Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression

Zhaohui Zheng,[1] **Ping Wang,**[1] **Wei Liu,**[2] **Jinze Li,**[3] **Rongguang Ye,**[1] **Dongwei Ren**[*2]

[1]School of Mathematics, Tianjin University, China
[2]College of Intelligence and Computing, Tianjin University, China
[3]School of Information Technology and Cyber Security, People's Public Security University of China

*Corresponding author: rendongweihit@gmail.com

Abstract

Bounding box regression is the crucial step in object detection. In existing methods, while ℓ_n-norm loss is widely adopted for bounding box regression, it is not tailored to the evaluation metric, i.e., Intersection over Union (IoU). Recently, IoU loss and generalized IoU (GIoU) loss have been proposed to benefit the IoU metric, but still suffer from the problems of slow convergence and inaccurate regression. In this paper, we propose a Distance-IoU (DIoU) loss by incorporating the normalized distance between the predicted box and the target box, which converges much faster in training than IoU and GIoU losses. Furthermore, this paper summarizes three geometric factors in bounding box regression, i.e., overlap area, central point distance and aspect ratio, based on which a Complete IoU (CIoU) loss is proposed, thereby leading to faster convergence and better performance. By incorporating DIoU and CIoU losses into state-of-the-art object detection algorithms, e.g., YOLO v3, SSD and Faster R-CNN, we achieve notable performance gains in terms of not only IoU metric but also GIoU metric. Moreover, DIoU can be easily adopted into non-maximum suppression (NMS) to act as the criterion, further boosting performance improvement. The source code and trained models are available at https://github.com/Zzh-tju/DIoU.

Object detection is one of the key issues in computer vision tasks, and has received considerable research attention for decades (Redmon et al. 2016; Redmon and Farhadi 2018; Ren et al. 2015; He et al. 2017; Yang et al. 2018; Wang et al. 2019; 2018). Generally, existing object detection methods can be categorized as: one-stage detection, such as YOLO series (Redmon et al. 2016; Redmon and Farhadi 2017; 2018) and SSD (Liu et al. 2016; Fu et al. 2017), two-stage detection, such as R-CNN series (Girshick et al. 2014; Girshick 2015; Ren et al. 2015; He et al. 2017), and even multi-stage detection, such as Cascade R-CNN (Cai and Vasconcelos 2018). Despite of these different detection frameworks, bounding box regression is the crucial step to predict a rectangular box to locate the target object.

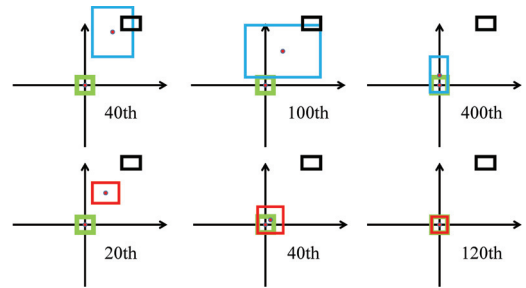In terms of evaluation metric for bounding box regression,

Figure 1: Bounding box regression steps by GIoU loss (first row) and DIoU loss (second row). Green and black denote target box and anchor box, respectively. Blue and red denote predicted boxes for GIoU loss and DIoU loss, respectively. GIoU loss generally increases the size of predicted box to overlap with target box, while DIoU loss directly minimizes normalized distance of central points.

Intersection over Union (IoU) is the most popular metric,

$$IoU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|}, \qquad (1)$$

where $B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$ is the ground-truth, and $B = (x, y, w, h)$ is the predicted box. Conventionally, ℓ_n-norm (e.g., $n = 1$ or 2) loss is adopted on the coordinates of B and B^{gt} to measure the distance between bounding boxes (Redmon et al. 2016; Girshick 2015; Ren et al. 2015; He et al. 2017; Bae 2019). However, as suggested in (Yu et al. 2016; Rezatofighi et al. 2019), ℓ_n-norm loss is not a suitable choice to obtain the optimal IoU metric. In (Rezatofighi et al. 2019), IoU loss is suggested to be adopted for improving the IoU metric,

$$\mathcal{L}_{IoU} = 1 - \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|}. \qquad (2)$$

However, IoU loss only works when the bounding boxes have overlap, and would not provide any moving gradient for non-overlapping cases. And then generalized IoU loss (GIoU) (Rezatofighi et al. 2019) is proposed by adding a

$$\mathcal{L}_{IoU} = 0.75 \qquad \mathcal{L}_{IoU} = 0.75 \qquad \mathcal{L}_{IoU} = 0.75$$
$$\mathcal{L}_{GIoU} = 0.75 \qquad \mathcal{L}_{GIoU} = 0.75 \qquad \mathcal{L}_{GIoU} = 0.75$$
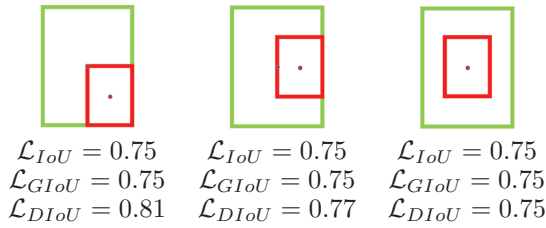$$\mathcal{L}_{DIoU} = 0.81 \qquad \mathcal{L}_{DIoU} = 0.77 \qquad \mathcal{L}_{DIoU} = 0.75$$

Figure 2: GIoU loss degrades to IoU loss for these cases, while our DIoU loss is still distinguishable. Green and red denote target box and predicted box respectively.

penalty term,

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}, \qquad (3)$$

where C is the smallest box covering B and B^{gt}. Due to the introduction of penalty term, the predicted box will move towards the target box in non-overlapping cases.

Although GIoU can relieve the gradient vanishing problem for non-overlapping cases, it still has several limitations. By a simulation experiment (see *Sec. **Analysis to IoU and GIoU Losses** for details*), we can evaluate the performance of GIoU loss for various bounding box positions. As shown in Fig. 1, one can see that GIoU loss intends to increase the size of predicted box at first, making it have overlap with target box, and then the IoU term in Eqn. (3) will work to maximize the overlap area of bounding box. And from Fig. 2, GIoU loss will totally degrade to IoU loss for enclosing bounding boxes. Due to heavily relying on the IoU term, GIoU empirically needs more iterations to converge, especially for horizontal and vertical bounding boxes (see Fig. 4). Usually GIoU loss cannot well converge in the state-of-the-art detection algorithms, yielding inaccurate detection.

In this paper, we propose a Distance-IoU (DIoU) loss for bounding box regression. In particular, we simply add a penalty term on IoU loss to directly minimize the normalized distance between central points of two bounding boxes, leading to much faster convergence than GIoU loss. From Fig. 1, DIoU loss can be deployed to directly minimize the distance between two bounding boxes. And with only 120 iterations, the predicted box matches with the target box perfectly, while GIoU does not converge even after 400 iterations. Furthermore, we suggest that a good loss for bounding box regression should consider three important geometric measures, i.e., overlap area, central point distance and aspect ratio, which have been ignored for a long time. By combining these geometric measures, we further propose a Complete IoU (CIoU) loss for bounding box regression, leading to faster convergence and better performance than IoU and GIoU losses. The proposed losses can be easily incorporated into the state-of-the-art object detection algorithms. Moreover, DIoU can be employed as a criterion in non-maximum suppression (NMS), by which not only the overlap area but also the distance between central points of two bounding boxes are considered when suppressing redundant boxes, making it more robust for the cases with occlusions.

To evaluate our proposed methods, DIoU loss and CIoU loss are incorporated into several state-of-the-art detection algorithms including YOLO v3 (Redmon and Farhadi 2018), SSD (Liu et al. 2016) and Faster R-CNN (Ren et al. 2015), and are evaluated on two popular benchmark datasets PAS-CAL VOC 2007 (Everingham et al. 2010) and MS COCO 2017 (Lin et al. 2014).

The contribution of work is summarized as follows:

1. A Distance-IoU loss, i.e., DIoU loss, is proposed for bounding box regression, which has faster convergence than IoU and GIoU losses.

2. A Complete IoU loss, i.e., CIoU loss, is further proposed by considering three geometric measures, i.e., overlap area, central point distance and aspect ratio, which better describes the regression of rectangular boxes.

3. DIoU is deployed in NMS, and is more robust than original NMS for suppressing redundant boxes.

4. The proposed methods can be easily incorporated into the state-of-the-art detection algorithms, achieving notable performance gains.

Related Work

In this section, we briefly survey relevant works including object detection methods, loss function for bounding box regression and non-maximum suppression.

Object Detection

In (Song et al. 2018), the central axis line is applied in pedestrian detection. CornerNet (Law and Deng 2018) suggested predicting a pair of corners to replace a rectangular box for locating object. In RepPoints (Yang et al. 2019), a rectangular box is formed by predicting several points. Recently, FSAF (Zhu, He, and Savvides 2019) proposed anchor-free branch to tackle the issues of non-optimality in online feature selection. There are also several loss functions for object detection, e.g., focal loss (Lin et al. 2017), class-balanced loss (Cui et al. 2019), balanced loss for classification and bounding box regression (Pang et al. 2019), and gradient flow balancing loss (Li, Liu, and Wang 2019). Nevertheless, the regression of rectangular boxes is still the most popular manner in the state-of-the-art object detection algorithms (Redmon and Farhadi 2018; He et al. 2017; Fu et al. 2017; Liu et al. 2016; Tian et al. 2019).

Loss Function for Bounding Box Regression

The ℓ_n-norm loss functions are usually adopted in bounding box regression, but are sensitive to variant scales. In YOLO v1 (Redmon et al. 2016), square roots for w and h are adopted to mitigate this effect, while YOLO v3 (Redmon and Farhadi 2018) uses $2 - wh$. IoU loss is also used since Unitbox (Yu et al. 2016), which is invariant to the scale. GIoU (Rezatofighi et al. 2019) loss is proposed to tackle the issues of gradient vanishing for non-overlapping cases, but is still facing the problems of slow convergence and inaccurate regression. In comparison, we propose DIoU and CIoU losses with faster convergence and better regression accuracy.

Algorithm 1 Simulation Experiment

Require: Loss $\mathcal{L}$ is a continuous bounded function defined on $\mathbb{R}_+^4$.
$\mathbb{M} = \{\{B_{n,s}\}_{s=1}^S\}_{n=1}^N$ is the set of anchor boxes at $N = 5,000$ uniformly scattered points within the circular region with center $(10,10)$ and radius 3, and $S = 7 \times 7$ covers 7 scales and 7 aspect ratios of anchor boxes.
$\mathbb{M}^{gt} = \{B_i^{gt}\}_{i=1}^7$ is the set of target boxes that are fixed at $(10,10)$ with area 1, and have 7 aspect ratios.

Ensure: Regression error $\mathbf{E} \in \mathbb{R}^{T \times N}$

1: Initialize $\mathbf{E} = \mathbf{0}$ and maximum iteration T.
2: Do bounding box regression:
3: **for** $n = 1$ to N **do**
4:    **for** $s = 1$ to S **do**
5:       **for** $i = 1$ to 7 **do**
6:          **for** $t = 1$ to T **do**
7:             $\eta = \begin{cases} 0.1 & \text{if} \quad t <= 0.8T \\ 0.01 & \text{if} \quad 0.8T < t <= 0.9T \\ 0.001 & \text{if} \quad t > 0.9T \end{cases}$
8:             $\nabla B_{n,s}^{t-1}$ is gradient of $\mathcal{L}(B_{n,s}^{t-1}, B_i^{gt})$ w.r.t. $B_{n,s}^{t-1}$
9:             $B_{n,s}^t = B_{n,s}^{t-1} + \eta(2 - IoU_{n,s}^{t-1})\nabla B_{n,s}^{t-1}$
10:           $\mathbf{E}(t,n) = \mathbf{E}(t,n) + |B_{n,s}^t - B_i^{gt}|$
11:          **end for**
12:       **end for**
13:    **end for**
14: **end for**
15: **return E**

Non-Maximum Suppression

NMS is the last step in most object detection algorithms, in which redundant detection boxes are removed as long as its overlap with the highest score box exceeds a threshold. Soft-NMS (Bodla et al. 2017) penalizes the detection score of neighbors by a continuous function w.r.t. IoU, yielding softer and more robust suppression than original NMS. IoU-Net (Jiang et al. 2018) introduces a new network branch to predict the localization confidence to guide NMS. Recently, adaptive NMS (Liu, Huang, and Wang 2019) and Softer-NMS (He et al. 2019) are proposed to respectively study proper threshold and weighted average strategies. In this work, DIoU is simply deployed as the criterion in original NMS, in which overlap area and distance between two central points of bounding boxes are simultaneously considered when suppressing redundant boxes.

Analysis to IoU and GIoU Losses

To begin with, we analyze the limitations of original IoU loss and GIoU loss. However, it is very difficult to analyze the procedure of bounding box regression simply from the detection results, where the regression cases in uncontrolled benchmarks are often not comprehensive, e.g., different distances, different scales and different aspect ratios. Instead, we suggest conducting simulation experiments, where the regression cases should be comprehensively considered, and then the issues of a given loss can be easily analyzed.

Simulation Experiment

In the simulation experiments, we try to cover most of the relationships between bounding boxes in terms of distance,

scale and aspect ratio, as shown in Fig. 3(a). In particular, we choose 7 unit boxes (i.e., the area of each box is 1) with different aspect ratios (i.e., 1:4, 1:3, 1:2, 1:1, 2:1, 3:1 and 4:1) as target boxes. Without loss of generality, the central points of the 7 target boxes are fixed at $(10,10)$. The anchor boxes are uniformly scattered at 5,000 points. (*i*) Distance: In the circular region centered at $(10,10)$ with radius 3, 5,000 points are uniformly chosen to place anchor boxes with 7 scales and 7 aspect ratios. In these cases, overlapping and non-overlapping boxes are included. (*ii*) Scale: For each point, the areas of anchor boxes are set as 0.5, 0.67, 0.75, 1, 1.33, 1.5 and 2. (*iii*) Aspect ratio: For a given point and scale, 7 aspect ratios are adopted, i.e., following the same setting with target boxes (i.e., 1:4, 1:3, 1:2, 1:1, 2:1, 3:1 and 4:1). All the $5,000 \times 7 \times 7$ anchor boxes should be fitted to each target box. To sum up, there are totally $1,715,000 = 7 \times 7 \times 7 \times 5,000$ regression cases.
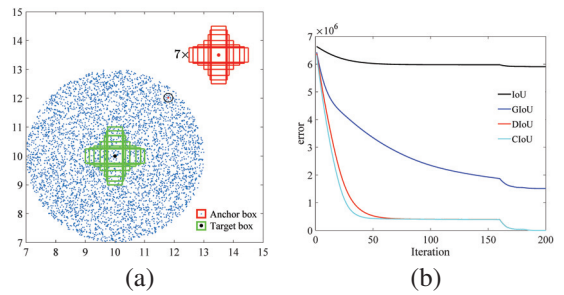

Figure 3: Simulation experiments: (a) 1,715,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e., $\sum_n \mathbf{E}(t,n)$) curves of different loss functions at iteration t.

Then given a loss function $\mathcal{L}$, we can simulate the procedure of bounding box regression for each case using gradient descent algorithm. For predicted box B_i, the current prediction can be obtained by

$$B_i^t = B_i^{t-1} + \eta(2 - IoU_i^{t-1})\nabla B_i^{t-1}, \qquad (4)$$

where B_i^t is the predicted box at iteration t, ∇B_i^{t-1} denotes the gradient of loss $\mathcal{L}$ w.r.t. B_i at iteration $t-1$, and η is the step. It is worth noting that in our implementation, the gradient is multiplied by $2 - IoU_i^{t-1}$ to accelerate the convergence. The performance of bounding box regression is evaluated using ℓ_1-norm. For each loss function, the simulation experiment is terminated when reaching iteration $T = 200$, and the error curves are shown in Fig. 3(b).

Limitations of IoU and GIoU Losses

In Fig. 4, we visualize the final regression errors at iteration T for 5,000 scattered points. From Fig. 4(a), it is easy to see that IoU loss only works for the cases of overlapping with target boxes. The anchor boxes without overlap will not move due to that ∇B is always 0.

By adding a penalty term as Eqn. (3), GIoU loss can better relieve the issues of non-overlapping cases. From Fig. 4(b), GIoU loss significantly enlarges the basin, i.e., the area that GIoU works. But the cases at horizontal and vertical

orientations are likely to still have large errors. This is because that the penalty term in GIoU loss is used to minimize $|C - A \cup B|$, but the area of $C - A \cup B$ is often small or 0 (when two boxes have inclusion relationships), and then GIoU almost degrades to IoU loss. GIoU loss would converge to good solution as long as running sufficient iterations with proper learning rates, but the convergence rate is indeed very slow. Geometrically speaking, from the regression steps as shown in Fig. 1, one can see that GIoU actually increases the predicted box size to overlap with target box, and then the IoU term will make the predicted box match with the target box, yielding a very slow convergence.

To sum up, IoU loss converges to bad solutions for non-overlapping cases, while GIoU loss is with slow convergence especially for the boxes at horizontal and vertical orientations. And when incorporating into object detection pipeline, both IoU and GIoU losses cannot guarantee the accuracy of regression. It is natural to ask that: *First*, is it feasible to directly minimize the normalized distance between predicted box and target box for achieving faster convergence? *Second*, how to make the regression more accurate and faster when having overlap even inclusion with target box?

The Proposed Method

Generally, the IoU-based loss can be defined as

$$\mathcal{L} = 1 - IoU + \mathcal{R}(B, B^{gt}), \tag{5}$$

where $\mathcal{R}(B, B^{gt})$ is the penalty term for predicted box B and target box B^{gt}. By designing proper penalty terms, in this section we propose DIoU loss and CIoU loss to answer the aforementioned two questions.

Distance-IoU Loss

To answer the *first* question, we propose to minimize the normalized distance between central points of two bounding boxes, and the penalty term can be defined as

$$\mathcal{R}_{DIoU} = \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2}, \tag{6}$$

where $\mathbf{b}$ and $\mathbf{b}^{gt}$ denote the central points of B and B^{gt}, $\rho(\cdot)$ is the Euclidean distance, and c is the diagonal length of the smallest enclosing box covering the two boxes. And then the DIoU loss function can be defined as

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2}. \tag{7}$$

As shown in Fig. 5, the penalty term of DIoU loss directly minimizes the distance between two central points, while GIoU loss aims to reduce the area of $C - B \cup B^{gt}$.

Comparison with IoU and GIoU losses   The proposed DIoU loss inherits some properties from IoU and GIoU loss.

1. DIoU loss is still invariant to the scale of regression problem.

2. Similar to GIoU loss, DIoU loss can provide moving directions for bounding boxes when non-overlapping with target box.

3. When two bounding boxes perfectly match, $\mathcal{L}_{IoU} = \mathcal{L}_{GIoU} = \mathcal{L}_{DIoU} = 0$. When two boxes are far away, $\mathcal{L}_{GIoU} = \mathcal{L}_{DIoU} \to 2$.

And DIoU loss has several merits over IoU loss and GIoU loss, which can be evaluated by simulation experiment.

1. As shown in Fig. 1 and Fig. 3, DIoU loss can directly minimize the distance of two boxes, and thus converges much faster than GIoU loss.

2. For the cases with inclusion of two boxes, or in horizontal and vertical orientations, DIoU loss can make regression very fast, while GIoU loss has almost degraded to IoU loss, i.e., $|C - A \cup B| \to 0$.

Complete IoU Loss

Then we answer the *second* question, by suggesting that a good loss for bounding box regression should consider three important geometric factors, i.e., overlap area, central point distance and aspect ratio. By uniting the coordinates, IoU loss considers the overlap area, and GIoU loss heavily relies on IoU loss. Our proposed DIoU loss aims at considering simultaneously the overlap area and central point distance of bounding boxes. However, the consistency of aspect ratios for bounding boxes is also an important geometric factor.

Therefore, based on DIoU loss, the CIoU loss is proposed by imposing the consistency of aspect ratio,

$$\mathcal{R}_{CIoU} = \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v, \tag{8}$$

where α is a positive trade-off parameter, and v measures the consistency of aspect ratio,

$$v = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2. \tag{9}$$

Then the loss function can be defined as

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v. \tag{10}$$

And the trade-off parameter α is defined as

$$\alpha = \frac{v}{(1 - IoU) + v}, \tag{11}$$

by which the overlap area factor is given higher priority for regression, especially for non-overlapping cases.

Finally, the optimization of CIoU loss is same with that of DIoU loss, except that the gradient of v w.r.t. w and h should be specified,

$$\begin{aligned} \frac{\partial v}{\partial w} &= \frac{8}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}) \times \frac{h}{w^2 + h^2}, \\ \frac{\partial v}{\partial h} &= -\frac{8}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}) \times \frac{w}{w^2 + h^2}. \end{aligned} \tag{12}$$

The dominator $w^2 + h^2$ is usually a small value for the cases h and w ranging in $[0, 1]$, which is likely to yield gradient explosion. And thus in our implementation, the dominator $w^2 + h^2$ is simply removed for stable convergence, by which the step size $\frac{1}{w^2+h^2}$ is replaced by 1 and the gradient direction is still consistent with Eqn. (12).
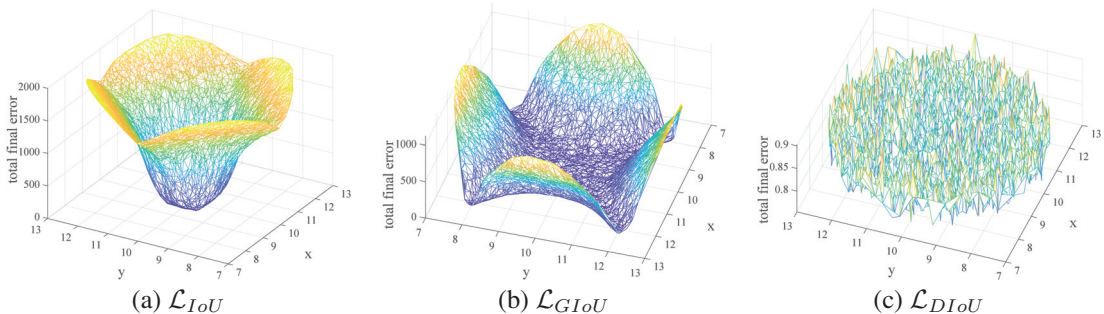
Figure 4: Visualization of regression errors of IoU, GIoU and DIoU losses at the final iteration T, i.e.. $\mathbf{E}(T, n)$ for every coordinate n. We note that the basins in (a) and (b) correspond to good regression cases. One can see that IoU loss has large errors for non-overlapping cases, GIoU loss has large errors for horizontal and vertical cases, and our DIoU loss leads to very small regression errors everywhere.
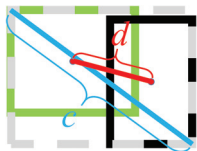

Figure 5: DIoU loss for bounding box regression, where the normalized distance between central points can be directly minimized. c is the diagonal length of the smallest enclosing box covering two boxes, and $d = \rho(\mathbf{b}, \mathbf{b}^{gt})$ is the distance of central points of two boxes.

Non-Maximum Suppression using DIoU

In original NMS, the IoU metric is used to suppress the redundant detection boxes, where the overlap area is the unique factor, often yielding false suppression for the cases with occlusion. We in this work suggest that DIoU is a better criterion for NMS, because not only overlap area but also central point distance between two boxes should also be considered in the suppression criterion. For the predicted box $\mathcal{M}$ with the highest score, the DIoU-NMS can be formally defined as

$$s_i = \begin{cases} s_i, & IoU - \mathcal{R}_{DIoU}(\mathcal{M}, B_i) < \varepsilon, \\ 0, & IoU - \mathcal{R}_{DIoU}(\mathcal{M}, B_i) \geq \varepsilon, \end{cases} \quad (13)$$

where box B_i is removed by simultaneously considering the IoU and the distance between central points of two boxes, s_i is the classification score and ε is the NMS threshold. We suggest that two boxes with distant central points probably locate different objects, and should not be removed. Moreover, the DIoU-NMS is very flexible to be integrated into any object detection pipeline with only a few lines of code.

Experimental Results

In this section, on two popular benchmarks including PASCAL VOC (Everingham et al. 2010) and MS COCO (Lin et al. 2014), we evaluate our proposed DIoU and CIoU losses by incorporating them into the state-of-the-art object detection algorithms including one-stage detection algorithms (i.e., YOLO v3 and SSD) and two-stage algorithm

(i.e., Faster R-CNN). All the source codes and our trained models will be made publicly available.

YOLO v3 on PASCAL VOC

PASCAL VOC (Everingham et al. 2010) is one of the most popular dataset for object detection. YOLO v3 is trained on PASCAL VOC using DIoU and CIoU losses in comparison with IoU and GIoU losses. We use VOC 07+12 (the union of VOC 2007 trainval and VOC 2012 trainval) as training set, containing 16,551 images from 20 classes. And the testing set is VOC 2007 test, which consists of 4,952 images. The backbone network is Darknet608. We follow exactly the GDarknet[1] training protocol released from (Rezatofighi et al. 2019), and the maximum iteration is set to 50K. The performance for each loss has been reported in Table 1. We use the same performance measure, i.e., AP (the average of 10 mAP across different IoU thresholds) = (AP50 + AP55 + . . . + AP95) / 10 and AP75 (mAP@0.75). We also report the evaluation results using GIoU metric.

Table 1: Quantitative comparison of YOLOv3 (Redmon and Farhadi 2018) trained using $\mathcal{L}_{IoU}$ (baseline), $\mathcal{L}_{GIoU}$, $\mathcal{L}_{DIoU}$ and $\mathcal{L}_{CIoU}$. (D) denotes using DIoU-NMS. The results are reported on the test set of PASCAL VOC 2007.

Loss / Evaluation	AP		AP75	
	IoU	GIoU	IoU	GIoU
$\mathcal{L}_{IoU}$	46.57	45.82	49.82	48.76
$\mathcal{L}_{GIoU}$	47.73	46.88	52.20	51.05
Relative improv. %	2.49%	2.31%	4.78%	4.70%
$\mathcal{L}_{DIoU}$	48.10	47.38	52.82	51.88
Relative improv. %	3.29%	3.40%	6.02%	6.40%
$\mathcal{L}_{CIoU}$	49.21	48.42	54.28	52.87
Relative improv. %	5.67%	5.67%	8.95%	8.43%
$\mathcal{L}_{CIoU}(D)$	49.32	48.54	54.74	53.30
Relative improv. %	5.91%	5.94%	9.88%	9.31%

As shown in Table 1, GIoU as a generalized version of IoU, it indeed achieves a certain degree of performance improvement. While DIoU loss can improve the performance

---

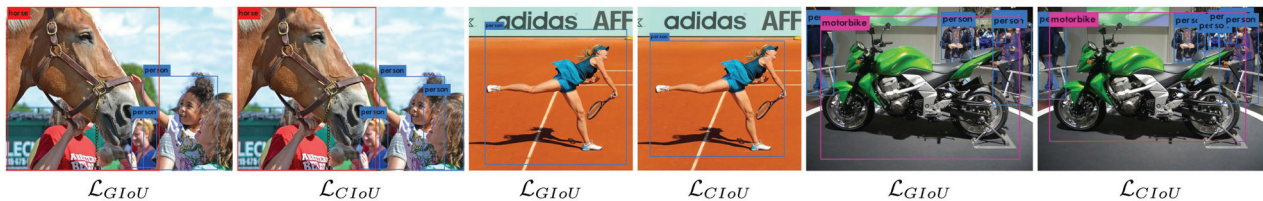[1]https://github.com/generalized-iou/g-darknet

Figure 6: Detection examples using YOLO v3 (Redmon and Farhadi 2018) trained on PASCAL VOC 07+12.

with gains of 3.29% AP and 6.02% AP75 using IoU as evaluation metric. CIoU loss takes the three important geometric factors of two bounding boxes into account, which brings an amazing performance gains, i.e., 5.67% AP and 8.95% AP75. From Fig. 6, one can see that the detection box by CIoU loss is more accurate than that by GIoU loss. Finally, CIoU loss combined with DIoU-NMS brings marvelous improvements of 5.91% AP and 9.88% AP75. Also in terms of GIoU metric, we can come to the same conclusion, validating the effectiveness of the proposed methods. We note that GIoU metric is actually consistent with IoU metric, and thus we only report the IoU metric for the following experiments.

SSD on PASCAL VOC

We use another popular one-stage method SSD to further conduct evaluation experiments. The latest PyTorch implementation of SSD[2] is adopted. Both the training set and testing set share the same setting with YOLO v3 on PASCAL VOC. Following the default training protocol, the max iteration is set to 120K. The backbone network is ResNet-50-FPN. The default bounding box regression loss is smooth ℓ_1-norm, which has different magnitudes with IoU-based losses. And thus there should be a more appropriate trade-off weight for the regression loss to balance with the classification loss. We have observed that for dense anchor algorithms, increasing the regression loss properly can improve the performance. Therefore, for a fair comparison, we fix the weight on regression loss as 5 for these IoU-based losses. And then we train the models using IoU, GIoU, DIoU and CIoU losses. Table 2 gives the quantitative comparison, in which AP and AP75 of IoU metric are reported. For SSD, we can see the consistent improvements of DIoU and CIoU losses in comparison with IoU and GIoU losses.

Faster R-CNN on MS COCO

We also evaluate the proposed method on another more difficult and complex dataset MS COCO 2017 (Lin et al. 2014) using Faster R-CNN[3]. MS COCO is a large-scale dataset, containing more than 118K images for training and 5K images for evaluation. Following the same training protocol of (Rezatofighi et al. 2019), we trained the models using DIoU and CIoU losses in comparison with IoU and GIoU losses. The backbone network is ResNet-50-FPN. Besides AP and AP75 metrics, the evaluation metrics in terms of large, medium and small scale objects are also included. As

[2]https://github.com/JaryHuang/awesome_SSD_FPN_GIoU
[3]https://github.com/generalized-iou/Detectron.pytorch

Table 2: Quantitative comparison of SSD (Liu et al. 2016) trained using $\mathcal{L}_{IoU}$ (baseline), $\mathcal{L}_{GIoU}$, $\mathcal{L}_{DIoU}$ and $\mathcal{L}_{CIoU}$. (D) denotes using DIoU-NMS. The results are reported on the test set of PASCAL VOC 2007.

Loss / Evaluation	AP	AP75
$\mathcal{L}_{IoU}$	51.01	54.74
$\mathcal{L}_{GIoU}$	51.06	55.48
Relative improv. %	0.10%	1.35%
$\mathcal{L}_{DIoU}$	51.31	55.71
Relative improv. %	0.59%	1.77%
$\mathcal{L}_{CIoU}$	51.44	56.16
Relative improv. %	0.84%	2.59%
$\mathcal{L}_{CIoU}(D)$	51.63	56.34
Relative improv. %	1.22%	2.92%

Table 3: Quantitative comparison of Faster R-CNN (Ren et al. 2015) trained using $\mathcal{L}_{IoU}$ (baseline), $\mathcal{L}_{GIoU}$, $\mathcal{L}_{DIoU}$ and $\mathcal{L}_{CIoU}$. (D) denotes using DIoU-NMS. The results are reported on the validation set of MS COCO 2017.

Loss / Evaluation	AP	AP75	APsmall	APmedium	APlarge
$\mathcal{L}_{IoU}$	37.93	40.79	21.58	40.82	50.14
$\mathcal{L}_{GIoU}$	38.02	41.11	21.45	41.06	50.21
Relative improv. %	0.24%	0.78%	-0.60%	0.59%	0.14%
$\mathcal{L}_{DIoU}$	38.09	41.11	21.66	41.18	50.32
Relative improv. %	0.42%	0.78%	0.31%	0.88%	0.36%
$\mathcal{L}_{CIoU}$	38.65	41.96	21.32	41.83	51.51
Relative improv. %	1.90%	2.87%	-1.20%	2.47%	2.73%
$\mathcal{L}_{CIoU}(D)$	38.71	42.07	21.37	41.93	51.60
Relative improv. %	2.06%	3.14%	-0.97%	2.72%	2.91%

for the trade-off weight for regression loss, we set the weight as 12 for all losses for a fair comparison. Table 3 reports the quantitative comparison.

Faster R-CNN is a detection algorithm with dense anchor boxes, and is usually with high IoU levels in the initial situation. Geometrically speaking, the regression cases of Faster R-CNN are likely to place in the basins of Fig. 4, where IoU, GIoU and DIoU losses all have good performance. Therefore, GIoU loss has very small gain than IoU loss, as shown in Table 3. But our DIoU and CIoU losses still contribute to performance improvements than IoU and GIoU losses in terms of AP, AP75, APmedium and APlarge. Especially the gains by CIoU loss are very significant. From Fig. 7, one can easily find more accurate detection boxes by CIoU loss than those by GIoU loss. One may have noticed that in terms of APsmall, CIoU loss is a little inferior to the original IoU
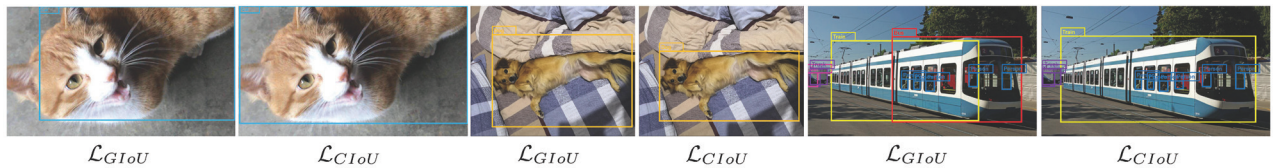
Figure 7: Detection examples using Faster R-CNN (Ren et al. 2015) trained on MS COCO 2017.

loss, while DIoU loss is better than all the other losses. That is to say the consistency of aspect ratio may not contribute to the regression accuracy for small objects. Actually it is reasonable that for small objects, the central point distance is more important than aspect ratio for regression, and aspect ratio may weaken the effect of normalized distance between the two boxes. Nevertheless, CIoU loss performs much better for medium and large objects, and for small objects, the adverse effects can be relieved by DIoU-NMS.

Discussion on DIoU-NMS

In Tables 1, 2 and 3, we report the results of CIoU loss cooperating with original NMS ($\mathcal{L}_{CIoU}$) and DIoU-NMS ($\mathcal{L}_{CIoU}(D)$), where the thresholds follow the default settings of original NMS, i.e., $\varepsilon = 0.45$ for YOLO v3 and SSD, and $\varepsilon = 0.50$ for Faster R-CNN. One can find that DIoU-NMS makes further performance improvements than original NMS for most cases. Fig. 8 shows that DIoU-NMS can better preserve the correct detection boxes, where YOLO v3 trained on PASCAL VOC is adopted to detect objects on MS COCO. To further validate the superiority of DIoU-NMS over original NMS, we conduct comparison experiments, where original NMS and DIoU-NMS are cooperated with YOLO v3 and SSD trained using CIoU loss. We present the comparison of original NMS and DIoU-NMS within a wide range of thresholds $[0.43, 0.48]$. From Fig. 9, one can see that DIoU-NMS is better than original NMS for every threshold. Furthermore, it is worth noting that even the worst performance of DIoU-NMS is at least comparable or better than the best performance of original NMS. That is to say our DIoU-NMS can generally perform better than original NMS even without carefully tuning the threshold ε.
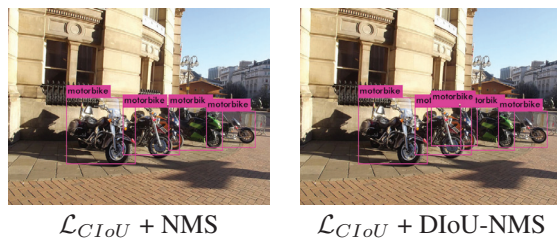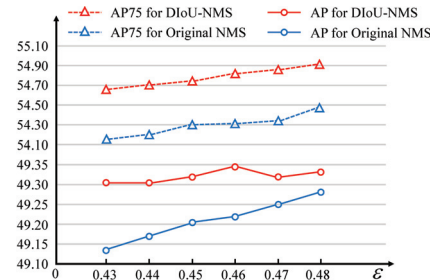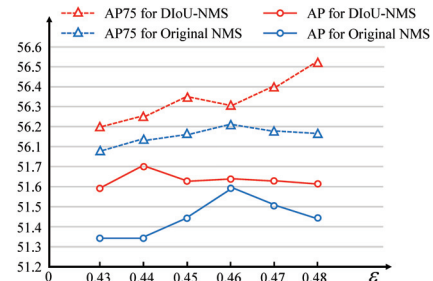

Figure 8: Detection example from MS COCO 2017 using YOLO v3 (Redmon and Farhadi 2018) trained on PASCAL VOC 07+12.

Conclusion

In this paper, we proposed two losses, i.e., DIoU loss and CIoU loss, for bounding box regression along with DIoU-

(a) YOLO v3 on the test set of PASCAL VOC 2007

(b) SSD on the test set of PASCAL VOC 2007

Figure 9: Comparison of DIoU-NMS and original NMS for different thresholds ε. The models of YOLO v3 and SSD are trained on PASCAL VOC 07+12 using $\mathcal{L}_{CIoU}$.

NMS for suppressing redundant detection boxes. By directly minimizing the normalized distance of two central points, DIoU loss can achieve faster convergence than GIoU loss. CIoU loss takes three geometric properties into account, i.e., overlap area, central point distance and aspect ratio, and leads to faster convergence and better performance. The proposed losses and DIoU-NMS can be easily incorporated to any object detection pipeline, and achieve superior results on benchmarks.

References

Bae, S.-H. 2019. Object detection based on region decomposition and assembly. In *The AAAI Conference on Artificial*

Intelligence.

Bodla, N.; Singh, B.; Chellappa, R.; and Davis, L. S. 2017. Soft-nms – improving object detection with one line of code. In *The IEEE International Conference on Computer Vision (ICCV)*.

Cai, Z., and Vasconcelos, N. 2018. Cascade r-cnn: Delving into high quality object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-balanced loss based on effective number of samples. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision* 88(2):303–338.

Fu, C.-Y.; Liu, W.; Ranga, A.; Tyagi, A.; and Berg, A. C. 2017. DSSD: Deconvolutional single shot detector. *arXiv:1701.06659*.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Girshick, R. 2015. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*.

He, K.; Gkioxari, G.; Dollar, P.; and Girshick, R. 2017. Mask r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*.

He, Y.; Zhu, C.; Wang, J.; Savvides, M.; and Zhang, X. 2019. Bounding box regression with uncertainty for accurate object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; and Jiang, Y. 2018. Acquisition of localization confidence for accurate object detection. In *The European Conference on Computer Vision (ECCV)*.

Law, H., and Deng, J. 2018. Cornernet: Detecting objects as paired keypoints. In *The European Conference on Computer Vision (ECCV)*.

Li, B.; Liu, Y.; and Wang, X. 2019. Gradient harmonized single-stage detector. In *The AAAI Conference on Artificial Intelligence*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *The European Conference on Computer Vision (ECCV)*.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollar, P. 2017. Focal loss for dense object detection. In *The IEEE International Conference on Computer Vision (ICCV)*.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. Ssd: Single shot multibox detector. In *The European Conference on Computer Vision (ECCV)*.

Liu, S.; Huang, D.; and Wang, Y. 2019. Adaptive nms: Re-fining pedestrian detection in a crowd. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; and Lin, D. 2019. Libra r-cnn: Towards balanced learning for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Redmon, J., and Farhadi, A. 2017. Yolo9000: Better, faster, stronger. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Redmon, J., and Farhadi, A. 2018. Yolov3: An incremental improvement. *arXiv:1804.02767*.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28*.

Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; and Savarese, S. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Song, T.; Sun, L.; Xie, D.; Sun, H.; and Pu, S. 2018. Small-scale pedestrian detection based on topological line localization and temporal feature aggregation. In *The European Conference on Computer Vision (ECCV)*.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. FCOS: Fully convolutional one-stage object detection. In *The IEEE International Conference on Computer Vision (ICCV)*.

Wang, H.; Wang, Q.; Gao, M.; Li, P.; and Zuo, W. 2018. Multi-scale location-aware kernel representation for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, H.; Wang, Q.; Yang, F.; Zhang, W.; and Zuo, W. 2019. Data augmentation for object detection via progressive and selective instance-switching. *arXiv:1906.00358*.

Yang, T.; Zhang, X.; Li, Z.; Zhang, W.; and Sun, J. 2018. Metaanchor: Learning to detect objects with customized anchors. In *Advances in Neural Information Processing Systems*.

Yang, Z.; Liu, S.; Hu, H.; Wang, L.; and Lin, S. 2019. Reppoints: Point set representation for object detection. In *The IEEE International Conference on Computer Vision (ICCV)*.

Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; and Huang, T. 2016. Unitbox: An advanced object detection network. In *Proceedings of the ACM International Conference on Multimedia*.

Zhu, C.; He, Y.; and Savvides, M. 2019. Feature selective anchor-free module for single-shot object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.