

Gaussian Guided IoU: A Better Metric for Balanced Learning on Object Detection

Shengkai Wu^{†1} Jinrong Yang^{†1} Lijun Gou¹ Hangcheng Yu¹ Xiaoping Li^{*1}

¹State Key Laboratory of Digital Manufacturing Equipment and Technology
Huazhong University of Science and Technology, Wuhan, 430074, China.

[†]Equal contribution

^{*}Corresponding Author

{ShengkaiWu, yangjinrong}@hust.edu.cn

Abstract

For most of the anchor-based detectors, Intersection over Union(IoU) is widely utilized to assign targets for the anchors during training. However, IoU pays insufficient attention to the closeness of the anchor's center to the truth box's center. This results in two problems: (1) only one anchor is assigned to most of the slender objects which leads to insufficient supervision information for the slender objects during training and the performance on the slender objects is hurt; (2) IoU can not accurately represent the alignment degree between the receptive field of the feature at the anchor's center and the object. Thus during training, some features whose receptive field aligns better with objects are missing while some features whose receptive field aligns worse with objects are adopted. This hurts the localization accuracy of models. To solve these problems, we firstly design Gaussian Guided IoU(GGIoU) which focuses more attention on the closeness of the anchor's center to the truth box's center. Then we propose GGIoU-balanced learning method including GGIoU-guided assignment strategy and GGIoU-balanced localization loss. The method can assign multiple anchors for each slender object and bias the training process to the features well-aligned with objects. Extensive experiments on the popular benchmarks such as PASCAL VOC and MS COCO demonstrate GGIoU-balanced learning can solve the above problems and substantially improve the performance of the object detection model, especially in the localization accuracy.

1. Introduction

With the advent of deep learning, object detection have developed quickly and many object detectors have been proposed. Most of object detectors can be classified into anchor-based detectors[18, 14, 7, 12, 16] and anchor-free

detectors[24, 15, 20, 11]. Anchor-based detectors tile densely anchors with different scales and aspect ratios at each position on the feature map. For anchor-based detectors, Intersection over Union(IoU) is adopted as a metric to assign targets for anchors during training. IoU is the most popular metric for comparing the similarity between two arbitrary shapes due to its invariance of the scale of boxes. However, IoU pays insufficient attention to the closeness of the anchor's center to the truth box's center, resulting in two problems when utilizing IoU to assign targets.

Firstly, only one anchor can be assigned to the truth box for most of the slender objects when IoU is used to assign targets. RetinaNet[12] sets IoU threshold (0.4, 0.5) to assign positive examples and negative examples. When the IoU between an anchor and the nearest truth box is smaller than 0.4, this anchor is assigned to be negative example and when the IoU between an anchor and the nearest truth box is not smaller than 0.5, this anchor is assigned to the corresponding nearest truth box as positive example. Besides, RetinaNet assigns the truth box with the nearest anchors to ensure there exists at least one anchor for each truth box. Other anchor-based detectors such as Faster R-CNN[18] and SSD[14] adopt a similar strategy to assign targets during training. With this assignment strategy, most of the slender objects can only be assigned with one anchor. Thus, there will be insufficient supervision information for training slender objects so that the trained models will perform badly for the slender objects. As Fig.1(a) shows, the red solid box is a slender object and the IoU between the nearby anchors and the truth box is below the positive threshold. Thus only the nearest green anchor can be assigned to the truth box when IoU is utilized to assign targets.

Secondly, IoU can not accurately represent the alignment degree of the receptive field of the feature at the anchor's center. Each anchor's center is always attached

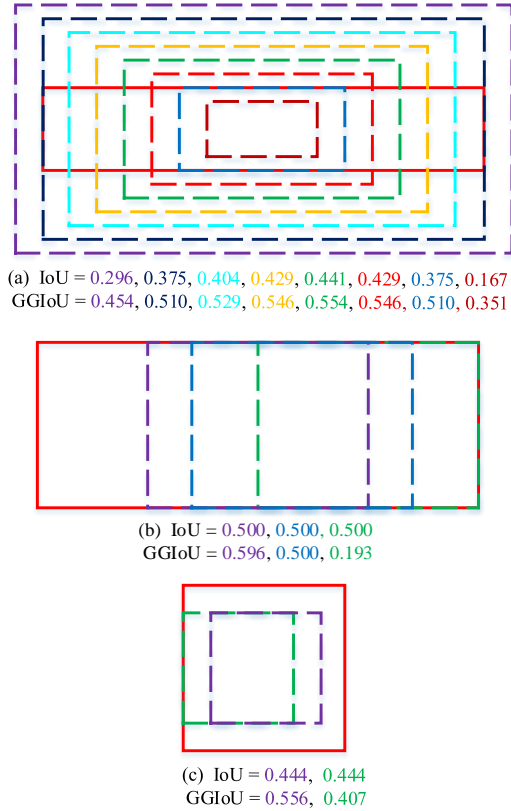


Figure 1. Comparison between the impact of IoU and GGIoU. The boxes with solid lines represent truth boxes and the boxes with dashed lines represent anchors. (a) The IoU between the slender object and most of the anchors is small and only the nearest anchor can be assigned to the slender object when IoU is used to assign targets. But the GGIoU is large and multiple anchors can be assigned to the slender object when GGIoU is used to assign targets. (b) The receptive field of the feature at the purple anchor’s center aligns better with the object than that of the features at the other anchors’ centers. But IoU of these three anchors is the same and can not accurately represent the alignment degree of the features. The GGIoU of these anchors is different and can represent the alignment degree more accurately. (c) is similar to (b).

with a feature on the feature map. And the closeness of a feature to the truth box’s center can represent the alignment degree between the receptive field of the feature and the truth box. The closer the feature is to the truth box’s center, the better the receptive field of the feature aligns with the truth box. However, the IoU of anchors can not accurately represent the alignment degree of the receptive field of the corresponding feature. For example, as Fig.1(b), (c) show, the purple anchor’s center are closer to the truth box’s center than the green anchor’s center. This reveals that the receptive field of the feature at the purple anchor’s center aligns better than that of the feature at the green anchor’s center. But the IoU of the purple anchor is the same as that

of the green anchor. Thus, IoU fails to reveal the different alignment degree between the purple anchor and green anchor. During training, when IoU is used to assign targets, some features whose receptive field aligns better with the object are missing such as the feature at the purple anchor’s center in Fig.1(c) while some features whose receptive field aligns worse with the object are adopted such as the feature at the green anchor’s center in Fig.1(b), which can severely hurt the model’s performance.

To solve these problems, we propose a better metric named Gaussian Guided IoU (GGIoU) to replace the standard IoU. We compute the Gaussian distance from each anchor’s center to the truth box’s center and then obtain GGIoU by multiplying the Gaussian distance with the standard IoU. Gaussian distance measures the closeness of the anchor’s center to the truth box’s center. The closer the anchor’s center is to the truth box’s center, the larger the Gaussian distance is. Thus, GGIoU pays more attention to the closeness of the anchor’s center to the truth box’s center than the standard IoU. We design GGIoU-balanced learning including GGIoU-guided assignment strategy and GGIoU-balanced localization loss. This method makes the learning process more balanced and the above two problems can be solved. Extensive experiments on PASCAL VOC [6] and MS COCO [13] demonstrate that GGIoU-balanced learning can substantially improve the performance of object detectors, especially in the localization accuracy.

The main contribution of the paper is summarized as follows:

- We design GGIoU by incorporating Gaussian distance into IoU as a better metric for comparing the similarity between two arbitrary shapes.
- We design GGIoU-balanced learning method including GGIoU-guided assignment strategy and GGIoU-balanced localization loss, which makes the detectors more powerful for accurate localization.
- We conduct extensive experiments on the popular benchmarks to demonstrate that GGIoU-balanced learning method can substantially improve the performance of the popular object detectors.

2. Related Work

IoU-based evaluation metric. UnitBox[24] directly uses IoU loss to regress the four bounds of a predicted box as a whole unit. GIoU[19] 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, DIoU and CIoU losses[27] is proposed to perform faster convergence and better regression accuracy. GIoU, CIoU and DIoU is designed for the localization loss and can’t be used for target assignment. Differently, GGIoU is mainly designed for the target assign-

ment.

Target assignment strategies in anchor-based detectors. Anchor-based detectors commonly utilize IoU to assign targets. RetinaNet[12] assigns the anchors with IoU below the negative threshold as negative examples and the anchors with IoU not smaller than the positive threshold as positive examples. Faster RCNN[18] and SSD[14] adopt the similar strategy. FreeAnchor[26] constructs a bag of anchors for each truth box based on the IoU and proposes detection customized loss for learning to match the best anchors. Similarly, MAL[10] constructs bag anchors for each truth box and select the most representative anchor based on the classification and localization score. Besides, anchor depression is designed to perturb the features of selected anchors to decrease their confidence. ATSS[25] select K anchors from each level of FPN for each truth box and compute IoU for these anchors. Then the IoU threshold is adaptively computed based on the IoU distribution for each truth box. Differently, we design GGIOU as an alternative and propose GGIOU-guided assignment strategy.

Feature alignment in object detection. Feature alignment is important for the performance of object detection. Multi-stage detectors[18, 9, 1] adopt RoI pooling to extract the aligned feature for each proposal. Deformable convolution[4, 28] predict the offset for each position in the convolution kernels and the receptive field of kernels can adaptively change based on the input feature. This makes feature align better with the object. Guided anchoring[21] utilize the predicted box from the first stage to predict the offsets of deformable convolution and the deformable convolution can generate well-aligned feature for the second stage detector. AlignDet[3] predict proposals at each position in the first stage and densely adopt RoIConv to extract the aligned region feature. Most of the models change the method of feature extraction to extract aligned feature. Differently, GGIOU-balanced learning make the training process focus more on the features close to the truth box center, which aligns better with objects.

3. GGIOU-balanced Learning Method

3.1. Gaussian Guided Intersection over Union

For anchor-based detectors, Intersection over Union (IoU) is widely used to assign targets for the anchors because of its appealing property of scale-invariant. IoU is computed by:

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

where $B^a = (x_1^a, y_1^a, x_2^a, y_2^a)$ is the coordinates of the top-left and bottom-right corners for the anchor box, and $B^{gt} = (x_1^{gt}, y_1^{gt}, x_2^{gt}, y_2^{gt})$ is the coordinates of the top-left and bottom-right corners for the ground truth box. As an-

Algorithm 1: Gaussian Guided Intersection over Union (GGIOU)

input : Parameter: β ,
 anchor box B^a : $B^a = (x_1^a, y_1^a, x_2^a, y_2^a)$,
 ground truth box B^{gt} :
 $B^{gt} = (x_1^{gt}, y_1^{gt}, x_2^{gt}, y_2^{gt})$.

output: GGIOU.

- 1 For the anchor B^a , ensuring $x_2^a > x_1^a$ and $y_2^a > y_1^a$:
 $\hat{x}_1^a = \min(x_1^a, x_2^a)$, $\hat{x}_2^a = \max(x_1^a, x_2^a)$,
 $\hat{y}_1^a = \min(y_1^a, y_2^a)$, $\hat{y}_2^a = \max(y_1^a, y_2^a)$.
 - 2 Calculating area of B^{gt} :
 $S^{gt} = (x_2^{gt} - x_1^{gt}) \times (y_2^{gt} - y_1^{gt})$.
 - 3 Calculating area of B^a :
 $S^a = (\hat{x}_2^a - \hat{x}_1^a) \times (\hat{y}_2^a - \hat{y}_1^a)$.
 - 4 Calculating intersection \mathcal{I} between B^a and B^{gt} :
 $x_1^{\mathcal{I}} = \max(\hat{x}_1^a, x_1^{gt})$, $x_2^{\mathcal{I}} = \min(\hat{x}_2^a, x_2^{gt})$,
 $y_1^{\mathcal{I}} = \max(\hat{y}_1^a, y_1^{gt})$, $y_2^{\mathcal{I}} = \min(\hat{y}_2^a, y_2^{gt})$,
 $w^{\mathcal{I}} = \begin{cases} (x_2^{\mathcal{I}} - x_1^{\mathcal{I}}) & \text{if } x_2^{\mathcal{I}} > x_1^{\mathcal{I}} \\ 0 & \text{otherwise.} \end{cases}$,
 $h^{\mathcal{I}} = \begin{cases} (y_2^{\mathcal{I}} - y_1^{\mathcal{I}}) & \text{if } y_2^{\mathcal{I}} > y_1^{\mathcal{I}} \\ 0 & \text{otherwise.} \end{cases}$,
 $\mathcal{I} = w^{\mathcal{I}} \times h^{\mathcal{I}}$.
 - 5 $IoU = \frac{\mathcal{I}}{\mathcal{U}}$, where $\mathcal{U} = S^a + S^{gt} - \mathcal{I}$.
 - 6 Calculating the center coordinates of B^a :
 $x_c^a = \frac{x_1^a + x_2^a}{2}$, $y_c^a = \frac{y_1^a + y_2^a}{2}$.
 - 7 Calculating the center coordinates of B^{gt} :
 $x_c^{gt} = \frac{x_1^{gt} + x_2^{gt}}{2}$, $y_c^{gt} = \frac{y_1^{gt} + y_2^{gt}}{2}$.
 - 8 Calculating Gaussian variance:
 $\sigma_1 = \beta \times w^{\mathcal{I}}$, $\sigma_2 = \beta \times h^{\mathcal{I}}$.
 - 9 Calculating the Gaussian distance between the centers of truth box and anchor box D_c :
 $D_c = e^{-\frac{1}{2} \left(\frac{(x_c^a - x_c^{gt})^2}{\sigma_1^2} + \frac{(y_c^a - y_c^{gt})^2}{\sigma_2^2} \right)}$.
 - 10 $GGIOU = IoU^{(1-\alpha)} D_c^\alpha$.
-

alyzed above, there exist two problems when utilizing the standard IoU to assign targets for the anchors during training.

To address these problems, we design a better metric named GGIOU to replace the standard IoU. We compute Gaussian distance D_c from each anchor's center to the truth box's center and obtain Gaussian Guided IoU (GGIOU) by multiplying Gaussian distance with the standard IoU as Eq.2 and Eq.4 show.

$$D_c = e^{-\frac{1}{2} \left(\frac{(x_c^a - x_c^{gt})^2}{\sigma_1^2} + \frac{(y_c^a - y_c^{gt})^2}{\sigma_2^2} \right)}, \quad (2)$$

$$\sigma_1 = \beta \times w^{\mathcal{I}}, \quad \sigma_2 = \beta \times h^{\mathcal{I}}, \quad (3)$$

$$GGIOU = IoU^{(1-\alpha)} D_c^\alpha, \quad (4)$$

(x_c^a, y_c^a) and (x_c^{gt}, y_c^{gt}) represent the coordinates of the anchor box's center and the truth box's center respectively. σ_1 and σ_2 represent the standard deviation in the x and y directions respectively and are computed by multiplying the parameter β with the width w^I and height h^I of the overlap box respectively. Gaussian distance D_c measures the closeness of the anchor's center to the truth box's center. The closer the anchor's center is to the truth box's center, the larger the Gaussian distance is. The range of D_c is $(0, 1]$. Adopting the width w^I and height h^I of the overlap box to compute σ_1 and σ_2 is an important design. Assuming that there exist two anchors whose centers are the same but the size of overlap boxes are different, the anchor with a larger overlap box will get a larger standard deviation and thus get larger D_c , vice versa. Thus, Gaussian distance D_c not only represents the closeness of the anchor's center to the truth box's center but also contains the size information of the overlap box between anchors and truth boxes. How about adopting the width and height of the truth box to compute the standard deviation σ_1 and σ_2 ? Obviously, D_c of the anchor with a larger overlap box is the same as that of the anchor with a smaller overlap box under this condition and the size information of the overlap box is missing. Thus, adopting the overlap box to compute the standard deviation is more reasonable and this will be demonstrated in the subsequent experimental results. Compared with IoU, GGIoU focuses more attention on the closeness of the anchor box's center to the truth box's center. Alg. 1 describes the specific computation process of GGIoU.

3.2. GGIoU-guided Assignment Strategy

To solve the above two problems, we design GGIoU-guided assignment strategy, which utilizes GGIoU to assign targets. Firstly, we compute the GGIoU between all the anchors and truth boxes. Then the anchors whose GGIoU with the nearest truth box is smaller than the negative threshold are assigned to be negative examples. The anchors whose GGIoU with the nearest truth box is not smaller than the positive threshold are assigned to be positive examples and are assigned to the nearest truth box. Finally, the truth boxes are assigned with the nearest anchors to ensure that all the truth boxes can be assigned with at least one anchor. GGIoU-guided assignment strategy can solve the above problem in the following ways.

Firstly, as analyzed above, only one anchor can be assigned to the truth box for most of the slender objects when the standard IoU is used to assign targets. This problem can be solved by GGIoU-guided assignment strategy. GGIoU is influenced by both IoU and Gaussian distance D_c . For the anchors that are close to the center of the slender object, the Gaussian distance D_c is large and can result in larger GGIoU even if IoU is small. As shown in Fig.1(a), the IoU of anchors (dotted boxes) with the slender object (red

solid box) is quite small (0.296, 0.375, 0.404, 0.429, 0.441, 0.429, 0.375, 0.167) and only the nearest green anchor can be assigned to the slender object. However, the GGIoU of these anchors is larger (0.454, 0.510, 0.529, 0.546, 0.554, 0.546, 0.510, 0.351) because these anchors are at the center of the truth box. When GGIoU is utilized to assign targets, multiple anchors will be assigned to this slender object and more supervision information will be provided to train the model for these slender objects. Thus, the performance on the slender objects is improved.

Secondly, IoU can not accurately represent the alignment degree of the feature at the anchor's center with the truth box, which leads to that some features that align badly with the objects are adopted while some features that align well with the objects are missing during training. This problem can be alleviated by GGIoU-guided assignment strategy. As Fig.1(b) shows, the IoU of the purple anchor is the same as that of the green anchor and they all can be assigned to the truth box. But obviously, the receptive field of the feature at the purple anchor's center aligns better than that of the features at the green anchor's center. Thus, IoU can not accurately represent the alignment degree of the features. When IoU is utilized to assign targets, the feature at the green anchor's center which aligns worst with the object is adopted for training the object. But when GGIoU is used, the GGIoU of the green anchor is smaller than IoU and this green anchor is discarded. As Fig.1(c) shows, when IoU is used to assign targets, both the green and purple anchors are assigned to be negative examples and the feature at the purple anchor's center which aligns well with the object is missing. But when GGIoU is used, the central purple anchor is assigned to be a positive example and the corresponding feature is adopted for training this object. In summary, GGIoU-guided assignment strategy makes the training process bias to the features aligning well with objects (the purple anchor in Fig.1(c)) while discarding the features aligning badly with objects (the green anchor in Fig.1(b)).

There still exists a problem. As Fig.1(b) shows, the feature at the blue anchor's center aligns worse than the feature at the purple anchor's center. No matter IoU or GGIoU is used, the blue anchor can be assigned to be a positive example and the feature at the purple anchor's center is treated the same as that at the purple anchor's center. Can we bias the model more to the features aligning better with the objects? Inspired by IoU-balanced localization loss[22], we design GGIoU-balanced localization loss to realize this goal.

3.3. GGIoU-balanced Localization Loss

Under the framework of traditional training method, all target samples are equally treated by sharing the same weight in localization loss function. To guide the model pay more attention to the features aligning better with the objects, we proposed GGIoU-balanced localization loss as

Table 1. The results of sampling positive and negative anchors based on GGIoU using the width and height of ground truths to export the standard deviation of Gaussian distances. $\alpha = 0$ means the baseline model using IoU to sampling.

β	α	mAP	AP ₅₀	AP ₆₀	AP ₇₀	AP ₈₀	AP ₉₀
-	0	0.308	0.496	0.450	0.376	0.261	0.086
1/6	0.4	0.309	0.487	0.448	0.377	0.270	0.094
1/6	0.3	0.313	0.492	0.451	0.381	0.271	0.095
1/6	0.2	0.309	0.490	0.448	0.377	0.268	0.089
1/5	0.4	0.306	0.485	0.444	0.374	0.264	0.090
1/5	0.3	0.313	0.496	0.455	0.384	0.271	0.094
1/5	0.2	0.312	0.493	0.452	0.383	0.268	0.093

Table 2. The results of sampling positive and negative anchors based on GGIoU using the width and height of overlap regions between ground truths and predicting boxes to export the standard deviation of Gaussian distances. $\alpha = 0$ means the baseline model using IoU to sampling.

β	α	mAP	AP ₅₀	AP ₆₀	AP ₇₀	AP ₈₀	AP ₉₀
-	0	0.308	0.496	0.450	0.376	0.261	0.086
1/6	0.3	0.317	0.499	0.456	0.387	0.277	0.101
1/7	0.3	0.317	0.500	0.458	0.387	0.278	0.100
1/8	0.3	0.315	0.494	0.452	0.385	0.277	0.097
1/6	0.4	0.315	0.496	0.456	0.383	0.273	0.098
1/7	0.4	0.316	0.498	0.456	0.383	0.276	0.099
1/8	0.4	0.314	0.493	0.451	0.384	0.275	0.098

Eq.5 and Eq.6 shows:

$$L_{loc} = \sum_{i \in Pos} \sum_{m \in cx, cy, w, h}^N w_i(GGIoU_i) * \text{smooth}_{L1}(l_i^m - \hat{g}_i^m) \quad (5)$$

$$w_i(GGIoU_i) = w_{loc} * GGIoU_i^\lambda \quad (6)$$

where $(l_i^{cx}, l_i^{cy}, l_i^w, l_i^h)$ represents the encoded coordinates of prediction box and $(\hat{g}_i^{cx}, \hat{g}_i^{cy}, \hat{g}_i^w, \hat{g}_i^h)$ represents the encoded coordinates of truth boxes, which is the same as the decoding style of Faster R-CNN[18]. w_{loc} represents the weight of localization loss which is manually adjusted to ensure the loss in the first iteration the same as the baseline counterpart.

There are two merits when using GGIoU-balanced localization function: (1) GGIoU-balanced localization loss can suppress the gradient of outliers during training; (2) GGIoU-balanced localization loss make the training process pay more attention to the features aligning better with objects.

4. Experimental Results

4.1. Experiment Setting

Dataset. All detection baselines are trained and evaluated on three object detection datasets including MS COCO [13], PASCAL VOC [6] and SCD[23]. MS COCO contains about 118K images for training(*train-2017*), 5k images for validation(*val-2017*) and 20k for testing(*test-dev*). For PASCAL VOC, the VOC2007 includes 5011

training images(*VOC2007 trainval*) and 4952 testing images(*VOC2007 test*) while VOC2012 contains 11540 training images and 10591 testing images. The combination of *VOC2007 trainval* and *VOC2012 trainval* are adopted to train models while VOC2007 (*VOC2007 test*) is used to evaluate models. SCD is a dense carton detection and segmentation dataset and contains LSCD and OSCD subsets. We conduct experiments on LSCD which contains 6735 image for training and 1000 images for testing.

Evaluation Protocol. We adopt the same evaluation metric as MS COCO 2017 Challenge [13]. This includes AP(averaged mAP across different IoU thresholds $IoU = \{.5, .55, \dots, .95\}$), AP₅₀(AP at IoU threshold of 0.5), AP₇₅(AP at IoU threshold of 0.75), AP_S (AP for small scales), AP_M (AP for medium scales) and AP_L (AP for large scales).

Implementation Details. All experiments are implemented by using PyTorch and MMDetection [2]. All models are trained on 4 NVIDIA GeForce 2080Ti GPU while the learning rate changes linearly based on the mini-batch [8]. For all ablation studies, RetinaNet with ResNet18 as backbone are trained on *train-2017* and evaluated on *val-2017* with image scale of [800, 1333]. For all baseline models, we follow the default settings of MMDetection.

4.2. Ablation studies on using GGIoU as sampling metric

Ablating the style of calculating the standard deviation of Gaussian distance. As shown in Table 1, when using the width and height of truth boxes to export the standard deviation σ_1 and σ_2 of Gaussian distances, the sampling strategy based on GGIoU can improve the model's AP by up to 0.5%. Table 2 reports the results that when the overlapping area between ground truth boxes and prediction boxes acts as base size to derive standard deviations, 0.9% mAP is boost higher 0.4% than the base size of IoU counterpart. These results echo the argument in Section 3.1 and verify that employing the shape of the overlapping area between the truth boxes and the prediction boxes can not only measures the distance of two boxes but reflects the overlapping scale which makes reasonable.

The analyse of improvements for localization accuracy based on using GGIoU as sampling metric. Table 2 highlights that the performance of detector is improved from 30.8% to 31.7%. With 0.3% \sim 0.6% improvements in AP₅₀ \sim AP₆₀ and 1.1% \sim 1.6% increases in AP₇₀ \sim AP₉₀, it reveals that the sampling strategy based on GGIoU is benefit to facilitate localization. It attributes to paying more attention to the features which receptive field is better aligned with the objects. This favor encourage model to touch the precise boundary.

The analyse of improvements for slender objects based on using GGIoU as sampling metric. As shown

method	γ	ω_{loc}	mAP	AP ₅₀	AP ₆₀	AP ₇₀	AP ₈₀	AP ₉₀	AP _S	AP _M	AP _L
baseline	-	1.0	0.308	0.496	0.450	0.376	0.261	0.086	0.161	0.340	0.407
IoU	1.4	2.6	0.315	0.493	0.450	0.381	0.275	0.101	0.160	0.345	0.419
IoU	1.3	2.5	0.316	0.496	0.452	0.384	0.276	0.104	0.168	0.345	0.423
IoU	1.2	2.4	0.315	0.496	0.452	0.383	0.275	0.104	0.167	0.347	0.418
GGIoU	1.5	2.9	0.317	0.494	0.451	0.386	0.279	0.107	0.166	0.345	0.426
GGIoU	1.4	2.8	0.319	0.497	0.455	0.386	0.282	0.107	0.167	0.348	0.430
GGIoU	1.3	2.6	0.319	0.500	0.457	0.388	0.278	0.104	0.165	0.348	0.432
GGIoU	1.2	2.5	0.316	0.495	0.452	0.384	0.276	0.105	0.170	0.344	0.420



Table 3. Comparison between IoU based and GGIoU based balanced localization function. γ and ω_{loc} are the coupled hyper-parameters.

method	γ	ω_{loc}	mAP	AP ₅₀	AP ₆₀	AP ₇₀	AP ₈₀	AP ₉₀	AP _S	AP _M	AP _L
baseline	-	1.0	0.308	0.496	0.450	0.376	0.261	0.086	0.161	0.340	0.407
IoU	1.4	2.6	0.315	0.493	0.450	0.381	0.275	0.101	0.160	0.345	0.419
IoU	1.3	2.5	0.316	0.496	0.452	0.384	0.276	0.104	0.168	0.345	0.423
IoU	1.2	2.4	0.315	0.496	0.452	0.383	0.275	0.104	0.167	0.347	0.418
GGIoU	1.5	2.9	0.317	0.494	0.451	0.386	0.279	0.107	0.166	0.345	0.426
GGIoU	1.4	2.8	0.319	0.497	0.455	0.386	0.282	0.107	0.167	0.348	0.430
GGIoU	1.3	2.6	0.319	0.500	0.457	0.388	0.278	0.104	0.165	0.348	0.432
GGIoU	1.2	2.5	0.316	0.495	0.452	0.384	0.276	0.105	0.170	0.344	0.420

Table 4. Comparison between IoU based and GGIoU based balanced localization function. γ and ω_{loc} are the coupled hyper-parameters.

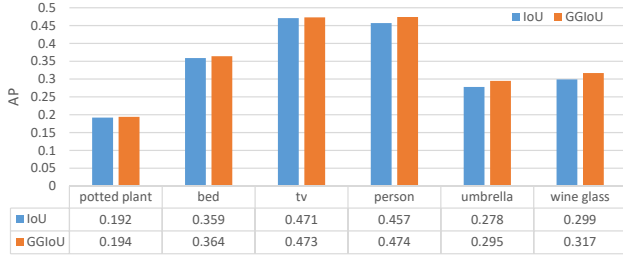


Figure 2. Comparison between IoU based and GGIoU based sampling strategy with respect to slender objects.

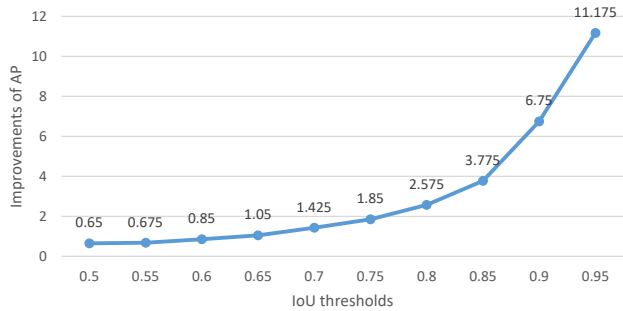


Figure 3. The average improvements of ResNet18, ResNet50, ResNet101 and ResNeXt-32x4d-101 with the increases of AP's thresholds on SCD.

in Figure 2, we list the performance improvements with respect to some non-slender objects, such as potted plant, bed and tv, decent gains are achieved by 0.2% \sim 0.5% when using our proposal GGIoU to assign positive and negative samples. For slender objects such as person, umbrella and wineglass, significant improvements are achieved by 1.7% \sim 1.8%. The results verify the thesis in Section 3.2 and reveal that slender objects harvest more positive samples instead of only one, balancing the attention to all objects and enhance the overall performance.

4.3. Ablation experiments for localization loss function based on GGIoU as weighting score.

For simplicity, the parameters of GGIoU directly adopt the optimal values by ablating in Table 2, setting $\sigma = 1/6$ and $\alpha = 0.3$. In Eq.5 and Eq.6, IoU is replaced by GGIoU to calculate weights for positive samples.

Ablating the controlling parameters. As shown in Table 4, when γ is set as 1.4, the proposal localization loss function balanced by GGIoU achieves best localization accuracy. Without adding bells and whistles, it improves the AP of baseline by 1.1% on MS COCO *val2017* while powerful gains are achieved by 1.0% \sim 2.1%. In addition, decent gains are achieved in AP_{50} and AP_{60} by 0.1% and 0.5% respectively. The experiments indicate that the GGIoU-balanced localization function can effectively promote localization performance.

Comparison between GGIoU and IoU on balanced lo-

Backbone	mAP	AP ₅₀	AP ₆₀	AP ₇₀	AP ₈₀	AP ₉₀	AP _S	AP _M	AP _L
ResNet18	0.308	0.496	0.450	0.376	0.261	0.086	0.161	0.340	0.407
ResNet50	0.356	0.555	0.510	0.432	0.311	0.113	0.200	0.396	0.468
ResNet101	0.377	0.575	0.533	0.460	0.337	0.130	0.211	0.422	0.495
ResNeXt101324d	0.390	0.594	0.522	0.476	0.349	0.141	0.226	0.434	0.509
ResNet18*	0.323	0.497	0.458	0.390	0.288	0.111	0.164	0.350	0.433
ResNet50*	0.370	0.558	0.516	0.450	0.338	0.141	0.210	0.408	0.483
ResNet101*	0.392	0.580	0.539	0.472	0.363	0.163	0.226	0.435	0.523
ResNeXt101324d*	0.410	0.601	0.561	0.492	0.382	0.175	0.235	0.452	0.539

Table 5. The results of balanced training method based on GGIoU. '*' means the method is used and the optimal parameters($\beta = 1/6$, $\alpha = 0.3$, $\gamma = 1.4$) from ablation experiments are set. RetinaNet are trained with the proposal method with the image scale of [800, 1333] on COCO *train-2017* and evaluated on *val-2017*.

Model	Backbone	Schedule	mAP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
YOLOv2 [16]	DarkNet-19	-	21.6	44.0	19.2	5.0	22.4	35.5
YOLOv3 [17]	DarkNet-53	-	33.0	57.9	34.4	18.3	35.4	41.9
SSD300 [14]	VGG16	-	23.2	42.1	23.4	5.3	23.2	39.6
SSD512 [14]	VGG16	-	26.8	46.5	27.8	9.0	28.9	41.9
Faster R-CNN [18]	ResNet-101-FPN	-	36.2	59.1	39.0	18.2	39.0	48.2
Deformable R-FCN [5]	Inception-ResNet-v2	-	37.5	58.0	40.8	19.4	40.1	52.5
Mask R-CNN [9]	ResNet-101-FPN	-	38.2	60.3	41.7	20.1	41.1	50.2
Faster R-CNN*	ResNet-50-FPN	1x	36.2	58.5	38.9	21.0	38.9	45.3
Faster R-CNN*	ResNet-101-FPN	1x	38.8	60.9	42.1	22.6	42.4	48.5
SSD300*	VGG16	120e	25.7	44.2	26.4	7.0	27.1	41.5
SSD512*	VGG16	120e	29.6	49.5	31.2	11.7	33.0	44.2
RetinaNet*	ResNet-18-FPN	1x	31.5	50.5	33.2	16.6	33.6	40.1
RetinaNet*	ResNet-50-FPN	1x	35.9	55.8	38.4	19.9	38.8	45.0
RetinaNet*	ResNet-101-FPN	1x	38.1	58.5	40.8	21.2	41.5	48.2
RetinaNet*	ResNeXt-32x4d-101-FPN	1x	39.4	60.2	42.3	22.5	42.8	49.8
GGIoU SSD300	VGG16	120e	27.2	45.2	28.6	8.2	29.4	43.1
GGIoU SSD512	VGG16	120e	30.9	50.1	32.9	12.6	34.5	45.1
GGIoU RetinaNet	ResNet-18-FPN	1x	32.8	50.6	35.3	17.7	34.6	41.9
GGIoU RetinaNet	ResNet-50-FPN	1x	37.3	56.1	40.1	20.9	40.1	46.4
GGIoU RetinaNet	ResNet-101-FPN	1x	39.5	58.5	42.7	22.1	42.7	50.2
GGIoU RetinaNet	ResNeXt-32x4d-101-FPN	1x	40.9	60.3	44.4	23.6	44.0	51.6

Table 6. Comparisons of different models for accurate object detection on MS-COCO. All models are trained on COCO *train-2017* and evaluated on COCO *test-dev*. The symbol '*' means the reimplemented results in MMDetection[2]. The training schedule "1x" and "120e" means the model is trained for 12 epochs and 120 epochs respectively.

calization function. As shown in Table 4, using IoU to balance localization function with best controlling parameter can improve the AP by 0.8%. Through comparison, adopting GGIoU can further boost performance by 0.3% which verifies the analyse in Section 3.3. The improved metric not only inherits the merit of balanced localization function which relieves the learning gradient is dominated by abnormal and noisy outliers, but pays more attention to features equipped with receptive field aligned better with objects, which is conducive to coordinate regression.

4.4. Experiments for GGIoU-balanced learning Method.

We combine the sampling strategy and balanced localization function based on GGIoU and call this union method as GGIoU-balanced training method. Table 5 reports that the union method yields considerable performance by 1.4% \sim 2.0% on RetinaNet with different backbones which reveals that the two proposal method based GGIoU can complement each other and shows the superiority of GGIoU. Meanwhile, the combined method consistently boosts the performance of $AP_{50} \sim AP_{60}$ by

Model	Backbone	mAP	AP ₅₀	AP ₆₀	AP ₇₀	AP ₈₀	AP ₉₀
RetinaNet	ResNet-18-FPN	47.5	74.9	69.7	58.3	39.9	13.9
RetinaNet	ResNet-50-FPN	52.3	79.2	74.5	64.2	45.8	17.4
RetinaNet	ResNet-101-FPN	54.5	79.7	75.8	66.2	50.7	21.0
RetinaNet	ResNeXt-32x4d-101-FPN	56.0	80.8	77.1	67.4	52.3	23.0
GGIoU RetinaNet	ResNet-18-FPN	49.4	74.4	70.5	60.2	43.9	17.3
GGIoU RetinaNet	ResNet-50-FPN	54.2	78.7	74.9	65.9	49.8	21.6
GGIoU RetinaNet	ResNet-101-FPN	56.4	79.9	76.4	68.3	53.0	24.6
GGIoU RetinaNet	ResNeXt-32x4d-101-FPN	58.0	80.7	77.4	69.8	54.9	28.0

Table 7. The results of RetinaNet are trained on the union of *VOC2007 trainval* and *VOC2012 trainval* and tested on *VOC2007 test* with the image scale of [600, 1000] and different backbones.

Backbone	mAP	AP ₅₀	AP ₅₅	AP ₆₀	AP ₆₅	AP ₇₀	AP ₇₅	AP ₈₀	AP ₈₅	AP ₉₀	AP ₉₅
ResNet18	78.0	94.6	93.8	92.8	91.5	89.3	86.4	82.3	75.4	59.6	14.0
ResNet50	80.7	95.4	95.0	94.2	93.0	91.3	88.7	84.9	78.6	65.2	21.0
ResNet101	82.7	96.0	95.4	94.9	93.8	92.2	90.1	87.1	81.8	70.0	25.9
ResNeXt-32x4d-101	83.2	96.2	95.8	95.1	94.2	92.7	90.4	87.3	82.8	70.7	26.4
ResNet18*	81.2	95.2	94.8	93.9	92.8	91.0	88.6	85.0	79.7	67.1	23.7
ResNet50*	84.0	96.2	95.6	95.0	94.2	92.6	90.6	87.6	83.2	72.8	32.6
ResNet101*	85.6	96.6	95.8	95.6	94.7	93.6	91.8	89.4	85.1	76.2	37.3
ResNeXt-32x4d-101*	86.1	96.8	96.5	95.9	95.0	94.0	92.0	89.9	85.7	76.4	38.4

Table 8. The main results of RetinaNet is trained on *SCD-train* and tested on *SCD-test* with the image scale of [800, 1333] and different backbones.

0.1% \sim 0.9% AP and $AP_{70} \sim AP_{90}$ by 1.2% \sim 3.4% AP which affirms it is versatile. Finally, all scale of backbones embedded into RetinaNet with our method achieve decent gains of performance that shows great robustness. Albeit notable gains by ResNeXt, these is still a powerful boost by 2.0%.

4.5. Main Results

To reveal the versatility and generalization of GGIoU-balanced learning strategy on different scenarios, we conduct experiments on general object detection dataset such as MS COCO and PASCAL VOC, as well as a large-scale stacked carton dataset such as SCD.

Main results on MS COCO. Table 6 reports the results of GGIoU-balanced learning strategy on COCO *test-dev* compared with some advanced detectors. The proposal method embedded into RetinaNet can improve the AP by 1.3% \sim 1.5% while significant gains are achieved in AP_{75} by 1.7% \sim 2.1%. As for SSD counterpart, 1.3% \sim 1.5% improvements are also gained in AP and 1.7% \sim 2.2% increases are yielded in AP_{75} . Therefore, it is concluded that anchor-based single shot detector equipped with the strategy method can yield decent improvements in challenging general object detection scenarios.

Main results on PASCAL VOC. To verify the generalization on general object detection dataset, we train RetinaNet with our method on PASCAL VOC and compare

with the baseline. As shown in Table 7, our proposal method brings 1.9% \sim 2.0% improvements in AP and 2.3% \sim 5.0% increases in AP_{80}/AP_{90} . The gain effects are the same as MS COCO, which indicates that the proposal approach can work well in different general object detection dataset scenarios.

Main results on SCD. We conduct experiments on SCD so as to show the performance on industrial application scenarios. RetinaNet fed the image scale of [800, 1333] with different backbones are trained on *LSCD-train* and tested on *LSCD-test* with the same image scale of [800, 1333]. As shown in Table 8, significant gains are achieved by 2.9% \sim 3.3% in AP. The localization accuracy boost are detailed in Figure 3, the accuracy is continuously boost with the increase of the AP threshold from 0.65% to 11.175%. It indicates that the method can be applied to industrial scenarios without costing inference overhead.

5. Conclusion

IoU lacks the ability to perceive the center distance between ground truths and the bounding box of objects, which damages the performance of anchor-based detectors during training. In this paper, we introduce a new metric, namely GGIoU, which can pay more attention to the center distance among bounding boxes. Based on GGIoU, a balanced training method is proposed, consisting of GGIoU-based sam-

pling strategy and GGIoU-balanced localization function. The union method can not only ease the dilemma of slender objects which are only assigned one positive sample, but guide model pay more attention to the feature whose receptive field better align with the region of objects. The property help improve localization accuracy and general performance. The proposed method shows powerful effect and generalization on general object detection dataset such as MS COCO and PASCAL VOC as well as large scale industrial dataset such as SCD.

References

- [1] 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*, pages 6154–6162, 2018. 3
- [2] 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. 5, 7
- [3] Y. Chen, C. Han, N. Wang, and Z. Zhang. Revisiting feature alignment for one-stage object detection. *arXiv preprint arXiv:1908.01570*, 2019. 3
- [4] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 3
- [5] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 7
- [6] 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*, 88(2):303–338, 2010. 2, 5
- [7] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017. 1
- [8] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 5
- [9] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3, 7
- [10] W. Ke, T. Zhang, Z. Huang, Q. Ye, J. Liu, and D. Huang. Multiple anchor learning for visual object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10206–10215, 2020. 3
- [11] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, and J. Shi. Foveabox: Beyond anchor-based object detection. *IEEE Transactions on Image Processing*, 29:7389–7398, 2020. 1
- [12] 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*, pages 2980–2988, 2017. 1, 3
- [13] 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*, pages 740–755. Springer, 2014. 2, 5
- [14] 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*, pages 21–37. Springer, 2016. 1, 3, 7
- [15] 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*, pages 779–788, 2016. 1
- [16] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1, 7
- [17] J. Redmon and A. Farhadi. Yolo3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 7
- [18] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 1, 3, 5, 7
- [19] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. 2
- [20] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636, 2019. 1
- [21] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin. Region proposal by guided anchoring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2974, 2019. 3
- [22] S. Wu, Y. Jinrong, X. Wang, and X. Li. Iou-balanced loss functions for single-stage object detection. *arXiv preprint arXiv:1908.05641*, 2019. 4
- [23] J. Yang, S. Wu, L. Gou, H. Yu, C. Lin, J. Wang, M. Li, and X. Li. Scd: A stacked carton dataset for detection and segmentation. *arXiv preprint arXiv:2102.12808*, 2021. 5
- [24] 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*, pages 516–520, 2016. 1, 2
- [25] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9759–9768, 2020. 3
- [26] X. Zhang, F. Wan, C. Liu, X. Ji, and Q. Ye. Learning to match anchors for visual object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3
- [27] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12993–13000, 2020. 2

- [28] X. Zhu, H. Hu, S. Lin, and J. Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 3