012, COCO) to demonstrate the effectiveness of our proposed simple solution in enhancing the detection performance by alleviating hard false positives. As shown in Figure 1 (a), our approach can significantly reduce the number of hard false positives and boost the detection performance by 2.7% in mAP on PASCAL VOC 2007 over a strong baseline as indicated in Figure 1 (b). All of our experiment results demonstrate that our proposed DCR module can provide consistent improvements over various detection baselines, as shown in Figure 2. Our contributions are threefold:

1. We analyze the error modes of region-based object detectors and formulate the hypotheses that might cause these failure cases.
2. We propose a set of design principles to improve the classification power of Faster RCNN styled object detectors along with the DCR module based on the proposed design principles.
3. Our DCR module consistently brings significant performance boost to strong object detectors on popular benchmarks. In particular, following common practice (ResNet-101 as backbone), we achieve mAP of 84.0% and 81.2% on the classic PASCAL VOC 2007 and 2012, respectively, and 43.1% on the more challenging COCO2015 *test-dev*, which are the new state-of-the-art.

2 Related Work

Object Detection Recent CNN based object detectors can generally be categorized into two-stage and single stage. One of the first two-stage detector is RCNN [11], where selective search [29] is used to generate a set of region proposals for object candidates, then a deep neural network to extract feature vector of each region followed by SVM classifiers. SPPNet [14] improves the efficiency of RCNN by sharing feature extraction stage and use spatial pyramid pooling to extract fixed length feature for each proposal. Fast RCNN [10] improves over SPPNet by introducing an differentiable ROI Pooling operation to

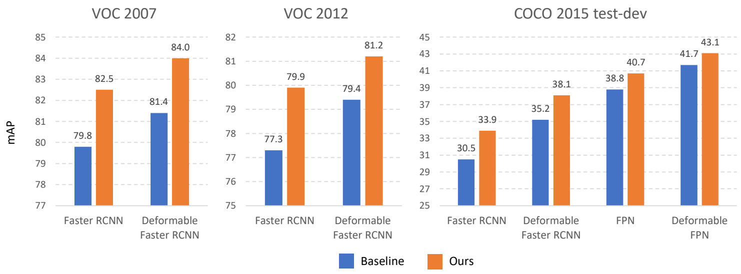
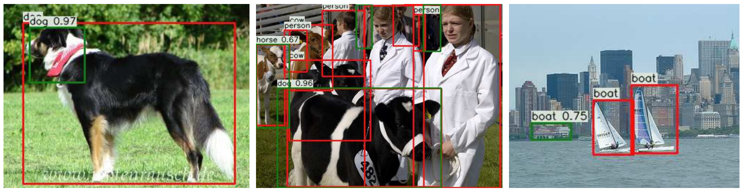


Fig. 2. Comparison of our approach and baseline in terms of different Faster RCNN series and benchmarks.

train the network end-to-end. Faster RCNN [27] embeds the region proposal step into a Region Proposal Network (RPN) that further reduce the proposal generation time. R-FCN [3] proposed a position sensitive ROI Pooling (PSROI Pooling) that can share computation among classification branch and bounding box regression branch. Deformable ConvNets (DCN) [4] further add deformable convolutions and deformable ROI Pooling operations, that use learned offsets to adjust position of each sampling bin in naive convolutions and ROI Pooling, to Faster RCNN. Feature Pyramid Networks (FPN) [21] add a top-down path with lateral connections to build a pyramid of features with different resolutions and attach detection heads to each level of the feature pyramid for making prediction. Finer feature maps are more useful for detecting small objects and thus a significant boost in small object detection is observed with FPN. Most of the current state-of-the-art object detectors are two-stage detectors based of Faster RCNN, because two-stage object detectors produce more accurate results and are easier to optimize. However, two-stage detectors are slow in speed and require very large input sizes due to the ROI Pooling operation. Aimed at achieving real time object detectors, one-stage method, such as OverFeat [28], SSD [9,24] and YOLO [25,26], predict object classes and locations directly. Though single stage methods are much faster than two-stage methods, their results are inferior and they need more extra data and extensive data augmentation to get better results. Our paper follows the method of two-stage detectors [10,11,27], but with a main focus on analyzing reasons why detectors make mistakes.

Classifier Cascade The method of classifier cascade commonly trains a stage classifier using misclassified examples from a previous classifier. This has been used a lot for object detection in the past. The Viola Jones Algorithm [30] for face detection used a hard cascades by Adaboost [8], where a strong region classifier is built with cascade of many weak classifier focusing attentions on different features and if any of the weak classifier rejects the window, there will be no more process. Soft cascades [1] improved [30] built each weak classifier based on the output of all previous classifiers. Deformable Part Model (DPM) [7] used a cascade of parts method where a root filter on coarse feature covering the



(a) (b) (c)

Fig. 3. Demonstration of hard false positives. Results are generate by Faster RCNN with 2 fully connected layer (2fc) as detector head [21,27], red boxes are ground truth, green boxes are hard false positives with scores higher than 0.3; (a) boxes covering only part of objects with high confidences; (b) incorrect classification due to similar objects; (c) misclassified backgrounds.

entire object is combined with some part filters on fine feature with greater localization accuracy. More recently, Li et al. [16] proposed the Convolutional Neural Network Cascade for fast face detection. Our paper proposed a method similar to the classifier cascade idea, however, they are different in the following aspects. The classifier cascade aims at producing an efficient classifier (mainly in speed) by cascade weak but fast classifiers and the weak classifiers are used to reject examples. In comparison, our method aims at improving the overall system accuracy, where exactly two strong classifiers are cascaded and they work together to make more accurate predictions. More recently, Cascade RCNN [2] proposes training object detector in a cascade manner with gradually increased IoU threshold to assign ground truth labels to align the testing metric, ie. average mAP with IOU 0.5:0.05:0.95.

3 Problems with Faster RCNN

Faster RCNN produces 3 typical types of hard false positives, as shown in Fig 3: (1) The classification is correct but the overlap between the predicted box and ground truth has low IoU, *e.g.* < 0.5 in Fig 3 (a). This type of false negative boxes usually cover the most discriminative part and have enough information to predict the correct classes due to translation invariance. (2) Incorrect classification for predicted boxes but the IoU with ground truth are large enough , *e.g.* in Fig 3 (b). It happens mainly because some classes share similar discriminative parts and the predicted box does not align well with the true object and happens to cover only the discriminative parts of confusion. Another reason is that the classifier used in the detector is not strong enough to distinguish between two similar classes. (3) the detection is a "confident" background, meaning that there is no intersection or small intersection with ground truth box but classifier's confidence score is large, *e.g.* in Fig 3 (c). Most of the background pattern in this case is similar to its predicted class and the classifier is too weak to distinguish. Another reason for this case is that the receptive field is fixed and it is too large for some box that it covers the actual object in its receptive field.

In Fig 3 (c), the misclassified background is close to a ground truth box (the left boat), and the large receptive field (covers more than 1000 pixels in ResNet-101) might "sees" too much object features to make the wrong prediction. Given above analysis, we can conclude that the hard false positives are mainly caused by the suboptimal classifier embedded in the detector. The reasons may be that: (1) feature sharing between classification and localization, (2) optimizing the sum of classification loss and localization loss, and (3) detector's receptive field does not change according to the size of objects.

Problem with Feature Sharing Detector backbones are usually adapted from image classification model and pre-trained on large image classification dataset. These backbones are original designed to learn scale invariant features for classification. Scale invariance is achieved by adding sub-sampling layers, *e.g.* max pooling, and data augmentation, *e.g.* random crop. Detectors place a classification branch and localization branch on top of the same backbone, however, classification needs **translation invariant** feature whereas localization needs **translation covariant** feature. During fine-tuning, the localization branch will force the backbone to gradually learn translation covariant feature, which might potentially down-grade the performance of classifier.

Problem with Optimization Faster RCNN series are built with a feature extractor as backbone and two task-specified branches for classifying regions and the other for localizing correct locations. Denote loss functions for classification and localization as L_{cls} and L_{bbox}, respectively. Then, the optimization of Faster RCNN series is to address a Multi-Task Learning (MTL) problem by minimizing the sum of two loss functions: $L_{detection} = L_{cls} + L_{bbox}$. However, the optimization might converge to a compromising suboptimal of two tasks by simultaneously considering the sum of two losses, instead of each of them.

Originally, such a MTL manner is found to be effective and observed improvement over state-wise learning in Fast(er) RCNN works. However, MTL for object detection is not studied under the recent powerful classification backbones, *e.g.* ResNets. Concretely, we hypothesize that MTL may work well based on a weak backbone (*e.g.* AlexNet or VGG16). As the backbone is getting stronger, the powerful classification capacity within the backbone may not be fully exploited and MTL becomes the bottleneck.

Problem with Receptive Field Deep convolutional neural networks have fixed receptive fields. For image classification, inputs are usually cropped and resized to have fixed sizes, *e.g.* 224 × 224, and network is designed to have a receptive field little larger than the input region. However, since contexts are cropped and objects with different scales are resized, the "effective receptive field" is covering the whole object.

Unlike image classification task where a single large object is in the center of a image, objects in detection task have various sizes over arbitrary locations.

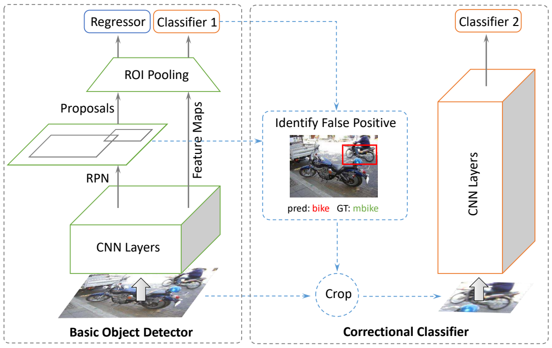


Fig. 4. Left: base detector (*e.g.* Faster RCNN). Right: our proposed Decoupled Classification Refinement (DCR) module.

In Faster RCNN, the ROI pooling is introduced to crop object from 2-D convolutional feature maps to a 1-D fixed size representation for the following classification, which results in fixed receptive field (*i.e.* the network is attending to a fixed-size window of the input image). In such a case, objects have various sizes and the fixed receptive field will introduce different amount of context. For a small object, the context might be too large for the network to focus on the object whereas for a large object, the receptive field might be too small that the network is looking at part of the object. Although some works introduce multi-scale features by aggregating features with different receptive field, the number of sizes is still too small comparing with the number various sizes of objects.

4 Revisiting RCNN for Improving Faster RCNN

In this section, we look back closely into the classic RCNN [11] method, and give an in-depth analysis of why RCNN can be used as a "complement" to improve Faster RCNN. Based on our findings, we provide a simple yet effective decoupled classification refinement module, that can be easily added to any current state-of-the-art object detectors to provide performance improvements.

4.1 Learning from RCNN Design

We train a modified RCNN with ResNet-50 as backbone and Faster RCNN predictions as region proposals. We find that with RCNN along, the detection result is deteriorated by more than 30% (from 79.8% to 44.7%)! Since RCNN does not modify box coordinate, the inferior result means worse classification. We find that many boxes having small intersections with an object are classified as that object instead of the background which Faster RCNN predicts. Based on this finding, we hypothesize that the drawback of RCNN is mainly root from that classification model is pre-trained without awaring object location. Since

ResNet-50 is trained on ImageNet in multi-crop manner, no matter how much the intersection of the crop to the object is, classifier is encouraged to predict that class. This leads to the classifier in RCNN being "too strong" for proposal classification, and this is why RCNN needs a carefully tuned sampling strategy, *i.e.* a ratio of 1:3 of fg to bg. Straightforwardly, we are interested whether RCNN is "strong" enough to correct hard negatives. We make a minor modification to multiply RCNN classification score with Faster RCNN classification score and observe a boost of 1.9% (from 79.8% to 81.7%)! Thus, we consider that RCNN can be seen as a compliment of Faster RCNN in the following sense: the classifier of Faster RCNN is weaker but aware of object location whereas the classifier of RCNN is unaware of object location but stronger. Based on our findings, we propose the following three principals to design a better object detector.

Decoupled Features Current detectors still place classification head and localization head on the same backbone, hence we propose that classification head and localization head should not share parameter (as the analysis given in Section 3), resulted in a decoupled feature using pattern by RCNN.

Decoupled Optimization RCNN also decouples the optimization for object proposal and classification. In this paper, we make a small change in optimization. We propose a novel two-stage training where, instead of optimizing the sum of classification and localization loss, we optimize the concatenation of classification and localization loss, $L_{detection} = [L_{cls} + L_{bbox}, L_{cls}]$, where each entry is being optimized independently in two steps.

Adaptive Receptive Field The most important advantage of RCNN is that its receptive field always covers the whole ROI, *i.e.* the receptive field size adjusts according to the size of the object by cropping and resizing each proposal to a fixed size. We agree that context information may be important for precise detection, however, we conjuncture that different amount of context introduced by fixed receptive filed might cause different performance to different sizes of objects. It leads to our last proposed principal that a detector should an adaptive receptive field that can change according to the size of objects it attends to. In this principal, the context introduced for each object should be proportional to its size, but how to decide the amount of context still remains an open question to be studied in the future. Another advantage of adaptive receptive field is that its features are well aligned to objects. Current detectors make predictions at hight-level, coarse feature maps usually having a large stride, *e.g.* a stride of 16 or 32 is used in Faster RCNN, due to sub-sampling operations. The sub-sampling introduces unaligned features, *e.g.* one cell shift on a feature map of stride 32 leads to 32 pixels shift on the image, and defects the predictions. With adaptive receptive field, the detector always attends to the entire object resulting in an aligned feature to make predictions. RCNN gives us a simple way to achieve adaptive receptive field, but how to find a more efficient way to do so remains an interesting problem needs studying.

4.2 Decoupled Classification Refinement (DCR)

Following these principals, we propose a DCR module that can be easily augmented to Faster RCNN as well as any object detector to build a stronger detector. The overall pipeline is shown in Fig 4. The left part and the right part are the original Faster RCNN and our proposed DCR module, respectively. In particular, DCR mainly consists a crop-resize layer and a strong classifier. The crop-resize layer takes two inputs, the original image and boxes produced by Faster RCNN, crops boxes on the original image and feeds them to the strong classifier after resizing them to a predefined size. Region scores of DCR module (Classifier 2) is aggregated with region scores of Faster RCNN (Classifier 1) by element-wise product to form the final score of each region. The two parts are trained separately in this paper and the scores are only combined during test time.

The DCR module does not share any feature with the detector backbone in order to preserve the quality of classification-aimed translation invariance feature. Furthermore, there is no error propagation between DCR module and the base detector, thus the optimization of one loss does not affect the other. This in turn results in a decoupled pattern where the base detector is focused more on localization whereas the DCR module focuses more on classification. DCR module introduces adaptive receptive field by resizing boxes to a predefined size. Noticed that this processing is very similar to moving an ROI Pooling from final feature maps to the image, however, it is quite different than doing ROI Pooling on feature maps. Even though the final output feature map sizes are the same, features from ROI Pooling sees larger region because objects embedded in an image has richer context. We truncated the context by cropping objects directly on the image and the network cannot see context outside object regions.

4.3 Training

Since there is no error propagates from the DCR module to Faster RCNN, we train our object detector in a two-step manner. First, we train Faster RCNN to converge. Then, we train our DCR module on mini-batches sampled from hard false positives of Faster RCNN. Parameters of DCR module are pre-trained by ImageNet dataset [5]. We follow the image-centric method [10] to sample N images with a total mini-batch size of R boxes, *i.e.* R/N boxes per image. We use $N = 1$ and $R = 32$ throughout experiments. We use a different sampling heuristic that we sample not only foreground and background boxes but also hard false positive **uniformly**. Because we do not want to apply any prior knowledge to impose unnecessary bias on classifier. However, we observed that boxes from the same image have little variance. Thus, we fix Batch Normalization layer with ImageNet training set statistics. The newly added linear classifier (fully connected layer) is set with 10 times of the base learning rate since we want to preserve translation invariance features learned on the ImageNet dataset.

Sample method	mAP
Baseline	79.8
Random	81.8
FP Only	81.4
FP+FG	81.6
FP+BG	80.3
FP+FG+BG	**82.3**
RCNN-like	81.7

(a)

FP Score	mAP
Baseline	79.8
0.20	82.2
0.25	81.9
0.30	**82.3**
0.35	82.2
0.40	82.0

(b)

Sample size	mAP
Baseline	79.8
8 Boxes	82.0
16 Boxes	82.1
32 Boxes	**82.3**
64 Boxes	82.1

(c)

ROI scale	mAP	Test Time
Baseline	79.8	0.0855
56 × 56	80.6	0.0525
112 × 112	82.0	0.1454
224 × 224	**82.3**	0.5481
320 × 320	82.0	1.0465

(d)

DCR Depth	mAP	Test Time
Baseline	79.8	0.0855
18	81.4	0.1941
34	81.9	0.3144
50	82.3	0.5481
101	82.3	0.9570
152	**82.5**	1.3900

(e)

Base detector	mAP
Faster	79.8
Faster+DCR	82.3
DCN	81.4
DCN+DCR	83.2

(f)

Model capacity	mAP
Faster w/ Res101	79.8
Faster w/ Res152	80.3
Faster Ensemble	81.1
Faster w/ Res101+DCR-50	82.3

(g)

Table 1. Ablation studies results. Evaluate on PASCAL VOC2007 test set. Baseline is Faster RCNN with ResNet-101 as backbone. DCR module uses ResNet-50. (a) Ablation study on sampling heuristics. (b) Ablation study on threshold for defining hard false positives. (c) Ablation study on sampling size. (d) Ablation study on ROI scale and test time (measured in seconds/image). (e) Ablation study on depth of DCR module and test time (measured in seconds/image). (f) DCR module with difference base detectors. Faster denotes Faster RCNN and DCN denotes Deformable Faster RCNN, both use ResNet-101 as backbone. (g) Comparison of Faster RCNN with same size as Faster RCNN + DCR.

5 Experiments

5.1 Implementation Details

We train base detectors, *e.g.* Faster RCNN, following their original implementations. We use default settings in 4.3 for DCR module, we use ROI size 224×224 and use a threshold of 0.3 to identify hard false positives. Our DCR module is first pre-trained on ILSVRC 2012 [5]. In fine-tuning, we set the initial learning rate to 0.0001 *w.r.t.* one GPU and weight decay of 0.0001. We follow linear scaling rule in [12] for data parallelism on multiple GPUs and use 4 GPUs for PASCAL VOC and 8 GPUs for COCO. Synchronized SGD with momentum 0.9 is used as optimizer. No data augmentation except horizontal flip is used.

5.2 Ablation Studies on PASCAL VOC

We comprehensively evaluate our method on the PASCAL VOC detection benchmark [6]. We use the union of VOC 2007 trainval and VOC 2012 trainval as well as their horizontal flip as training data and evaluate results on the VOC 2007 test set. We primarily evaluate the detection mAP with IoU 0.5 (mAP@0.5). Unless otherwise stated, all ablation studies are performed with ResNet-50 as classifier for our DCR module.

Ablation study on sampling heuristic We compare results with different sampling heuristic in training DCR module:

- random sample: a minibatch of ROIs are randomly sampled for each image
- hard false positive only: a minibatch of ROIs that are hard postives are sampled for each image
- hard false positive and background: a minibatch of ROIs that are either hard postives or background are sampled for each image
- hard false positive and foreground: a minibatch of ROIs that are either hard postives or foreground are sampled for each image
- hard false positive, background and foreground: the difference with random sample heuristic is that we ignore easy false positives during training.
- RCNN-like: we follow the Fast RCNN's sampling heuristic, we sample two images per GPU and 64 ROIs per image with fg:bg=1:3.

Results are shown in Table 1 (a). We find that the result is insensitive to sampling heuristic. Even with random sampling, an improvement of 2.0% in mAP is achieved. With only hard false positive, the DCR achieves an improvement of 1.6% already. Adding foreground examples only further gains a 0.2% increase. Adding background examples to false negatives harms the performance by a large margin of 1.1%. We hypothesize that this is because comparing to false positives, background examples dominating in most images results in a classifier bias to predicting background. This finding demonstrate the importance of hard negative in DCR training. Unlike RCNN-like detectors, we do not make any assumption of the distribution of hard false positives, foregrounds and backgrounds. To balance the training of classifier, we simply uniformly sample from the union set of hard false positives, foregrounds and backgrounds. This uniform sample heuristic gives the largest gain of 2.5% mAP. We also compare our training with RCNN-like training. Training with RCNN-like sampling heuristic with fg:bg=1:3 only gains a margin of 1.9%.

Ablation study on other hyperparameters We compare results with different threshold for defining hard false positive: [0.2, 0.25, 0.3, 0.35, 0.4]. Results are shown in Table 1 (b). We find that the results are quite insensitive to threshold of hard false positives and we argue that this is due to our robust uniform sampling heuristic. With hard false positive threshold of 0.3, the performance is the best with a gain of 2.5%. We also compare the influence of size of sampled RoIs during training: [8, 16, 32, 64]. Results are shown in Table 1 (c). Surprisingly, the difference of best and worst performance is only 0.3%, meaning our method is highly insensitive to the sampling size. With smaller sample size, the training is more efficient without severe drop in performance.

Speed and accuracy trade-off There are in general two ways to reduce inference speed, one is to reduce the size of input and the other one is to reduce the depth of the network. We compare 4 input sizes: 56×56, 112×112, 224×224, 320×320 as well as 5 depth choices: 18, 34, 50, 101, 152 and their speed. Results

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Faster [15]	76.4	79.8	80.7	76.2	68.3	55.9	85.1	85.3	89.8	56.7	87.8	69.4	88.3	88.9	80.9	78.4	41.7	78.6	79.8	85.3	72.0
R-FCN [3]	80.5	79.9	87.2	81.5	72.0	69.8	86.8	88.5	**89.8**	67.0	88.1	74.5	89.8	**90.6**	79.9	81.2	53.7	81.8	81.5	85.9	79.9
SSD [9, 24]	80.6	84.3	87.6	**82.6**	71.6	59.0	88.2	88.1	89.3	64.4	85.6	76.2	88.5	88.9	**87.5**	83.0	53.6	83.9	82.2	87.2	81.3
DSSD [9]	81.5	86.6	86.2	**82.6**	74.9	62.5	89.0	88.7	88.8	65.2	87.0	**78.7**	88.2	89.0	**87.5**	83.7	51.1	86.3	81.6	85.7	83.7
Faster (2fc)	79.8	79.6	87.5	79.5	72.8	66.7	88.5	88.0	88.9	64.5	84.8	71.9	88.7	88.2	84.8	79.8	53.8	80.3	81.4	**87.9**	78.5
Faster-Ours (2fc)	82.5	80.5	**89.2**	80.2	75.1	74.8	79.8	89.4	89.7	70.1	88.9	76.0	89.5	89.9	86.9	80.4	57.4	86.2	83.5	87.2	**85.3**
DCN (2fc)	81.4	83.9	85.4	80.1	75.9	68.8	88.4	88.6	89.2	68.0	87.2	75.5	89.5	89.0	86.3	84.8	54.1	85.2	82.6	86.2	80.3
DCN-Ours (2fc)	**84.0**	**89.3**	88.7	80.5	**77.7**	**76.3**	**90.1**	**89.6**	**89.8**	**72.9**	**89.2**	77.8	**90.1**	90.0	**87.5**	**87.2**	**58.6**	**88.2**	**84.3**	87.5	85.0

Table 2. PASCAL VOC2007 `test` detection results.

are shown in Table 1 (d) and (e). The test speed is linearly related to the area of input image size and there is a severe drop in accuracy if the image size is too small, *e.g.* 56×56. For the depth of classifier, deeper model results in more accurate predictions but also more test time. We also notice that the accuracy is correlated with the classification accuracy of classification model, which can be used as a guideline for selecting DCR module.

Generalization to more advanced object detectors We evaluate the DCR module on Faster RCNN and advanced Deformable Convolution Nets (DCN) [4]. Results are shown in Table 1 (f). Although DCN is already among one of the most accurate detectors, its classifier still produces hard false positives and our proposed DCR module is effective in eliminating those hard false positives.

Where is the gain coming from? One interesting question is where the accuracy gain comes from. Since we add a large convolutional network on top of the object detector, does the gain simply comes from more parameters? Or, is DCR an ensemble of two detectors? To answer this question, we compare the results of Faster RCNN with ResNet-152 as backbone (denoted Faster-152) and Faster RCNN with ResNet-101 backbone + DCR-50 (denoted Faster-101+DCR-50) and results are shown in Table 1 (g). Since the DCR module is simply a classifier, the two network have approximately the same number of parameters. However, we only observe a marginal gain of 0.5% with Faster-152 while our Faster-101+DCR-50 has a much larger gain of 2.5%. To show DCR is not simply then ensemble to two Faster RCNNs, we further ensemble Faster RCNN with ResNet-101 and ResNet-152 and the result is 81.1% which is still 1.1% worse than our Faster-101+DCR-50 model. This means that the capacity does not merely come from more parameters or ensemble of two detectors.

5.3 PASCAL VOC Results

VOC 2007 We use a union of VOC2007 trainval and VOC2012 trainval for training and we test on VOC2007 test. We use the default training setting and ResNet-152 as classifier for the DCR module. We train our model for 7 epochs and reduce learning rate by $\frac{1}{10}$ after 4.83 epochs. Results are shown in Table 2. Notice that based on DCN as base detector, our single DCR module achieves

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Faster [15]	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
R-FCN [3]	77.6	86.9	83.4	81.5	63.8	62.4	81.6	81.1	93.1	58.0	83.8	60.8	92.7	86.0	84.6	84.4	59.0	80.8	68.6	86.1	72.9
SSD [9, 24]	79.4	90.7	**87.3**	78.3	66.3	56.5	84.1	83.7	94.2	62.9	84.5	**66.3**	92.9	**88.6**	87.9	85.7	55.1	83.6	**74.3**	88.2	76.8
DSSD [9]	80.0	**92.1**	86.6	80.3	68.7	58.2	84.3	**85.0**	**94.6**	63.3	85.9	65.6	93.0	88.5	87.8	86.4	57.4	85.2	73.4	87.8	76.8
Faster (2fc)	77.3	87.3	82.6	78.8	66.8	59.8	82.5	80.3	92.6	58.8	82.3	61.4	91.3	86.3	84.3	84.6	57.3	80.9	68.3	87.5	71.4
Faster-Ours (2fc)	79.9	89.1	84.6	81.6	70.9	66.1	**84.4**	83.8	93.7	61.5	85.2	63.0	92.8	87.1	86.4	86.3	62.9	84.1	69.6	87.8	**76.9**
DCN (2fc)	79.4	87.9	86.2	81.6	71.1	62.1	83.1	83.0	94.2	61.0	84.5	63.9	93.1	87.9	87.2	86.1	60.4	84.0	70.5	**89.0**	72.1
DCN-Ours (2fc)	**81.2**	89.6	86.7	**83.8**	**72.8**	**68.4**	83.7	**85.0**	94.5	**64.1**	**86.6**	66.1	**94.3**	88.5	**88.5**	**87.2**	**63.7**	**85.6**	71.4	88.1	76.1

Table 3. PASCAL VOC2012 `test` detection results.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster (2fc)	ResNet-101	30.0	50.9	30.9	9.9	33.0	49.1
Faster-Ours (2fc)	ResNet-101 + ResNet-152	33.1	56.3	34.2	13.8	36.2	51.5
DCN (2fc)	ResNet-101	34.4	53.8	37.2	14.4	37.7	53.1
DCN-Ours (2fc)	ResNet-101 + ResNet-152	37.2	58.6	39.9	17.3	41.2	55.5
FPN	ResNet-101	38.2	61.1	41.9	21.8	42.3	50.3
FPN-Ours	ResNet-101 + ResNet-152	40.2	63.8	44.0	24.3	43.9	52.6
FPN-DCN	ResNet-101	41.4	63.5	45.3	24.4	45.0	55.1
FPN-DCN-Ours	ResNet-101 + ResNet-152	**42.6**	**65.3**	**46.5**	**26.4**	**46.1**	**56.4**

Table 4. COCO2014 `minival` detection results.

the new state-of-the-art result of 84.0% without using extra data (*e.g.* COCO data), multi scale training/testing, ensemble or other post processing tricks.

VOC 2012 We use a union of VOC2007 trainvaltest and VOC2012 trainval for training and we test on VOC2012 test. We use the same training setting of VOC2007. Results are shown in Table 3. Our model DCN-DCR is the first to achieve over 81.0% on the VOC2012 test set. The new state-of-the-art 81.2% is achieved using only single model, without any post processing tricks.

5.4 COCO Results

All experiments on COCO follow the default settings and use ResNet-152 for DCR module. We train our model for 8 epochs on the COCO dataset and reduce the learning rate by $\frac{1}{10}$ after 5.33 epochs. We report results on two different partition of COCO dataset. One partition is training on the union set of COCO2014 train and COCO2014 val35k together with 115k images and evaluate results on the COCO2014 minival with 5k images held out from the COCO2014 val. The other partition is training on the standard COCO2014 trainval with 120k images and evaluate on the COCO2015 test-dev. We use Faster RCNN [27], Feature Pyramid Networks (FPN) [21] and the Deformable ConvNets [4] as base detectors.

COCO minival Results are shown in Table 4. Our DCR module improves Faster RCNN by 3.1% from 30.0% to 33.1% in COCO AP metric. Faster RCNN with DCN is improved by 2.8% from 34.4% to 37.2% and FPN is improved by 2.0% from 38.2% to 40.2%. Notice that FPN+DCN is the base detector by top-3 teams in the COCO2017 detection challenge, but there is still an improvement of 1.2% from 41.4% to 42.6%. This observation shows that currently there is no

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
SSD [9, 24]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [9]	ResNet-101-DSSD	36.2	59.1	39.0	18.2	39.0	48.2
Mask RCNN [13]	ResNeXt-101-FPN [32]	39.8	62.3	43.4	22.1	43.2	51.2
RetinaNet [22]	ResNeXt-101-FPN	40.8	61.1	44.1	24.1	44.2	51.2
Faster (2fc)	ResNet-101	30.5	52.2	31.8	9.7	32.3	48.3
Faster-Ours (2fc)	ResNet-101 + ResNet-152	33.9	57.9	35.3	14.0	36.1	50.8
DCN (2fc)	ResNet-101	35.2	55.1	38.2	14.6	37.4	52.6
DCN-Ours (2fc)	ResNet-101 + ResNet-152	38.1	59.7	41.1	17.9	41.2	54.7
FPN	ResNet-101	38.8	61.7	42.6	21.9	42.1	49.7
FPN-Ours	ResNet-101 + ResNet-152	40.7	64.4	44.6	24.3	43.7	51.9
FPN-DCN	ResNet-101	41.7	64.0	45.9	23.7	44.7	53.4
FPN-DCN-Ours	ResNet-101 + ResNet-152	**43.1**	**66.1**	**47.3**	**25.8**	**45.9**	**55.3**

Table 5. COCO2015 `test-dev` detection results.

perfect detector that does not produce hard false positives.

COCO test-dev Results are shown in Table 5. The trend is similar to that on the COCO minival, with Faster RCNN improved from 30.5% to 33.9%, Faster RCNN+DCN improved from 35.2% to 38.1%, FPN improved from 38.8% to 40.7% and FPN+DCN improved from 41.7% to 43.1%. We also compare our results with recent state-of-the-arts reported in publications and our best model achieves state-of-the-art result on COCO2015 test-dev with ResNet as backbone.

6 Conclusion

In this paper, we analyze error modes of state-of-the-art region-based object detectors and study their potentials in accuracy improvement. We hypothesize that good object detectors should be designed following three principles: decoupled features, decoupled optimization and adaptive receptive field. Based on these principles, we propose a simple, effective and widely-applicable DCR module that achieves new state-of-the-art. In the future, we will further study what architecture makes a good object detector, adaptive feature representation in multi-task learning, and efficiency improvement of our DCR module.

Acknowledgements. This work is in part supported by IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) - a research collaboration as part of the IBM AI Horizons Network; and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number D17PC00341. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government. We thank Jiashi Feng for helpful discussions.

References

1. Bourdev, L., Brandt, J.: Robust object detection via soft cascade. In: IEEE CVPR. vol. 2, pp. 236–243 (2005)
2. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: IEEE CVPR (June 2018)
3. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: NIPS. pp. 379–387 (2016)
4. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: IEEE ICCV. pp. 764–773 (2017)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE CVPR. pp. 248–255 (2009)
6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV **88**(2), 303–338 (Jun 2010)
7. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE TPAMI **32**(9), 1627–1645 (2010)
8. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences **55**(1), 119–139 (1997)
9. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659 (2017)
10. Girshick, R.: Fast r-cnn. In: IEEE ICCV. pp. 1440–1448 (2015)
11. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE CVPR. pp. 580–587 (2014)
12. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K.: Accurate, large minibatch sgd: training imagenet in 1 hour. arXiv preprint arXiv:1706.02677 (2017)
13. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: IEEE ICCV. pp. 2980–2988 (2017)
14. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: ECCV. pp. 346–361 (2014)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE CVPR. pp. 770–778 (2016)
16. Li, H., Lin, Z., Shen, X., Brandt, J., Hua, G.: A convolutional neural network cascade for face detection. In: IEEE CVPR. pp. 5325–5334 (2015)
17. Li, J., Liang, X., Li, J., Wei, Y., Xu, T., Feng, J., Yan, S.: Multistage object detection with group recursive learning. IEEE Transactions on Multimedia **20**(7), 1645–1655 (2018)
18. Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., Yan, S.: Perceptual generative adversarial networks for small object detection. In: IEEE CVPR (2017)
19. Li, J., Wei, Y., Liang, X., Dong, J., Xu, T., Feng, J., Yan, S.: Attentive contexts for object detection. IEEE Transactions on Multimedia **19**(5), 944–954 (2017)
20. Liang, X., Liu, S., Wei, Y., Liu, L., Lin, L., Yan, S.: Towards computational baby learning: A weakly-supervised approach for object detection. In: IEEE ICCV. pp. 999–1007 (2015)
21. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE CVPR. vol. 1, p. 4 (2017)

22. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. In: IEEE ICCV. pp. 2980–2988 (2017)
23. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755 (2014)
24. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV. pp. 21–37 (2016)
25. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: IEEE CVPR. pp. 779–788 (2016)
26. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. In: IEEE CVPR. pp. 6517–6525 (2017)
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS. pp. 91–99 (2015)
28. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)
29. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. IJCV **104**(2), 154–171 (2013)
30. Viola, P., Jones, M.J.: Robust real-time face detection. IJCV **57**(2), 137–154 (2004)
31. Wei, Y., Shen, Z., Cheng, B., Shi, H., Xiong, J., Feng, J., Huang, T.: Ts2c: Tight box mining with surrounding segmentation context for weakly supervised object detection. In: ECCV (2018)
32. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: IEEE CVPR. pp. 5987–5995 (2017)
33. Xu, H., Lv, X., Wang, X., Ren, Z., Chellappa, R.: Deep regionlets for object detection. arXiv preprint arXiv:1712.02408 (2017)