

SCALoss: Side and Corner Aligned Loss for Bounding Box Regression

Tu Zheng, Shuai Zhao, Yang Liu, Zili Liu, Deng Cai
Zhejiang University

arXiv:2104.00462v1 [cs.CV] 1 Apr 2021

Abstract—Bounding box regression is an important component in object detection. Recent work achieves promising performance by optimizing the Intersection over Union (IoU) as loss. However, IoU-based loss has the gradient vanish problem in the case of low overlapping bounding boxes, and the model could easily ignore these simple cases. In this paper, we propose Side Overlap (SO) loss by maximizing the side overlap of two bounding boxes, which puts more penalty for low overlapping bounding box cases. Besides, to speed up the convergence, the Corner Distance (CD) is added into the objective function. Combining the Side Overlap and Corner Distance, we get a new regression objective function, *Side and Corner Align Loss (SCALoss)*. The SCALoss is well-correlated with IoU loss, which also benefits the evaluation metric but produces more penalty for low-overlapping cases. It can serve as a comprehensive similarity measure, leading the better localization performance and faster convergence speed. Experiments on COCO and PASCAL VOC benchmarks show that SCALoss can bring consistent improvement and outperform ℓ_n loss and IoU based loss with popular object detectors such as YOLOV3, SSD, Reppoints, Faster-RCNN.

Index Terms—Object Detection, Bounding Box Regression, SCALoss

I. INTRODUCTION

OBJECT detection has been improved rapidly with the development of advanced deep convolutional neural networks. A series of state-of-the-art CNN-based detectors emerge in recent years, such as Faster R-CNN [1], SSD [2], YOLOV3 [3], Reppoints [4], and *etc*. Generally, object detection consists of object classification and object localization. Current state-of-the-art object detectors (*e.g.* Faster-RCNN, Mask R-CNN [5], RetinaNet [6]) have shown the importance of bounding box regression in object detection pipeline. In this paper, we focus on the problem of object localization.

Intersection over Union (IoU) is the most popular evaluation metric for bounding box regression. In the existing methods, ℓ_n loss is the widely used loss, but it is not tailored to the evaluation metric (IoU). Thus, IoU [7] is proposed to directly optimize the evaluation metric. However, IoU is infeasible to optimize in the case of non-overlapping bounding boxes. Then Generalized IoU (GIoU) [8] addresses this weakness by introducing a generalized version as the new loss. After that, Distance IoU (DIoU) [9] adds the normalized center distance between the predicted box and the target box, which helps converge faster than GIoU. Although the IoU-bases loss can achieve the more accurate result than ℓ_n loss, they still have several limitations as shown in Fig. 1. Below, we describe these issues in turn:

1) Gradient Vanish problem: IoU-based methods (IoU, GIoU, DIoU) improve baseline for high overlapping metric

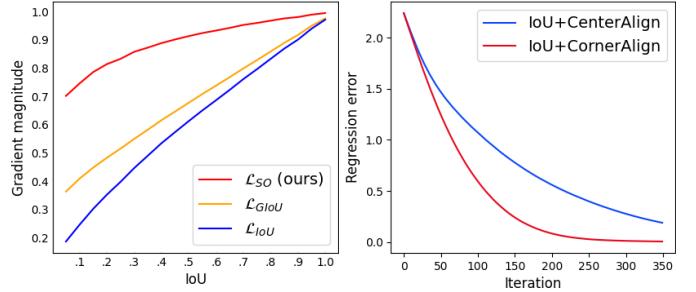


Fig. 1: (left) The relationship between gradient magnitude and IoU. Gradient magnitude is the mean of $\|\frac{\partial \mathcal{L}}{\partial x}\|_2$ in different IoU intervals, where IoU interval = {[0., 0.05], [0.05, 0.1], ..., [0.95, 1.0]}, $x = (x_1, y_1, x_2, y_2)$ is the bounding box corner point. The gradient of \mathcal{L}_{IOU} significantly drops in low overlapping cases, but \mathcal{L}_{SO} still has a large gradient. (right) Compare with center alignment and corner alignment and the loss weight of corner alignment is 0.5x as center alignment. The result is calculated with 100k random bounding boxes.

like AP75, but have relatively inferior performance in AP50. We further investigate this phenomenon and notice that IoU loss will lead to gradient vanish problem for non-overlapping cases and produce small gradient for low overlapping cases. In Fig. 1 (left), we visualize the relationship between gradient magnitude and IoU for different loss functions. It shows that lower IoU cases will have relatively smaller gradient value. During the training process, the small gradients produced by low overlapping boxes (hard samples) may be drowned into the large gradients by high overlapping ones (easy samples), thus limiting the overall performance. Since IoU is a component of GIoU and DIoU, they still encounter this problem. When predicted boxes lie within ground truth boxes, as shown in Fig. 2, all IoU value is the same and GIoU degrades into IoU. When the center of box is close to its ground truth, the normalized center distance in DIoU is near zero. In this case, the DIoU is roughly the same as IoU. For the aforementioned cases, GIoU and DIoU still produce small gradient, resulting in inferior performance.

2) Slow convergence speed: Although DIoU [9] can speed up the convergence to a certain, the designed objective function is still not optimal. As DIoU discusses, GIoU tends to increase the size of box for non-overlapping cases until it has overlap with the ground truth box, which makes GIoU slow for convergence. Thus, DIoU adds a penalty term, *i.e.*,

the normalized center distance to directly "pull" closer boxes, which makes the DIoU convergence faster than GIoU. However, as Fig. 2 shows, DIoU contributes little when the center of predicted box is near target box. In these cases, the corner distance is still far from the ground truth box. We further compare center alignment (normalized center distance) with the corner alignment (normalized corner distance) as shown in Fig. 1 (right). It illustrates the corner alignment converges faster than center alignment. Therefore, regressing the two corner points can be a better choice.

In this work, we propose *Side and Corner Aligned Loss (SCALoss)* to solve the shortcoming of IoUs and speed up the convergence. It is a combination of *Side Overlap (SO) loss* (\mathcal{L}_{SO} , Eq. (2)) and *Corner Distance (CD) loss* (\mathcal{L}_{CD} , Eq. (3)). The Side Overlap maximizes the side overlap of bounding boxes, which puts more penalty for low-overlapping cases and focuses more on hard samples. As shown in Fig. 1 (left), SO still keeps a large gradient in the low overlapping cases, while the gradient of IoU significantly drops. Furthermore, SO loss is well-correlated with IoU loss (see the Sec. III-B Relationship with IoU and GIoU). Specifically, it can also benefit the evaluation metric (IoU). The Corner Distance adds the normalized corner distance to achieve accurate corner alignment and faster convergence speed. By incorporating the Side Overlap loss and Corner Distance loss, SCALoss can serve a more comprehensive similarity measure, leading the better localization performance and faster convergence speed.

To demonstrate the generality of SCALoss, we evaluate it with various CNN-based object detection frameworks including YOLOV3 [3], SSD [2], Reppoints [4], Faster R-CNN [1] on PASCAL VOC [10] and MS-COCO [11] dataset. Experimental results demonstrate that our approach achieves better object localization accuracy and gets consistent improvements.

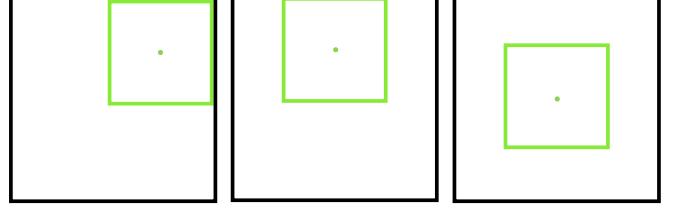
Our contributions can be summarized as follows:

- We show that IoU based method has the gradient problem for low overlapping bounding boxes and the normalized corner distance can speed up convergence.
- We propose SCALoss to evaluate the similarity of bounding boxes by using corner points and box sides, which outperforms ℓ_n loss and IoU-based loss (including IoU [7], GIoU [8], DIoU [9], and CIoU [9]). It can be easily plugged into any detection framework to achieve better localization accuracy.
- We experimentally demonstrate that SCALoss can achieve noticeable and consistent improvement with different detection frameworks on PASCAL VOC and COCO benchmarks.

II. RELATED WORK

A. Object Detection

Current object detection methods can be roughly categorized into two classes: anchor-based detectors and anchor-free detectors. Anchor-based detectors can be divided into two-stage and one-stage methods.



$$\begin{array}{lll} \mathcal{L}_{IoU} = \mathcal{L}_{GIoU} = 0.75 & \mathcal{L}_{IoU} = \mathcal{L}_{GIoU} = 0.75 & \mathcal{L}_{IoU} = \mathcal{L}_{GIoU} = 0.75 \\ \mathcal{L}_{DIoU} = 0.81 & \mathcal{L}_{DIoU} = 0.78 & \mathcal{L}_{DIoU} = 0.75 \\ \mathcal{L}_{Center} = 0.06 & \mathcal{L}_{Center} = 0.03 & \mathcal{L}_{Center} = 0.0 \\ \mathcal{L}_{CD} = 0.25 & \mathcal{L}_{CD} = 0.18 & \mathcal{L}_{CD} = 0.12 \\ \mathcal{L}_{SCA} = 1.25 & \mathcal{L}_{SCA} = 1.18 & \mathcal{L}_{SCA} = 1.12 \end{array}$$

Fig. 2: Green and black bounding box denote the predicted box and ground truth box respectively. \mathcal{L}_{CD} is Eq.(3) and \mathcal{L}_{Center} is the normalized center distance in DIoU. GIoU degrades into IoU and DIoU heavily relies IoU for these cases.

1) *Anchor-based Detectors*: Anchor-based detectors consist of two-stage detectors and one-stage detectors. For two-stage detectors, R-CNN based methods [12], [13], [1] generate object proposals with sliding window for second stage classifier as well as bounding box refinement. After that, lots of algorithms are proposed to improve its performance [14], [15], [16], [17], [18]. Compared to two-stage methods, the one-stage detectors directly predict bounding boxes and class scores without object proposal generation such as SSD [2] and YOLO series [19], [20], [3], [21], [22]. Thereafter, plenty of works are presented to boost its performance[23], [24], [25]. These methods are superior in inference speed but inferior in accuracy compared to two-stage methods. Among these methods, Focal loss [6] solves the problem of extreme foreground-background class imbalance. Generally, one-stage method is considered to be promising to achieve similar accuracy with two-stage method.

2) *Anchor-free Detectors*: Anchor-free detectors mainly locate several pre-defined keypoints and generate bounding boxes to detect objects. CornerNet [26] detects an object bounding box as a pair of keypoints while CenterNet [27] detects object center and regress the size of the object. ExtremeNet [28] detects four extreme points and one center to generate the object bounding box. Reppoints [4] represents objects as a set of sample points to adaptively position themselves over an object and utilizes deformable convolution [29] to get more accurate features. These anchor-free detectors are able to eliminate those hyper-parameters related to anchors and have achieved similar performance with anchor-based detectors.

B. Bounding box representation

To achieve robust optimization and better localization, the bounding box representation is a crucial component for these detectors. YOLO [19] proposes to directly predict bounding box parameters with the square root of the bounding box size for scale sensitivity. In RCNN series [12], [13], [1] and SSD [23], they introduce the concept of anchor and the predicted parameter is transformed as the residual of predicted and ground truth boxes. Reppoints [4] directly regress the top-left and bottom-right corners to avoid the tuning of weights

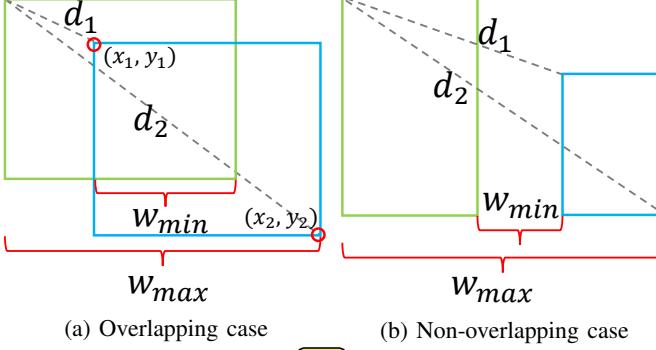


Fig. 3: Side and Corner Aligned loss for bounding box regression. It directly regresses two corner points by minimizing the normalized distance $\frac{d_1}{d_2}$ and enlarge width by minimizing the $1 - \frac{w_{min}}{w_{max}}$. We omit another corner and height for simplicity.

as done in computing the distance between bounding box regression vectors (size and center).

C. Bounding box Regression Loss

Various bounding box regression losses have been proposed in recent years. ℓ_1 -smooth loss [13] proposes to combine ℓ_1 loss and ℓ_2 loss so that the loss is less sensitive to outliers and more stable for inliers. Balanced L1 loss [30] proposes to promote the crucial regression gradients (inliers) for the balance between classification and localization. Bounded IoU loss [31] derives a novel bounding box regression loss based on a set of IoU upper bounds that better matches the goal of IoU maximization while still providing good convergence properties. KLLoss [32] proposes a bounding box regression loss for learning bounding box transformation and localization variance together. The learned localization variance can merge neighboring bounding boxes during non-maximum suppression (NMS), which further improves the localization performance. UnitBox [7] first proposes IoU Loss for object detection, which regresses the bounding box as a whole unit. GIoU [8] discusses the weakness of IoU for the case of non-overlapping bounding boxes and introduces a generalized version of IoU as a new loss. DIoU [9] adds the normalized center distance between the predicted and the target box on IoU loss, which helps converge faster in training.

III. APPROACH

In this section, we first introduce our Side and Corner Aligned loss for bounding box regression, then we analyze the SO loss and compare it with IoU based loss.

A. Side and Corner Aligned Loss

Following [32], we regress the corners of a bounding box separately. We adopt the parameter of the $(x_1, y_1, x_2, y_2) \in \mathcal{R}^4$ coordinate as bounding box representation, where (x_1, y_1) , (x_2, y_2) are top left and bottom right corner respectively. Our loss function includes side overlap (SO) and corner distance (CD) two parts.

Algorithm 1 SCA as a bounding box loss

Input: Predicted B^p and ground truth B^g bounding box coordinates: $B^p = (x_1^p, y_1^p, x_2^p, y_2^p)$, $B^g = (x_1^g, y_1^g, x_2^g, y_2^g)$, weight factor α

Output: \mathcal{L}_{SCA}

- 1: Calculating intersection between B^p and B^g :
 $x_1^I = \max(x_1^p, x_1^g), y_1^I = \max(y_1^p, y_1^g)$
 $x_2^I = \max(x_2^p, x_2^g), y_2^I = \max(y_2^p, y_2^g)$
 - 2: Calculating w_{min} and h_{min} : $w_{min} = x_2^I - x_1^I, h_{min} = y_2^I - y_1^I$
 - 3: Finding the coordinate of the smallest enclosing box:
 $x_1^C = \min(x_1^p, x_1^g), y_1^C = \min(y_1^p, y_1^g)$
 $x_2^C = \max(x_2^p, x_2^g), y_2^C = \max(y_2^p, y_2^g)$
 - 4: Calculating w_{max} and h_{max} : $w_{max} = x_2^C - x_1^C, h_{max} = y_2^C - y_1^C$
 - 5: $\mathcal{L}_{SO} = 2 - (\frac{h_{min}}{h_{max}} + \frac{w_{max}}{w_{min}})$
 - 6: Calculating two corner points distance:
 $D_{lt} = (x_1^p - x_1^g)^2 + (y_1^p - y_1^g)^2$
 $D_{rb} = (x_2^p - x_2^g)^2 + (y_2^p - y_2^g)^2$
 - 7: Calculating the diagonal length of the smallest enclosing box: $D_{diag} = (x_2^C - x_1^C)^2 + (y_2^C - y_1^C)^2$
 - 8: $\mathcal{L}_{CD} = \frac{D_{lt}}{D_{diag}} + \frac{D_{rb}}{D_{diag}}$
 - 9: $\mathcal{L}_{SCA} = \mathcal{L}_{SO} + \alpha \mathcal{L}_{CD}$
 - 10: **return** \mathcal{L}_{SCA}
-

1) *Side Overlap*: We propose Side Overlap (SO) loss to measure bounding box similarity by maximizing the overlap of width and height. It is a more stricter constraint and puts more gradient for low overlapping bounding box. As shown in Fig. 3, given the predicted box (x_1, x_2, y_1, y_2) and ground truth box $(x_1^g, y_1^g, x_2^g, y_2^g)$, the SO loss simultaneously maximizes the overlap for both sides of a predicted box with its ground truth. SO is defined as follows:

$$SO = \frac{w_{min}}{w_{max}} + \frac{h_{min}}{h_{max}}, \quad (1)$$

where $w_{min} = \min(x_2, x_2^g) - \max(x_1, x_1^g)$, $w_{max} = \max(x_2, x_2^g) - \min(x_1, x_1^g)$, $h_{min} = \min(y_2, y_2^g) - \max(y_1, y_1^g)$, $h_{max} = \max(y_2, y_2^g) - \min(y_1, y_1^g)$. Note w_{min}, h_{min} may be negative when bounding boxes are non-overlapping as shown in Fig. 3b. Thus, SO loss can also be optimized for non-overlapping cases. The SO loss can be formulated as follows:

$$\mathcal{L}_{SO} = 2 - SO. \quad (2)$$

2) *Corner Distance*: Furthermore, we introduce Corner Distance (CD) loss to achieve better corner alignment. As shown in Fig. 2, the normalized center distance in DIoU is roughly near zero in these cases, but the corner point still misaligns. Therefore, we add CD in the loss to achieve accurate box regression. The CD directly minimizes the normalized corner distance. It is defined as follows:

$$\mathcal{L}_{CD} = \frac{D(p_1, p_1^g)}{D(p_{c_1}, p_{c_2})} + \frac{D(p_2, p_2^g)}{D(p_{c_1}, p_{c_2})}, \quad (3)$$

where $D(\cdot, \cdot)$ is the Euclidean distance, p_1, p_2 denote the top left and bottom right corner points of predicted box, p_1^g, p_2^g are

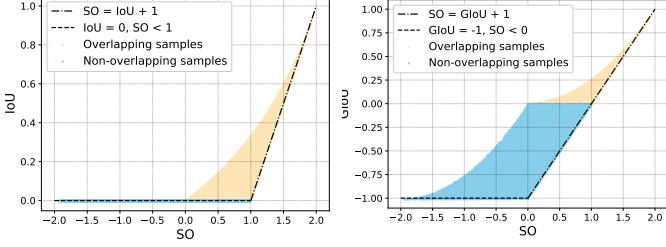


Fig. 4: (left) Relationship between IoU and SO, (right) Relationship between GIoU and SO for overlapping and non-overlapping samples.

corresponding ground truth points, p_{c_1}, p_{c_2} are corner points of the smallest enclosing box covering two boxes.

The final SCALoss can be formulated as follows:

$$\mathcal{L}_{SCA} = \mathcal{L}_{SO} + \alpha \mathcal{L}_{CD}, \quad (4)$$

where α is the weight factor, and α is set to 0.2 in our experiment. The detail of the algorithm is shown in Algorithm 1.

B. Relationship with IoU and GIoU

For two arbitrary axis-aligned bounding boxes $A, B \in \mathcal{R}^4$, we can calculate the SO, IoU, and GIoU by their definitions respectively. The SO has the following properties:

- Similar to IoU and GIoU, SO is invariant to the scale of the problem. Because the loss is normalized by the scale of box.
- SO is always a lower bound for $IoU + 1$ and $GIoU + 1$, and this lower bound becomes tighter when A and B have a stronger shape similarity.
- $\forall A, B \in \mathcal{R}^4, 0 \leq IoU(A, B) \leq 1$, SO and GIoU have a symmetric range, $\forall A, B \in \mathcal{R}^4, -1 \leq GIoU(A, B) \leq 1$, $-2 \leq SO(A, B) \leq 2$.
- 1) Similar to IoU and GIoU, the max value occurs when two objects match perfectly, i.e. if $|A \cup B| = |A \cap B|$, then $SO = 2$, $IoU = GIoU = 1$.
- 2) SO value asymptotically converges to -2 when two bounding boxes are far away.
- Different from IoU, SO still has gradient for non-overlaps cases and it has a larger gradient than GIoU.

We also demonstrate this correlation qualitatively in Fig. 4 by taking over samples from 20K random samples from coordinates of two 2D rectangles. It shows that SO has a strong correlation with IoU and GIoU in high IoU values. However, in the case of low overlap, SO can make the bounding box change position and shape faster compared with IoU and GIoU. Thus, SO is promising to have a larger gradient in these cases. In conclusion, optimizing SO loss can be a better choice than optimizing IoU and GIoU loss.

C. Simulation experiment

To better understand the efficiency of our \mathcal{L}_{SCA} , we also provide a simple simulation experiment to compare \mathcal{L}_{IoU} , \mathcal{L}_{GIoU} , and \mathcal{L}_{DIoU} . In the simulation experiment, we try to

Algorithm 2 Simulation experiment

Input: $S = \{B_i^p\}_{i=0}^P$ is the set of pre-defined anchor boxes. $S^G = \{B_j^g\}_{j=0}^G$ is the set of different scale and ratio target boxes. A loss function \mathcal{L} for calculating loss of predicted boxes and ground truth boxes.

Output: Mean regression error of every anchor with its corresponding target box.

```

1: Initialize  $IoU = 0$ , learning rate  $\eta$  and maximum iteration  $T$ 
2: Do bounding box regression.
3: for  $i = 1$  to  $G$  do
4:   for  $j = 1$  to  $P$  do
5:     for  $t = 1$  to  $T$  do
6:        $Loss = \mathcal{L}(B_{i,j,t}^p, B_i^g)$ 
7:        $\nabla B_{i,j,t}^p = \frac{\partial Loss}{\partial B_{i,j,t}^p}$ 
8:        $B_{i,j,t}^p = B_{i,j,t-1}^p + \eta \nabla B_{i,j,t}^p$ 
9:       Calculating the bounding box regression error:
          RegressionError $_t = |B_i^p - B_j^g|$ 
10:      end for
11:    end for
12:  end for
13: MeanRegressionError = mean(RegressionError $_t$ )
14: return MeanRegressionError

```

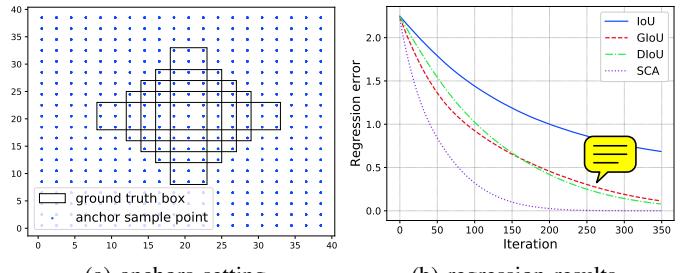


Fig. 5: (a) 3600 regression anchor boxes are adopted by considering different scales and aspect ratios, (b) regression results curves with iteration t of different bounding box regression losses.

enumerate all possible anchor boxes. In particular, we choose 5 specific boxes with different aspect ratios (e.g. 4:1, 2:1, 1:1, 1:2, 1:4) as ground truth boxes. Then anchor boxes are uniformly sampled in 20×20 grid with the ratio of (2:1, 1:1, 1:2) and scale of (5, 10, 15) and thus we have 3600 anchors as we can see in Fig. 5a. All the anchor boxes should be regressed to each ground truth box.

Given a loss function \mathcal{L} , we can simulate the procedure of bounding box regression using gradient descent algorithm. For each predicted box, we first calculate the loss function and the gradient.

$$Loss = \mathcal{L}(B_i^t, B_j^g), \quad (5)$$

$$\nabla B_i^t = \frac{\partial Loss}{\partial B_i^t}, \quad (6)$$

Then, the update process can be obtained by:

$$B_i^t = B_i^{t-1} + \eta \nabla B_i^t. \quad (7)$$

where η is the learning rate, B_i^t is the predicted box at iteration t and ∇B_i^t is the gradient of the corresponding box. The regression error is:

$$\text{RegressionError} = |B_i^t - B_j^g|, \quad (8)$$

Finally, we calculate the mean regression error to evaluate the regression accuracy with different loss functions. The final regression result has shown in Fig. 5.

As shown in Figure. 5b, SCALoss can converge faster as it produces large gradient for low overlapping boxes as we discuss in Sec. I. SCALoss also has smaller regression error than other loss functions. This demonstrates SCALoss is a stricter similarity measure, and evaluating the similarity of corners and sides is more efficient in accurating bounding box regression.

IV. EXPERIMENT

In this section, we construct experiments to evaluate the performance of our SCALoss by incorporating it into the most popular object detectors such as YOLOV3, SSD, Reppoints, Faster R-CNN. To this end, we replace their default regression losses with \mathcal{L}_{SCA} , *i.e.* we replace ℓ_n loss in YOLOV3 / SSD / Reppoints / Faster R-CNN. We compare \mathcal{L}_{SCA} against \mathcal{L}_{IoU} , \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , and \mathcal{L}_{CIoU} . We use the mmdetection [33] toolbox to conduct all our experiments except YOLOV3. We use Pytorch [34] with 4 NVIDIA 1080Ti in Ubuntu. All models are pre-trained on ImageNet [35].

The ratio of classification and regression loss would influence the results. For all detectors, we select the best weight ratios according to their performances on the validation set and report it on the test set as other detector methods.

A. Dataset

All results are reported on two popular object detection benchmarks, the PASCAL VOC challenge and the Microsoft Common Objects in Context detection (MS-COCO) challenges. **PASCAL VOC:** The Pascal Visual Object Classes (VOC) dataset is one of the most popular benchmarks for category classification, detection, and semantic segmentation. For object detection, it has 20 pre-defined classes with annotated bounding boxes. We use PASCAL VOC 2007 + 2012 (the union of VOC 2007 and VOC 2012 trainval) with 16551 images as the training set and PASCAL VOC 2007 test with 4952 images as the test set.

MS COCO: Microsoft Common Objects in Context (MS-COCO) is another popular dataset for object detection, instance segmentation, and object keypoint detection. It is a large scale dataset with 80 pre-defined classes. We use COCO *train2017* with 135k images as the training set, *val2017* with 5k images as the validation set and *test-dev* with 20k images as the test set.

B. Evaluation

In this paper, we adopt the same mAP calculation method as MS COCO to report all our results. The mAP score is calculated by taking mean AP over all classes and over all 10

IoU thresholds, *i.e.* IoU= 0.5, 0.55, ..., 0.95. While PASCAL VOC only considers one IoU, *i.e.*, IoU = 0.5, we modify it same as COCO for better performance comparison. Since the COCO server only returns the result with IoU is 0.5 or 0.75, we only report AP50, AP75, and mAP on COCO *test-dev* set.

C. YOLOV3 and SSD

We first use two representative one-stage detectors, *i.e.*, YOLOV3 and SSD, to construct experiments.

1) **YOLOV3 on COCO:** Following its training protocol¹, we train YOLOV3 with every aforementioned bounding box regression loss on the training set for 273 epochs. The backbone network is Darknet53 and the input image size is 416×416 . The performance of each loss has been shown in Table I. We also compared them on the MS COCO test dev set by submitting the results to the COCO server. All results are shown in Table II. The result shows that training YOLOV3 with \mathcal{L}_{SCA} can considerably improve its performance compared to its \mathcal{L}_{IoU} , \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , and \mathcal{L}_{CIoU} . Note that the improvement mostly comes from high overlap metrics, like AP75. Our method promotes the gradient for low overlapping cases, so the AP50 is much better than \mathcal{L}_{IoU} , \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , and \mathcal{L}_{CIoU} .

TABLE I: Comparison between the performance of YOLOV3 trained using \mathcal{L}_{IoU} , \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , \mathcal{L}_{CIoU} , and \mathcal{L}_{SCA} losses on COCO 2017 *val* set.

Loss	mAP	AP ₅₀	AP ₆₅	AP ₇₅	AP ₈₀	AP ₉₀
\mathcal{L}_{IoU}	34.5	54.3	45.7	36.2	29.7	12.2
\mathcal{L}_{GIoU}	34.7	55	45.8	36.2	29.6	12.5
relative improv.(%)	0.58%	1.29%	0.22%	0%	-0.34%	2.46%
\mathcal{L}_{DIoU}	34.7	54.7	46	36.4	29.6	12.5
relative improv.(%)	0.58%	0.74%	0.66%	0.55%	-0.34%	2.46%
\mathcal{L}_{CIoU}	34.8	54	46.1	36.3	29.7	12.6
relative improv.(%)	0.87%	-0.55%	0.88%	0.28%	0%	3.28%
\mathcal{L}_{SCA}	35.2	55.6	46.5	36.8	30.4	12.8
relative improv.(%)	2.03%	2.39%	1.75%	1.66%	2.36%	4.92%

TABLE II: Comparison between the performance of YOLOV3 trained using \mathcal{L}_{IoU} , \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , \mathcal{L}_{CIoU} , and \mathcal{L}_{SCA} losses on COCO 2017 *test* set.

Loss	mAP	AP ₅₀	AP ₇₅
\mathcal{L}_{IoU}	34.5	54.6	36.4
\mathcal{L}_{GIoU}	34.7	55	36.6
relative improv.(%)	0.58%	0.73%	0.55%
\mathcal{L}_{DIoU}	34.8	55	36.6
relative improv.(%)	0.87%	0.73%	0.55%
\mathcal{L}_{CIoU}	34.8	54.9	36.8
relative improv.(%)	0.87%	0.55%	1.1%
\mathcal{L}_{SCA}	35.3	55.3	37.4
relative improv.(%)	2.32%	1.28%	2.75%

¹<https://github.com/ultralytics/yolov3>

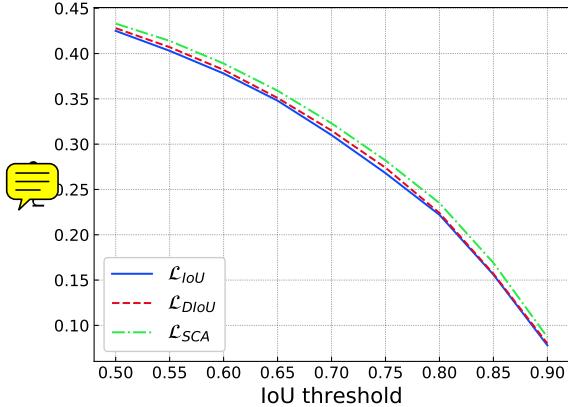


Fig. 6: IoU threshold against mAP for SSD using \mathcal{L}_{IoU} , \mathcal{L}_{DIoU} , and \mathcal{L}_{SCA} on COCO *val* set.

2) *SSD on COCO*: We train SSD using the aforementioned bounding box regression losses on the training set for 120 epochs. The backbone network is VGG16 [36] and the input image size is 300×300 . The results are shown in III. We also compare the results on the MS COCO test set in IV. SCA can improve the baseline (\mathcal{L}_{IoU}) with more than 1AP. However, \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , \mathcal{L}_{CIoU} can slightly improve the baseline. The improvement is inferior compared to \mathcal{L}_{SCA} . We also plot the relationship between IoU and mAP, as we can see in Fig. 6, our SCALoss can get consistent higher AP for different IoU threshold.

3) *SSD on PASCAL VOC*: When training SSD on PASCAL VOC, we use the same setting as the COCO. The training epochs is 72. The performance for each loss has been shown in Table V. The result shows that training SSD with \mathcal{L}_{SCA} can considerably improve its performance compared to \mathcal{L}_{IoU} . Moreover, \mathcal{L}_{SCA} can get better performance than \mathcal{L}_{GIoU} , and \mathcal{L}_{DIoU} .

TABLE III: Comparison between the performance of SSD trained using \mathcal{L}_{IoU} , \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , \mathcal{L}_{CIoU} , and \mathcal{L}_{SCA} losses on COCO 2017 *val* set.

Loss	mAP	AP ₅₀	AP ₆₅	AP ₇₅	AP ₈₀	AP ₉₀
\mathcal{L}_{IoU}	26	42.5	34.8	26.8	22.2	7.8
\mathcal{L}_{GIoU}	26.1	42.7	34.9	27.1	22.2	7.8
relative improv.(%)	0.38%	0.47%	0.29%	1.12%	0%	0%
\mathcal{L}_{DIoU}	26.3	42.8	35.1	27.4	22.4	8
relative improv.(%)	1.15%	0.71%	0.86%	2.24%	0.9%	2.56%
\mathcal{L}_{CIoU}	26.3	43	35.2	27.4	22.4	7.7
relative improv.(%)	1.15%	1.18%	1.15%	2.24%	0.9%	-1.28%
\mathcal{L}_{SCA}	27.1	43.3	35.9	28.2	23.5	8.7
relative improv.(%)	4.23%	1.88%	3.16%	5.22%	5.86%	11.54%

D. Reppoints

We also train Reppoints to verify our method, which is an anchor-free detector. We use ResNet50-FPN [37], [38] as our backbone. The default bounding box regression is ℓ_1 -smooth loss. We also replace it with the aforementioned losses.

TABLE IV: Comparison between the performance of SSD trained using \mathcal{L}_{IoU} , \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , \mathcal{L}_{CIoU} , and \mathcal{L}_{SCA} losses on COCO 2017 *test* set.

Loss	mAP	AP ₅₀	AP ₇₅
\mathcal{L}_{IoU}	26.2	42.8	26.9
\mathcal{L}_{GIoU}	26.3	43.1	27.3
relative improv.(%)	0.38%	0.7%	1.49%
\mathcal{L}_{DIoU}	26.4	43.2	27.6
relative improv.(%)	0.76%	0.93%	2.6%
\mathcal{L}_{CIoU}	26.4	43.1	27.5
relative improv.(%)	0.76%	0.7%	2.23%
\mathcal{L}_{SCA}	27.3	43.7	28.4
relative improv.(%)	4.2%	2.1%	5.58%

TABLE V: Comparison between the performance of SSD trained using \mathcal{L}_{IoU} , \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , \mathcal{L}_{CIoU} , and \mathcal{L}_{SCA} losses on the PASCAL VOC 2007 *test* set.

Loss	mAP	AP ₅₀	AP ₆₅	AP ₇₅	AP ₈₀	AP ₉₀
\mathcal{L}_{IoU}	52.28	78.26	68.51	56.22	46.9	20.2
\mathcal{L}_{GIoU}	52.5	78.57	69.11	56.71	46.87	19.88
relative improv.(%)	0.42%	0.4%	0.88%	0.87%	-0.06%	-1.58%
\mathcal{L}_{DIoU}	52.65	78.64	69.11	56.55	47.9	20.09
relative improv.(%)	0.71%	0.49%	0.88%	0.59%	2.13%	-0.54%
\mathcal{L}_{CIoU}	52.75	78.76	69.03	56.45	48.56	20.69
relative improv.(%)	0.9%	0.64%	0.76%	0.41%	3.54%	2.43%
\mathcal{L}_{SCA}	53.23	79.05	69.29	57.01	48.91	21.56
relative improv.(%)	1.82%	1.01%	1.14%	1.41%	4.29%	6.73%

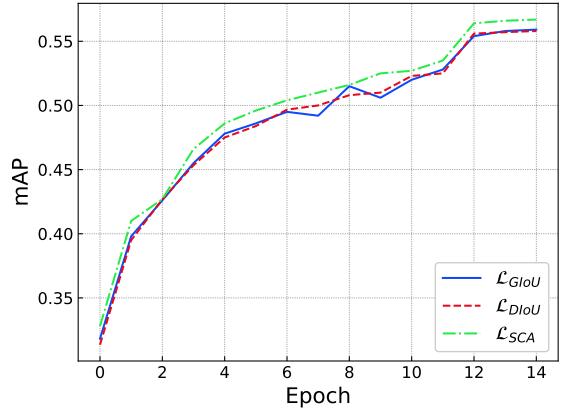


Fig. 7: mAP value against train epochs for Reppoints using \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , and \mathcal{L}_{SCA} on PASCAL VOC 2007 *test* set.

1) *Reppoints on PASCAL VOC*: We train Reppoints on PASCAL VOC dataset. The size of the input image is 1000×600 . We train the network with every different loss on the training set for 18 epochs. The performance for each loss has been evaluated using the PASCAL VOC 2007 test and the results have been reported in Table VI. We get decent improvement (near 0.9AP) with IoU. Our SCALoss improves performance with gains of near 0.8 AP than DIoU and GIoU loss.

We also plot the relationship between training epoch and mAP, as we can see in Fig. 7, our SCALoss can get consistent higher AP during the training process.

TABLE VI: Comparison between the performance of Repoints trained using \mathcal{L}_{IoU} , \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , \mathcal{L}_{CIoU} , and \mathcal{L}_{SCA} losses on the PASCAL VOC 2007 test set.

Loss	mAP	AP ₅₀	AP ₆₅	AP ₇₅	AP ₈₀	AP ₉₀
\mathcal{L}_{IoU}	55.83	80.36	72.23	59.89	51.83	25.61
\mathcal{L}_{GIoU} relative improv.(%)	55.91 0.14%	80.88 0.65%	72.32 0.12%	60.07 0.3%	51.66 -0.33%	24.66 -3.71%
\mathcal{L}_{DIoU} relative improv.(%)	55.89 0.11%	80.88 0.65%	72.67 0.61%	60.26 0.62%	51.69 -0.27%	24.53 -4.22%
\mathcal{L}_{CIoU} relative improv.(%)	56.01 0.32%	80.69 0.41%	72.67 0.61%	60.41 0.87%	51.96 0.25%	25.8 0.74%
\mathcal{L}_{SCA} relative improv.(%)	56.69 1.54%	81.23 1.08%	73.19 1.33%	60.8 1.52%	53.17 2.59%	26.29 2.66%

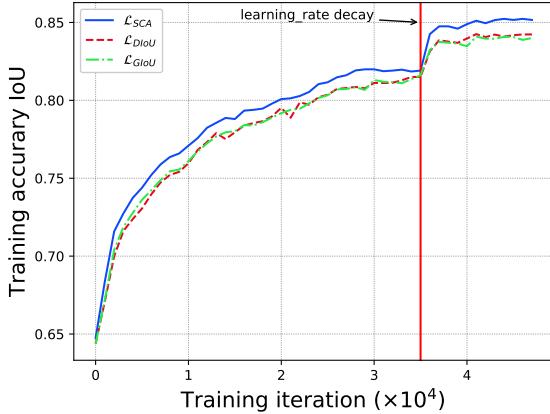


Fig. 8: Training accuracy (IoU) against train iteration for Faster R-CNN using \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , and \mathcal{L}_{SCA} on PASCAL VOC 2007 test set.

E. Faster R-CNN

Faster R-CNN is a two-stage detector, which generates object proposals for the second stage to classify and refine bounding boxes. We use the ResNet50-FPN backbone network. We replace the ℓ_1 -smooth loss in the second stage in Faster R-CNN. We multiply regression loss by a factor of 12 for all experiments.

1) *Faster R-CNN on PASCAL VOC:* We train Faster R-CNN for 12 epochs on PASCAL VOC dataset, and the input image is resized to 1000×600 . The final results have been reported in Table VII. The results show that training Faster-RCNN using our SCALoss can consistently improve its performance compared to IoU loss (near 1.4%). SCALoss can improve the performance with gains of near 0.7 AP / 0.5 AP than GIoU loss, CIoU loss respectively.

To verify the conclusion in our simulation experiment, we plot the relationship of training accuracy (IoU) and training iteration for faster R-CNN as show in Fig. 8. It's obviously that SCA can converge faster than other losses and achieve higher training accuracy.

2) *Faster R-CNN on MS-COCO:* Similarly, we train Faster R-CNN on the MS-COCO dataset for 12 epochs. We resize the input images to 1333×800 . The results on the validation set of MS COCO have been reported in Table VIII. We also compared them on the MS COCO test dev set by submitting the results to the COCO server. All results are shown in

TABLE VII: Comparison between the performance of Faster-RCNN trained using \mathcal{L}_{IoU} , \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , \mathcal{L}_{CIoU} , and \mathcal{L}_{SCA} losses on the PASCAL VOC 2007 test set.

Loss	mAP	AP ₅₀	AP ₆₅	AP ₇₅	AP ₈₀	AP ₉₀
\mathcal{L}_{IoU}	50.85	79.6	69.85	55.14	43.23	13.01
\mathcal{L}_{GIoU} relative improv.(%)	50.9 0.1%	79.69 0.11%	70.56 1.02%	55 -0.25%	43.34 0.25%	12.7 -2.38%
\mathcal{L}_{DIoU} relative improv.(%)	50.86 0.02%	79.99 0.49%	70.48 0.9%	54.56 -1.05%	42.79 -1.02%	12.8 -1.61%
\mathcal{L}_{CIoU} relative improv.(%)	51.08 0.45%	79.52 -0.1%	70.07 0.31%	55.12 -0.04%	44.04 1.87%	13.1 0.69%
\mathcal{L}_{SCA} relative improv.(%)	51.56 1.4%	80.11 0.64%	70.95 1.57%	56.14 1.81%	44.19 2.22%	13.49 3.69%

Table IX. Similar to the PASCAL VOC experiment, the results show a consistent improvement in the performance of Faster R-CNN when it is trained using \mathcal{L}_{SCA} as bounding box regression loss. Since Faster R-CNN is a two-stage detector, the proposal bounding boxes are more dense than one-stage detectors with high overlapping with its ground truth, so the amount of improvement between different losses is less than previous experiments.

TABLE VIII: Comparison between the performance of Faster-RCNN trained using \mathcal{L}_{IoU} , \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , \mathcal{L}_{CIoU} , and \mathcal{L}_{SCA} losses on the COCO 2017 val set.

Loss	mAP	AP ₅₀	AP ₆₅	AP ₇₅	AP ₈₀	AP ₉₀
\mathcal{L}_{IoU}	37.5	58	50.2	40.7	33.5	11.9
\mathcal{L}_{GIoU} relative improv.(%)	37.7 0.53%	57.9 -0.17%	50.2 0%	40.6 -0.25%	34 1.49%	12.5 5.04%
\mathcal{L}_{DIoU} relative improv.(%)	37.6 0.27%	57.9 -0.17%	50.3 0.2%	40.4 -0.74%	34 1.49%	12.1 1.68%
\mathcal{L}_{CIoU} relative improv.(%)	37.7 0.53%	57.8 -0.34%	50.3 0.2%	40.5 -0.49%	33.9 1.19%	12.2 2.52%
\mathcal{L}_{SCA} relative improv.(%)	37.8 0.8%	57.9 -0.17%	50.4 0.4%	41 0.74%	34 1.49%	12.5 5.04%

TABLE IX: Comparison between the performance of Faster R-CNN trained using \mathcal{L}_{IoU} , \mathcal{L}_{GIoU} , \mathcal{L}_{DIoU} , \mathcal{L}_{CIoU} and \mathcal{L}_{SCA} losses on COCO 2017 test set.

Loss	mAP	AP ₅₀	AP ₇₅
\mathcal{L}_{IoU}	37.7	58.4	40.7
\mathcal{L}_{GIoU} relative improv.(%)	37.8 0.27%	58.4 0%	40.7 0%
\mathcal{L}_{DIoU} relative improv.(%)	37.7 0%	58.6 0.34%	40.5 -0.49%
\mathcal{L}_{CIoU} relative improv.(%)	37.8 0.27%	58.5 0.17%	40.6 -0.25%
\mathcal{L}_{SCA} relative improv.(%)	38.1 1.06%	58.6 0.34%	41.1 0.98%

3) *Compare with other state-of-the-art bounding box regression losses:* In this subsection, we conduct experiments to compare other state-of-the-art bounding box regression losses. We train Faster R-CNN on the MS-COCO dataset for 24 epochs. The backbone network is ResNet50-FPN. The input image is resized to 1333×800 . The baseline loss function

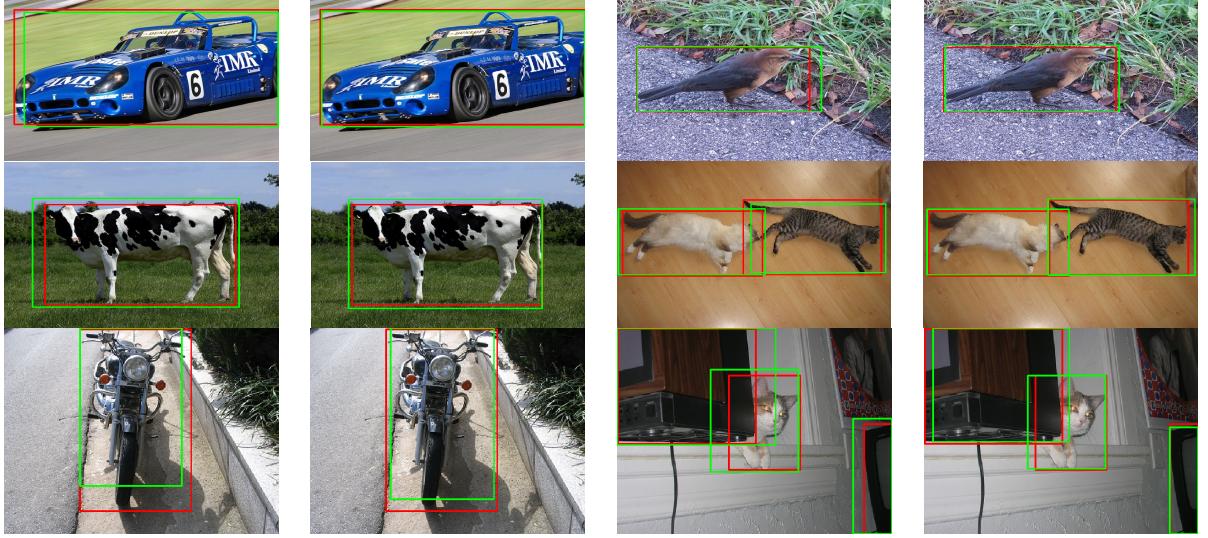


Fig. 9: Examples results from PASCAL VOC dataset using Faster R-CNN trained with \mathcal{L}_{GIoU} and \mathcal{L}_{SCA} (left to right). Red: the ground truth box, green: the predicted box.

is ℓ_1 -smooth. In addition, we replace it with Bounded IoU Loss [31], Balanced 11 loss [30], KL loss [32], and SCA loss. We submit the results to COCO server and the results are shown in X. Note that KL loss [32] adds another branch to estimate standard deviation along with box location. This will increase the inference time. We also add soft NMS [39] to fairly compare with KL loss. Our method outperforms KL loss (with var voting [32] and soft-NMS [39]) with the gain of 0.4 AP even if we do not add var voting. Overall, our SCALoss achieves better performance than other losses. Specifically, +1.2AP and +0.6AP improvement compare with ℓ_1 -smooth and \mathcal{L}_{KL} respectively.

TABLE X: Comparison between the performance of Faster-RCNN trained using its own loss (ℓ_1 -smooth), $\mathcal{L}_{BoundedIoU}$, $\mathcal{L}_{BalancedL1}$, \mathcal{L}_{KL} , and \mathcal{L}_{SCA} losses on the COCO 2017 test set. We also add soft-NMS for faily comparision.

Loss / Evaluation	mAP	AP ₅₀	AP ₇₅
ℓ_1 -smooth	37.9	59.2	41.1
$\mathcal{L}_{BoundedIoU}$	38.3	58.4	41.5
$\mathcal{L}_{BalancedL1}$	38.3	58.6	41.5
\mathcal{L}_{KL}	38.5	57.8	41.2
\mathcal{L}_{SCA}	39.1	59.3	42.3
$\mathcal{L}_{BoundedIoU}$ + soft-NMS	38.8	58.4	42.6
$\mathcal{L}_{BalancedL1}$ + soft-NMS	38.9	58.7	42.6
\mathcal{L}_{KL} + var voting	38.8	57.8	41.6
\mathcal{L}_{KL} + var voting + soft-NMS	39.2	57.6	42.5
\mathcal{L}_{SCA} + soft-NMS	39.6	59.4	43.3

F. Ablation Study of Weight factor α

In this section, we study the weight factor α for \mathcal{L}_{SO} and \mathcal{L}_{CD} in Eq.(4). We use Reppoints detection framework with ResNet50-FPN as the backbone and PASCAL VOC dataset. The input image is resized to 1000×1000 . We replace the ℓ_1 -smooth with our \mathcal{L}_{SCA} and use different $\alpha = \{0.0, 0.2, \dots, 0.8\}$ to conduct experiment. The loss weight is 3.0 for all settings. The results have been shown in Table XI.

TABLE XI: Ablation study for different weight factor α for Reppoints trained using \mathcal{L}_{SCA} on the PASCAL VOC 2007 test set.

α / Evaluation	mAP	AP ₅₀	AP ₆₅	AP ₇₅	AP ₈₀	AP ₉₀
0.0	56.43	81.05	72.78	60.93	52.73	25.16
0.2	56.69	81.23	73.19	60.80	53.17	26.29
0.4	56.28	80.79	72.83	60.78	52.15	25.51
0.6	56.59	80.62	72.62	60.88	52.85	26.34
0.8	56.05	79.88	72.44	60.85	52.06	25.69

For different models and datasets, we can make efforts to search an optimal α for better performance. However, for simplicity and saving computational resources, we choose $\alpha = 0.2$ for all our settings.

V. CONCLUSIONS

In this paper, we propose Side and Corner Aligned Loss (SCALoss) for bounding box regression. SCALoss consists of Side Overlap and Corner Distance, which takes bounding box side and corner points into account. Combine the advantage of these two parts, SCALoss not only produces more penalty for low overlapping boxes and focuses more on hard samples but also speeds up the model convergence. In a result, SCALoss can serve as a more comprehensive measure than ℓ_n loss and IoU-based loss. Experiments on COCO and PASCAL VOC benchmarks show that SCALoss can bring consistent improvement and outperform ℓ_n loss and IoU based loss with popular object detectors, such as YOLOV3, SSD, Faster R-CNN, and Reppoints.

In the future, we plan to investigate the feasibility of deriving an extension for SCALoss in the case of 3D object detection. This extension is promising to improve the performance of 3D object detection.

REFERENCES

- [1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
- [2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.
- [3] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
- [4] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "Reppoints: Point set representation for object detection," *arXiv preprint arXiv:1904.11490*, 2019.
- [5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
- [6] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [7] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network," in *Proceedings of the 24th ACM international conference on Multimedia*. ACM, 2016, pp. 516–520.
- [8] H. Rezatofighi, N. Tsai, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 658–666.
- [9] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," *arXiv preprint arXiv:1911.08287*, 2019.
- [10] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.
- [12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [13] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.
- [14] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.
- [15] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.
- [16] A. Shrivastava and A. Gupta, "Contextual priming and feedback for faster r-cnn," in *European conference on computer vision*. Springer, 2016, pp. 330–348.
- [17] Y. Li, Y. Chen, N. Wang, and Z. Zhang, "Scale-aware trident networks for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6054–6063.
- [18] X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan, "Grid r-cnn," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7363–7372.
- [19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [20] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.
- [21] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [22] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," *arXiv preprint arXiv:2011.08036*, 2020.
- [23] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.
- [24] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "Ron: Reverse connection with objectness prior networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5936–5944.
- [25] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4203–4212.
- [26] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734–750.
- [27] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6569–6578.
- [28] X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 850–859.
- [29] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9308–9316.
- [30] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra r-cnn: Towards balanced learning for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 821–830.
- [31] L. Tychsen-Smith and L. Petersson, "Improving object localization with fitness nms and bounded iou loss," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6877–6885.
- [32] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2888–2897.
- [33] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu et al., "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.
- [34] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.
- [35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [38] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [39] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms-improving object detection with one line of code," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5561–5569.