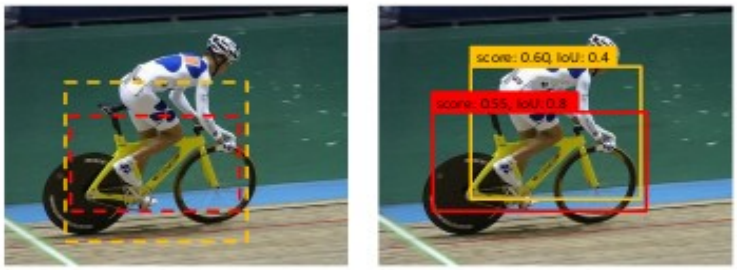
Consistent Optimization for Single-Shot Object Detection

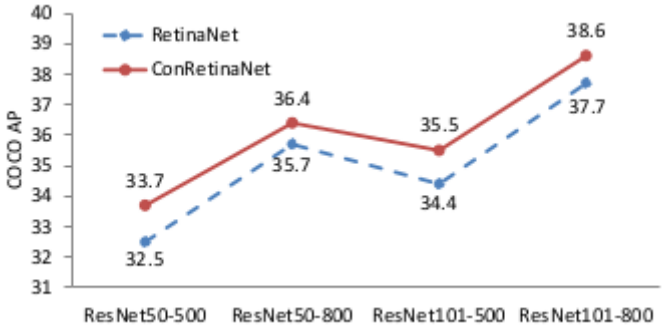
INTRODUCTION

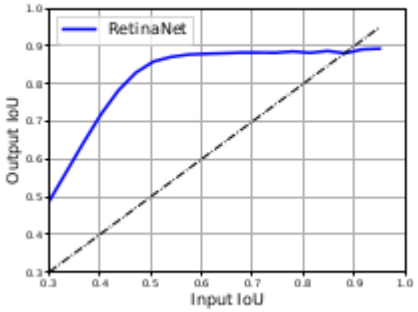
Current state-of-the-art deep learning based object detection frameworks can be generally divided into two major groups: two-stage, proposal-driven methods and one-stage, proposal-free methods.



(a) (b)

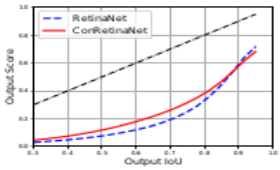
Model is named as ConRetinaNet. A vanilla implementation outperforms RetinaNet with different model capacities (ResNet-50/ResNet-101), input resolutions (short-size from 500 to 800), and localization qualities on challenging MS COCO dataset. In particular, ConRetinaNet is able to achieve 40.1 AP on the MS COCO dataset, which is the first ResNet-101 based single stage object detector to achieve such performance without any testing time bells or whistles.



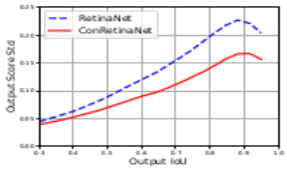


SINGLE SHOT OBJECT DETECTION

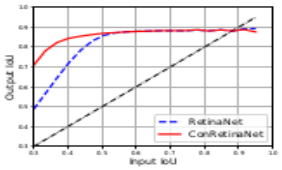
The key idea of the single shot object detector is to associate the pre-defined anchor which is centered at each feature map location with the convolutional operation and results. The single stage detector is composed of a backbone network and two task-specific sub-networks. The backbone is responsible for computing a convolutional feature map over an entire input image and is an off-the-shelf convolutional network.



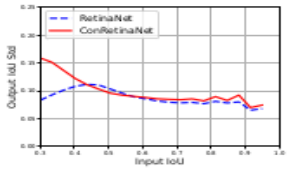
(a) Detection Output Mean



(b) Detection Output Std



(c) Regression Output Mean



(d) Regression Output std

The score of the the refined box is given by the classification sub-networks. Finally, non-maximum-suppression (NMS) is applied to remove duplicated bounding boxes.

THE MISALIGNMENTS

There are two misalignments between training and inference in the current single stage detector.
(a) The localization qualities between original anchors and the refined anchors are very different.
(b) Anchors whose IoU overlap with the groundtruth lower than 0.5 are treated as negative samples.
The localization performance is evaluated as a function of the input IoUs. Utilizing the regressed anchors are more accurate for the training of a high quality object detector.

CONSISTENT OPTIMIZATION (ALGORITHMIC CHANGE)

Consistent Detection: The loose training signal for the classification sub-networks has hindered the accuracy of the detector, since the behaviors between the training anchors and refined anchors at prediction phase are different.

$$L_{cls} = \frac{1}{N_{cls}} \sum_i [L_{cls}(c_i, c_i^*) + \alpha L_{cls}(c_i, c_i^\dagger)]. \quad (1)$$

Here, i is the index of an anchor in a mini-batch and ci is the predicted probability of the (refined) anchor i being an object. The ground-truth label c*i is the label for the original anchor i, while c † i is the label for the refined one. Ncls is the mini-batch size for the classification branch. α balances the weights between two terms. Combining the two loss together makes the training process more stable, and does not harm the performance.

Consistent Localization: RetinaNet seems to produce tight boxes, which is different from the observations of two-stage object detectors. To keep consistency with the classification branch, we also add the subsequent regressions. The localization loss function becomes

$$L_{reg} = \frac{1}{N_{reg}^0} \sum_i L_{reg}^0(t_i^0, t_i^*) + \frac{1}{N_{reg}^1} \sum_i L_{reg}^1(t_i^1, t_i^\dagger), \quad (2)$$

where t0 i is the predicted offset of the original anchor i, t1 i the predicted offset of the refined one. t* and t † are corresponding groundtruch offsets for the original and refined anchor. N reg is the mini-batch size for the regression branch.

ADVANTAGES

Combining the loss : Combining the loss together makes the training process more stable, and does not harm the performance overall all increases the productivity of the system as a whole.
Maintained the consistency : To keep consistency with the classification branch, add the subsequent regressions and through this way one can maintain the consistency of the system.
Boost to the performance : Due to the simple and effective consistent optimization one can boost the performance due to the significant amount.
Requires no addtional parameters : Utilizing consistent optimization requires almost no additional parameters, and it shows its effectiveness.

DISADVANTAGES

Negative impact of adding more regression stages : Adding more regression stages leads to a slight performance decrease.
No improvement on increasing classification terms : Including more classification terms leads to no improvement.

COMPARISON TO THE STATE OF ART

The consistent optimization extension of RetinaNet, is compared to state-of-the-art object detectors. The standard COCO metrics including AP , AP50, AP75, and APS, APM, APL on the test-dev set. The first group of detectors are two-stage detectors, the second group one-stage detectors, and the last group the consistent optimization extension of RetinaNet. The extension from RetinaNet to ConRetinaNet improves detection performance by ∼1 point, and it also outperforms all single-stage detector under ResNet-101 backbone, under all evaluation metrics. This includes the very recent one-stage RefineDet and two-stage relation networks.

An intersecting direction is to utilize the region based branch to get more accurate features. The deformable convolution shows better capability to model the geometric transformation of the objects. Replacing the backbone networks with Deformable ConvNets is supposed to get better performance, which is beyond the focus of this paper. At last, Cascade R-CNN and Deformbale ConvNets both require more parameters to get such results.

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
two-stage							
Faster R-CNN+++[15]*	ResNet-101	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN by G-RMI[17]	Inception-ResNet-v2	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w FPN[23]	ResNet-101	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN w TDM[36]	Inception-ResNet-v2	36.8	57.7	39.2	16.2	39.8	52.1
Deformable R-FCN [6]*	Aligned-Inception-ResNet	37.5	58.0	40.8	19.4	40.1	52.5
Mask R-CNN[13]	ResNet-101	38.2	60.3	41.7	20.1	41.1	50.2
Relation[16]	DCN-101	39.0	58.6	42.9	-	-	-
Regionlets [16]	ResNet-101	39.3	59.8	-	21.7	43.7	50.9
DeNet768 [39]	ResNet-101	39.5	58.0	42.6	18.9	43.5	54.1
IoU-Net[18]	ResNet-101	40.6	59.0	-	-	-	-
Cascade R-CNN[4]	ResNet-101	42.8	62.1	46.3	23.7	45.5	55.2
Faster R-CNN by MSRA[45]	DCN-v2-101	44.0	65.9	48.1	23.2	47.7	59.6
one-stage							
YOLOv2 [31]	DarkNet-19	21.6	44.0	19.2	5.0	22.4	35.5
RON384 [20]*	VGG-16	27.4	49.5	27.1	-	-	-
SSD513 [10]	ResNet-101	31.2	50.4	33.3	10.2	34.5	49.8
YOLOv3 [33]	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9
DSSD513 [10]	ResNet-101	33.2	53.3	35.2	13.0	35.4	51.1
RefineDet512 [44]	ResNet-101	36.4	57.5	39.5	16.6	39.9	51.4
RetinaNet [24]	ResNet-101	39.1	59.1	42.3	21.8	42.7	50.2
CornerNet511[22]*	Hourglass-104	40.5	56.5	43.1	19.4	42.7	53.9
ours							
ConRetinaNet	ResNet-50	37.3	56.4	40.3	21.3	40.2	51.0
ConRetinaNet	ResNet-101	40.1	59.6	43.5	23.4	44.2	53.3

CONCLUSION

The simple and effective consistent optimization used to boost the performance of single stage object detectors. By examination of the model behaviors, it is find that the optimization misalignment between training and inference is the bottleneck to get better results. Conducted extensive experiments to compare different model design choices, and demonstrate that the consistent optimization is the most important factor. Utilizing consistent optimization requires almost no additional parameters, and it shows its effectiveness using the strong RetinaNet baseline on challenging MS COCO dataset.