Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression

Hamid Rezatofighi[1,2] Nathan Tsoi[1] JunYoung Gwak[1] Amir Sadeghian[1,3]
Ian Reid[2] Silvio Savarese[1]

[1]Computer Science Department, Stanford University, United states
[2]School of Computer Science, The University of Adelaide, Australia
[3]Aibee Inc, USA
hamidrt@stanford.edu

Abstract

Intersection over Union (IoU) is the most popular evaluation metric used in the object detection benchmarks. However, there is a gap between optimizing the commonly used distance losses for regressing the parameters of a bounding box and maximizing this metric value. The optimal objective for a metric is the metric itself. In the case of axis-aligned 2D bounding boxes, it can be shown that IoU can be directly used as a regression loss. However, IoU has a plateau making it infeasible to optimize in the case of non-overlapping bounding boxes. In this paper, we address the weaknesses of IoU by introducing a generalized version as both a new loss and a new metric. By incorporating this generalized IoU ($GIoU$) as a loss into the state-of-the art object detection frameworks, we show a consistent improvement on their performance using both the standard, IoU based, and new, $GIoU$ based, performance measures on popular object detection benchmarks such as PASCAL VOC and MS COCO.

1. Introduction

Bounding box regression is one of the most fundamental components in many 2D/3D computer vision tasks. Tasks such as object localization, multiple object detection, object tracking and instance level segmentation rely on accurate bounding box regression. The dominant trend for improving performance of applications utilizing deep neural networks is to propose either a better architecture backbone [15, 13] or a better strategy to extract reliable local features [6]. However, one opportunity for improvement that is widely ignored is the replacement of the surrogate regression losses such as ℓ_1 and ℓ_2-norms, with a metric loss calculated based on Intersection over Union (IoU).

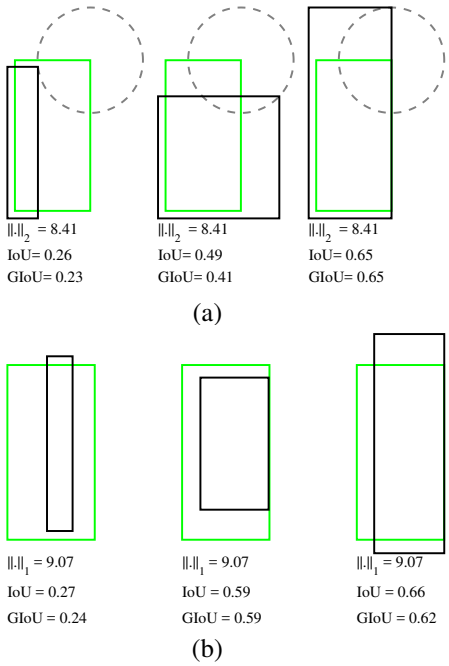


Figure 1. Two sets of examples (a) and (b) with the bounding boxes represented by (a) two corners (x_1, y_1, x_2, y_2) and (b) center and size (x_c, y_c, w, h). For all three cases in each set (a) ℓ_2-norm distance, $||.||_2$, and (b) ℓ_1-norm distance, $||.||_1$, between the representation of two rectangles are exactly same value, but their IoU and $GIoU$ values are very different.

IoU, also known as Jaccard index, is the most commonly used metric for comparing the similarity between two arbitrary shapes. IoU encodes the shape properties of the objects under comparison, *e.g.* the widths, heights and locations of two bounding boxes, into the region property and then calculates a normalized measure that focuses on their

areas (or volumes). This property makes IoU *invariant to the scale* of the problem under consideration. Due to this appealing property, all performance measures used to evaluate for segmentation [2, 1, 25, 14], object detection [14, 4], and tracking [11, 10] rely on this metric.

However, it can be shown that there is not a strong correlation between minimizing the commonly used losses, *e.g.* ℓ_n-norms, defined on parametric representation of two bounding boxes in 2D/3D and improving their IoU values. For example, consider the simple 2D scenario in Fig. 1 (a), where the predicted bounding box (black rectangle), and the ground truth box (green rectangle), are represented by their top-left and bottom-right corners, *i.e.* (x_1, y_1, x_2, y_2). For simplicity, let's assume that the distance, *e.g.* ℓ_2-norm, between one of the corners of two boxes is fixed. Therefore any predicted bounding box where the second corner lies on a circle with a fixed radius centered on the second corner of the green rectangle (shown by a gray dashed line circle) will have exactly the same ℓ_2-norm distance from the ground truth box; however their IoU values can be significantly different (Fig. 1 (a)). The same argument can be extended to any other representation and loss, *e.g.* Fig. 1 (b). It is intuitive that a good local optimum for these types of objectives may not necessarily be a local optimum for IoU. Moreover, in contrast to IoU, ℓ_n-norm objectives defined based on the aforementioned parametric representations are not invariant to the scale of the problem. To this end, several pairs of bounding boxes with the same level of overlap, but different scales due to *e.g.* perspective, will have different objective values. In addition, some representations may suffer from lack of regularization between the different types of parameters used for the representation. For example, in the center and size representation, (x_c, y_c) is defined on the location space while (w, h) belongs to the size space. Complexity increases as more parameters are incorporated, *e.g.* rotation, or when adding more dimensions to the problem. To alleviate some of the aforementioned problems, state-of-the-art object detectors introduce the concept of an anchor box [22] as a hypothetically good initial guess. They also define a non-linear representation [19, 5] to naively compensate for the scale changes. Even with these handcrafted changes, there is still a gap between optimizing the regression losses and IoU values.

In this paper, we explore the calculation of IoU between two axis aligned rectangles, or generally two axis aligned n-orthotopes, which has a straightforward analytical solution and in contrast to the prevailing belief, IoU in this case can be backpropagated [24], *i.e.* it can be directly used as the objective function to optimize. It is therefore preferable to use IoU as the objective function for 2D object detection tasks. Given the choice between optimizing a metric itself vs. a surrogate loss function, the optimal choice is the metric itself. However, IoU as both a metric and a loss has a major issue: if two objects do not overlap, the IoU value will be zero and will not reflect how far the two shapes are from each other. In this case of non-overlapping objects, if IoU is used as a loss, its gradient will be zero and cannot be optimized.

In this paper, we will address this weakness of IoU by extending the concept to non-overlapping cases. We ensure this generalization (a) follows the same definition as IoU, *i.e.* encoding the shape properties of the compared objects into the region property; (b) maintains the scale invariant property of IoU, and (c) ensures a strong correlation with IoU in the case of overlapping objects. We introduce this generalized version of IoU, named $GIoU$, as a new metric for comparing any two arbitrary shapes. We also provide an analytical solution for calculating $GIoU$ between two axis aligned rectangles, allowing it to be used as a loss in this case. Incorporating $GIoU$ loss into state-of-the art object detection algorithms, we consistently improve their performance on popular object detection benchmarks such as PASCAL VOC [4] and MS COCO [14] using both the standard, *i.e.* IoU based [4, 14], and the new, $GIoU$ based, performance measures.

The main contribution of the paper is summarized as follows:

- We introduce this generalized version of IoU, as a new metric for comparing any two arbitrary shapes.
- We provide an analytical solution for using $GIoU$ as loss between two axis-aligned rectangles or generally n-orthotopes[1].
- We incorporate $GIoU$ loss into the most popular object detection algorithms such as Faster R-CNN, Mask R-CNN and YOLO v3, and show their performance improvement on standard object detection benchmarks.

2. Related Work

Object detection accuracy measures: Intersection over Union (IoU) is the defacto evaluation metric used in object detection. It is used to determine true positives and false positives in a set of predictions. When using IoU as an evaluation metric an accuracy threshold must be chosen. For instance in the PASCAL VOC challenge [4], the widely reported detection accuracy measure, *i.e.* mean Average Precision (mAP), is calculated based on a fixed IoU threshold, *i.e.* 0.5. However, an arbitrary choice of the IoU threshold does not fully reflect the localization performance of different methods. Any localization accuracy higher than the threshold is treated equally. In order to make this performance measure less sensitive to the choice of IoU threshold, the MS COCO Benchmark challenge [14] averages mAP across multiple IoU thresholds.

[1] Extension provided in supp. material

Bounding box representations and losses: In 2D object detection, learning bounding box parameters is crucial. Various bounding box representations and losses have been proposed in the literature. Redmon *et al*. in YOLO v1[19] propose a direct regression on the bounding box parameters with a small tweak to predict square root of the bounding box size to remedy scale sensitivity. Girshick *et al*. [5] in R-CNN parameterize the bounding box representation by predicting location and size offsets from a prior bounding box calculated using a selective search algorithm [23]. To alleviate scale sensitivity of the representation, the bounding box size offsets are defined in log-space. Then, an ℓ_2-norm objective, also known as MSE loss, is used as the objective to optimize. Later, in Fast R-CNN [7], Girshick proposes ℓ_1-smooth loss to make the learning more robust against outliers. Ren *et al*. [22] propose the use of a set of dense prior bounding boxes, known as anchor boxes, followed by a regression to small variations on bounding box locations and sizes. However, this makes training the bounding box scores more difficult due to significant class imbalance between positive and negative samples. To mitigate this problem, the authors later introduce focal loss [13], which is orthogonal to the main focus of our paper.

Most popular object detectors [20, 21, 3, 12, 13, 16] utilize some combination of the bounding box representations and losses mentioned above. These considerable efforts have yielded significant improvement in object detection. We show there may be some opportunity for further improvement in localization with the use of $GIoU$, as their bounding box regression losses are not directly representative of the core evaluation metric, *i.e*. IoU.

Optimizing IoU using an approximate or a surrogate function: In the semantic segmentation task, there have been some efforts to optimize IoU using either an approximate function [18] or a surrogate loss [17]. Similarly, for the object detection task, recent works [8, 24] have attempted to directly or indirectly incorporate IoU to better perform bounding box regression. However, they suffer from either an approximation or a plateau which exist in optimizing IoU in non-overlapping cases. In this paper we address the weakness of IoU by introducing a generalized version of IoU, which is directly incorporated as a loss for the object detection problem.

3. Generalized Intersection over Union

Intersection over Union (IoU) for comparing similarity between two arbitrary shapes (volumes) $A, B \subseteq \mathbb{S} \in \mathbb{R}^n$ is attained by:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

Two appealing features, which make this similarity measure popular for evaluating many 2D/3D computer vision tasks are as follows:

Algorithm 1: Generalized Intersection over Union

input : Two arbitrary convex shapes: $A, B \subseteq \mathbb{S} \in \mathbb{R}^n$
output: $GIoU$

1 For A and B, find the smallest enclosing convex object C, where $C \subseteq \mathbb{S} \in \mathbb{R}^n$
2 $IoU = \frac{|A \cap B|}{|A \cup B|}$
3 $GIoU = IoU - \frac{|C \backslash (A \cup B)|}{|C|}$

- IoU as a distance, *e.g*. $\mathcal{L}_{IoU} = 1 - IoU$, is a metric (by mathematical definition) [9]. It means $\mathcal{L}_{IoU}$ fulfills all properties of a metric such as non-negativity, identity of indiscernibles, symmetry and triangle inequality.
- IoU is invariant to the scale of the problem. This means that the similarity between two arbitrary shapes A and B is independent from the scale of their space $\mathbb{S}$ (the proof is provided in supp. material).

However, IoU has a major weakness:

- If $|A \cap B| = 0$, $IoU(A, B) = 0$. In this case, IoU does not reflect if two shapes are in vicinity of each other or very far from each other.

To address this issue, we propose a general extension to IoU, namely Generalized Intersection over Union $GIoU$.

For two arbitrary convex shapes (volumes) $A, B \subseteq \mathbb{S} \in \mathbb{R}^n$, we first find the smallest convex shapes $C \subseteq \mathbb{S} \in \mathbb{R}^n$ enclosing both A and B[2]. For comparing two specific types of geometric shapes, C can be from the same type. For example, two arbitrary ellipsoids, C could be the smallest ellipsoids enclosing them. Then we calculate a ratio between the volume (area) occupied by C excluding A and B and divide by the total volume (area) occupied by C. This represents a normalized measure that focuses on the empty volume (area) between A and B. Finally $GIoU$ is attained by subtracting this ratio from the IoU value. The calculation of $GIoU$ is summarized in Alg. 1.

$GIoU$ as a new metric has the following properties: [3]

1. Similar to IoU, $GIoU$ as a distance, *e.g*. $\mathcal{L}_{GIoU} = 1 - GIoU$, holding all properties of a metric such as non-negativity, identity of indiscernibles, symmetry and triangle inequality.
2. Similar to IoU, $GIoU$ is invariant to the scale of the problem.
3. $GIoU$ is always a lower bound for IoU, *i.e*. $\forall A, B \subseteq \mathbb{S}$ $GIoU(A, B) \leq IoU(A, B)$, and this lower bound becomes tighter when A and B have a stronger shape

[2] Extension to non-convex cases has been provided in supp. material.
[3] Their proof has been provided in supp. material.

similarity and proximity, *i.e.* $\lim_{A \to B} GIoU(A, B) = IoU(A, B)$.

4. $\forall A, B \subseteq \mathbb{S}$, $0 \leq IoU(A, B) \leq 1$, but $GIoU$ has a symmetric range, *i.e.* $\forall A, B \subseteq \mathbb{S}$, $-1 \leq GIoU(A, B) \leq 1$.

   I) Similar to IoU, the value 1 occurs only when two objects overlay perfectly, *i.e.* if $|A \cup B| = |A \cap B|$, then $GIoU = IoU = 1$

   II) $GIoU$ value asymptotically converges to -1 when the ratio between occupying regions of two shapes, $|A \cup B|$, and the volume (area) of the enclosing shape $|C|$ tends to zero, *i.e.* $\lim_{\frac{|A \cup B|}{|C|} \to 0} GIoU(A, B) = -1$.

In summary, this generalization keeps the major properties of IoU while rectifying its weakness. Therefore, $GIoU$ can be a proper substitute for IoU in all performance measures used in 2D/3D computer vision tasks. In this paper, we only focus on 2D object detection where we can easily derive an analytical solution for $GIoU$ to apply it as both metric and loss. The extension to non-axis aligned 3D cases is left as future work.

3.1. GIoU as Loss for Bounding Box Regression

So far, we introduced $GIoU$ as a metric for any two arbitrary shapes. However as is the case with IoU, there is no analytical solution for calculating intersection between two arbitrary shapes and/or for finding the smallest enclosing convex object for them.

Fortunately, for the 2D object detection task where the task is to compare two axis aligned bounding boxes, we can show that $GIoU$ has a straightforward solution. In this case, the intersection and the smallest enclosing objects both have rectangular shapes. It can be shown that the coordinates of their vertices are simply the coordinates of one of the two bounding boxes being compared, which can be attained by comparing each vertices' coordinates using *min* and *max* functions. To check if two bounding boxes overlap, a condition must also be checked. Therefore, we have an exact solution to calculate IoU and $GIoU$.

Since back-propagating *min*, *max* and piece-wise linear functions, *e.g.* *Relu*, are feasible, it can be shown that every component in Alg. 2 has a well-behaved derivative. Therefore, IoU or $GIoU$ can be directly used as a loss, *i.e.* $\mathcal{L}_{IoU}$ or $\mathcal{L}_{GIoU}$, for optimizing deep neural network based object detectors. In this case, we are directly optimizing a metric as loss, which is an optimal choice for the metric. However, in all non-overlapping cases, IoU has zero gradient, which affects both training quality and convergence rate. $GIoU$, in contrast, has a gradient in all possible cases, including non-overlapping situations. In addition, using property 3,

Algorithm 2: IoU and $GIoU$ as bounding box losses

input : Predicted B^p and ground truth B^g bounding box coordinates:
$B^p = (x_1^p, y_1^p, x_2^p, y_2^p)$, $B^g = (x_1^g, y_1^g, x_2^g, y_2^g)$.
output: $\mathcal{L}_{IoU}$, $\mathcal{L}_{GIoU}$.

For the predicted box B^p, ensuring $x_2^p > x_1^p$ and $y_2^p > y_1^p$:
$\hat{x}_1^p = \min(x_1^p, x_2^p)$, $\hat{x}_2^p = \max(x_1^p, x_2^p)$,
$\hat{y}_1^p = \min(y_1^p, y_2^p)$, $\hat{y}_2^p = \max(y_1^p, y_2^p)$.
Calculating area of B^g: $A^g = (x_2^g - x_1^g) \times (y_2^g - y_1^g)$.
Calculating area of B^p: $A^p = (\hat{x}_2^p - \hat{x}_1^p) \times (\hat{y}_2^p - \hat{y}_1^p)$.
Calculating intersection $\mathcal{I}$ between B^p and B^g:
$x_1^{\mathcal{I}} = \max(\hat{x}_1^p, x_1^g)$, $x_2^{\mathcal{I}} = \min(\hat{x}_2^p, x_2^g)$,
$y_1^{\mathcal{I}} = \max(\hat{y}_1^p, y_1^g)$, $y_2^{\mathcal{I}} = \min(\hat{y}_2^p, y_2^g)$,
$\mathcal{I} = \begin{cases} (x_2^{\mathcal{I}} - x_1^{\mathcal{I}}) \times (y_2^{\mathcal{I}} - y_1^{\mathcal{I}}) & \text{if } x_2^{\mathcal{I}} > x_1^{\mathcal{I}}, y_2^{\mathcal{I}} > y_1^{\mathcal{I}} \\ 0 & \text{otherwise.} \end{cases}$
Finding the coordinate of smallest enclosing box B^c:
$x_1^c = \min(\hat{x}_1^p, x_1^g)$, $x_2^c = \max(\hat{x}_2^p, x_2^g)$,
$y_1^c = \min(\hat{y}_1^p, y_1^g)$, $y_2^c = \max(\hat{y}_2^p, y_2^g)$.
Calculating area of B^c: $A^c = (x_2^c - x_1^c) \times (y_2^c - y_1^c)$.
$IoU = \frac{\mathcal{I}}{\mathcal{U}}$, where $\mathcal{U} = A^p + A^g - \mathcal{I}$.
$GIoU = IoU - \frac{A^c - \mathcal{U}}{A^c}$.
$\mathcal{L}_{IoU} = 1 - IoU$, $\mathcal{L}_{GIoU} = 1 - GIoU$.

we show that $GIoU$ has a strong correlation with IoU, especially in high IoU values. We also demonstrate this correlation qualitatively in Fig. 2 by taking over 10K random samples from the coordinates of two 2D rectangles. In Fig. 2, we also observe that in the case of low overlap, *e.g.* $IoU \leq 0.2$ and $GIoU \leq 0.2$, $GIoU$ has the opportunity to change more dramatically compared to IoU. To this end, $GIoU$ can potentially have a steeper gradient in any possible state in these cases compared to IoU. Therefore, optimizing $GIoU$ as loss, $\mathcal{L}_{GIoU}$ can be a better choice compared to $\mathcal{L}_{IoU}$, no matter which IoU-based performance measure is ultimately used. Our experimental results verify this claim.

Loss Stability: We also investigate if there exist any extreme cases which make the loss unstable/undefined given any value for the predicted outputs.

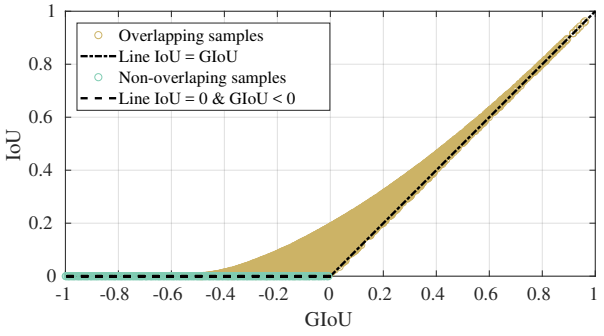


Figure 2. Correlation between GIoU and IOU for overlapping and non-overlapping samples.

Considering the ground truth bounding box, B^g is a rectangle with area bigger than zero, *i.e.* $A^g > 0$. Alg. 2 (**1**) and the Conditions in Alg. 2 (**4**) respectively ensure the predicted area A^p and intersection $\mathcal{I}$ are non-negative values, *i.e.* $A^p \geq 0$ and $\mathcal{I} \geq 0 \; \forall B^p \in \mathbb{R}^4$. Therefore union $\mathcal{U} > 0$ for any predicted value of $B^p = (x_1^p, y_1^p, x_2^p, y_2^p) \in \mathbb{R}^4$. This ensures that the denominator in IoU cannot be zero for any predicted value of outputs. Moreover, for any values of $B^p = (x_1^p, y_1^p, x_2^p, y_2^p) \in \mathbb{R}^4$, union is always bigger than intersection, *i.e.* $\mathcal{U} \geq \mathcal{I}$. Consequently, $\mathcal{L}_{IoU}$ is always bounded, *i.e.* $0 \leq \mathcal{L}_{IoU} \leq 1 \; \forall B^p \in \mathbb{R}^4$.

To check the stability of $\mathcal{L}_{GIoU}$, the extra term, *i.e.* $\frac{A^c - \mathcal{U}}{A^c}$, should always be a defined and bounded value. It can be easily perceived that the smallest enclosing box B^c cannot be smaller than B^g for all predicted values. Therefore the denominator in $\frac{A^c - \mathcal{U}}{A^c}$ is always a positive non-zero value, because $A^c \geq A^g \; \forall B^p \in \mathbb{R}^4$ and $A^g \geq 0$. Moreover, the area of the smallest enclosing box cannot be smaller than union for any value of predictions, *i.e.* $A^c \geq \mathcal{U}$ $\forall B^p \in \mathbb{R}^4$. Therefore, the extra term in $GIoU$ is positive and bounded. Consequently, $\mathcal{L}_{GIoU}$ is always bounded, *i.e.* $0 \leq \mathcal{L}_{GIoU} \leq 2 \; \forall B^p \in \mathbb{R}^4$.

$\mathcal{L}_{GIoU}$ behaviour when IoU = 0: For $GIoU$ loss, we have $\mathcal{L}_{GIoU} = 1 - GIoU = 1 + \frac{A^c - \mathcal{U}}{A^c} - IoU$. In the case when B^g and B^p do not overlap, *i.e.* $\mathcal{I} = 0$ and $IoU = 0$, $GIoU$ loss simplifies to $\mathcal{L}_{GIoU} = 1 + \frac{A^c - \mathcal{U}}{A^c} = 2 - \frac{\mathcal{U}}{A^c}$. In this case, by minimizing $\mathcal{L}_{GIoU}$, we actually maximize the term $\frac{\mathcal{U}}{A^c}$. This term is a normalized measure between 0 and 1, *i.e.* $0 \leq \frac{\mathcal{U}}{A^c} \leq 1$, and is maximized when the area of the smallest enclosing box A^c is minimized while the union $\mathcal{U} = A^g + A^p$, or more precisely the area of predicted bounding box A^p, is maximized. To accomplish this, the vertices of the predicted bounding box B^p should move in a direction that encourages B^g and B^p to overlap, making $IoU \neq 0$.

4. Experimental Results

We evaluate our new bounding box regression loss $\mathcal{L}_{GIoU}$ by incorporating it into the most popular 2D object detectors such as Faster R-CNN [22], Mask R-CNN [6] and YOLO v3 [21]. To this end, we replace their default regression losses with $\mathcal{L}_{GIoU}$, *i.e.* we replace ℓ_1-smooth in Faster /Mask-RCNN [22, 6] and MSE in YOLO v3 [21]. We also compare the baseline losses against $\mathcal{L}_{IoU}$[4].

Dataset. We train all detection baselines and report all the results on two standard object detection benchmarks, *i.e.* the PASCAL VOC [4] and the Microsoft Common Objects in Context (MS COCO) [14] challenges. The details of their training protocol and their evaluation have been provided in their own sections.

PASCAL VOC 2007: The Pascal Visual Object Classes (VOC) [4] benchmark is one of the most widely used datasets for classification, object detection and semantic segmentation. It consists of 9963 images with a 50/50 split for training and test, where objects from 20 pre-defined categories have been annotated with bounding boxes.

MS COCO: Another popular benchmark for image captioning, recognition, detection and segmentation is the more recent Microsoft Common Objects in Context (MS-COCO) [14]. The COCO dataset consists of over 200,000 images across train, validation and test sets with over 500,000 annotated object instances from 80 categories.

Evaluation protocol. In this paper, we adopt the same performance measure as the MS COCO 2018 Challenge [14] to report all our results. This includes the calculation of mean Average precision (mAP) over different class labels for a specific value of IoU threshold in order to determine true positives and false positives. The main performance measure used in this benchmark is shown by **AP**, which is averaging mAP across different value of IoU thresholds, *i.e.* $IoU = \{.5, .55, \cdots, .95\}$. Additionally, we modify this evaluation script to use $GIoU$ instead of IoU as a metric to decide about true positives and false positives. Therefore, we report another value for **AP** by averaging mAP across different values of $GIoU$ thresholds, $GIoU = \{.5, .55, \cdots, .95\}$. We also report the mAP value for IoU and $GIoU$ thresholds equal to 0.75, shown as **AP75** in the tables.

All detection baselines have also been evaluated using the test set of the MS COCO 2018 dataset, where the annotations are not accessible for the evaluation. Therefore in this case, we are only able to report results using the standard performance measure, *i.e.* IoU.

4.1. YOLO v3

Training protocol. We used the original Darknet implementation of YOLO v3 released by the authors [5]. For baseline results (training using MSE loss), we used DarkNet-608 as backbone network architecture in all experiments and followed exactly their training protocol using the reported default parameters and the number of iteration on each benchmark. To train YOLO v3 using IoU and $GIoU$ losses, we simply replace the bounding box regression MSE loss with $\mathcal{L}_{IoU}$ and $\mathcal{L}_{GIoU}$ losses explained in Alg. 2. Considering the additional MSE loss on classification and since we replace an unbounded distance loss such as MSE distance with a bounded distance, *e.g.* $\mathcal{L}_{IoU}$ or $\mathcal{L}_{GIoU}$, we need to regularize the new bounding box regression against the classification loss. However, we performed a very minimal effort to regularize these new regression losses against

[4] All source codes including the evaluation scripts, the training codes, trained models and all loss implementations in PyTorch, TensorFlow and darknet are available at: https://giou.stanford.edu.

[5] Available at: https://pjreddie.com/darknet/yolo/

Table 1. Comparison between the performance of **YOLO v3** [21] trained using its own loss (MSE) as well as $\mathcal{L}_{IoU}$ and $\mathcal{L}_{GIoU}$ losses. The results are reported on the **test set of PASCAL VOC 2007.**

Loss / Evaluation	AP		AP75	
	IoU	**GIoU**	**IoU**	**GIoU**
MSE [21]	.461	.451	.486	.467
$\mathcal{L}_{IoU}$	.466	.460	.504	.498
Relative improv %	1.08%	2.02%	3.70%	6.64%
$\mathcal{L}_{GIoU}$	**.477**	**.469**	**.513**	**.499**
Relative improv %	**3.45%**	**4.08%**	**5.56%**	**6.85%**

Table 2. Comparison between the performance of **YOLO v3** [21] trained using its own loss (MSE) as well as $\mathcal{L}_{IoU}$ and $\mathcal{L}_{GIoU}$ losses. The results are reported on 5K of the **2014 validation set of MS COCO**.

Loss / Evaluation	AP		AP75	
	IoU	**GIoU**	**IoU**	**GIoU**
MSE [21]	0.314	0.302	0.329	0.317
$\mathcal{L}_{IoU}$	0.322	0.313	0.345	0.335
Relative improv %	2.55%	3.64%	4.86%	5.68%
$\mathcal{L}_{GIoU}$	**0.335**	**0.325**	**0.359**	**0.348**
Relative improv %	**6.69%**	**7.62%**	**9.12%**	**9.78%**

Table 3. Comparison between the performance of **YOLO v3** [21] trained using its own loss (MSE) as well as using $\mathcal{L}_{IoU}$ and $\mathcal{L}_{GIoU}$ losses. The results are reported on the **test set of MS COCO 2018**.

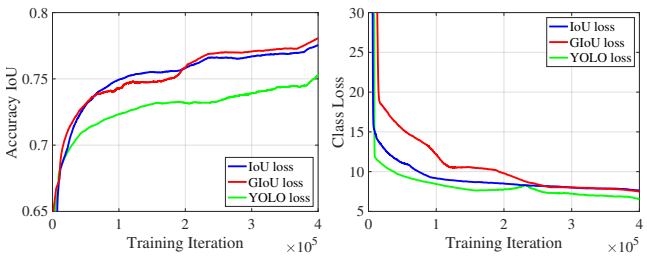
Loss / Evaluation	AP	AP75
MSE [21]	.314	.333
$\mathcal{L}_{IoU}$	.321	.348
Relative improv %	2.18%	4.31%
$\mathcal{L}_{GIoU}$	**.333**	**.362**
Relative improv %	**5.71%**	**8.01%**

the MSE classification loss.

PASCAL VOC 2007. Following the original code's training protocol, we trained the network using each loss on both training and validation set of the dataset up to $50K$ iterations. Their performance using the best network model for each loss has been evaluated using the PASCAL VOC 2007 test and the results have been reported in Tab. 1.

Considering both standard IoU based and new $GIoU$ based performance measures, the results in Tab. 1 show that training YOLO v3 using $\mathcal{L}_{GIoU}$ as regression loss can considerably improve its performance compared to its own regression loss (MSE). Moreover, incorporating $\mathcal{L}_{IoU}$ as regression loss can slightly improve the performance of YOLO v3 on this benchmark. However, the improvement is inferior compared to the case where it is trained by $\mathcal{L}_{GIoU}$.

MS COCO. Following the original code's training protocol, we trained YOLO v3 using each loss on both the training set and 88% of the validation set of MS COCO 2014 up to $502k$ iterations. Then we evaluated the results using the remaining 12% of the validation set and reported the results in Tab. 2. We also compared them on the MS COCO 2018 Challenge by submitting the results to the COCO server. All results using the IoU based performance measure are reported in Tab. 3. Similar to the PASCAL VOC experiment, the results show consistent improvement in performance for YOLO v3 when it is trained using $\mathcal{L}_{GIoU}$ as regression loss. We have also investigated how each component, *i.e.* bounding box regression and classification losses, contribute to the final AP performance measure. We believe the localization accuracy for YOLO v3 significantly improves when $\mathcal{L}_{GIoU}$ loss is used (Fig. 3 (a)). However, with the current naive tuning of regularization parameters, balancing bounding box loss vs. classification loss, the classification scores may not be optimal, compared to the baseline (Fig. 3 (b)). Since AP based performance measure is considerably affected by small classification error, we believe the results can be further improved with a better search for regularization parameters.



(a) (b)

Figure 3. The classification loss and accuracy (average IoU) against training iterations when YOLO v3 [21] was trained using its standard (MSE) loss as well as $\mathcal{L}_{IoU}$ and $\mathcal{L}_{GIoU}$ losses.

4.2. Faster R-CNN and Mask R-CNN

Training protocol. We used the latest PyTorch implementations of Faster R-CNN [22] and Mask R-CNN [6] [6], released by Facebook research. This code is analogous to the original Caffe2 implementation [7]. For baseline results (trained using ℓ_1-smooth), we used ResNet-50 the backbone network architecture for both Faster R-CNN and Mask R-CNN in all experiments and followed their training protocol using the reported default parameters and the number of iteration on each benchmark. To train Faster R-CNN and Mask R-CNN using IoU and $GIoU$ losses, we replaced their ℓ_1-smooth loss in the final bounding box refinement stage with $\mathcal{L}_{IoU}$ and $\mathcal{L}_{GIoU}$ losses explained in Alg. 2. Similar to the YOLO v3 experiment, we undertook minimal effort to regularize the new regression loss against the other losses such as classification and segmentation losses. We simply multiplied $\mathcal{L}_{IoU}$ and $\mathcal{L}_{GIoU}$ losses by a factor of 10 for all experiments.

PASCAL VOC 2007. Since there is no instance mask annotation available in this dataset, we did not evaluate Mask R-CNN on this dataset. Therefore, we only trained Faster R-CNN using the aforementioned bounding box re-

[6] https://github.com/roytseng-tw/Detectron.pytorch
[7] https://github.com/facebookresearch/Detectron

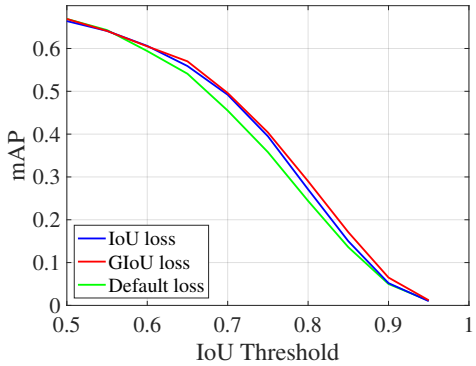


Figure 4. mAP value against different IoU thresholds, *i.e.* $.5 \leq IoU \leq .95$, for **Faster R-CNN** trained using ℓ_1-smooth (green), $\mathcal{L}_{IoU}$ (blue) and $\mathcal{L}_{GIoU}$ (red) losses.

gression losses on the training set of the dataset for 20k iterations. Then, we searched for the best-performing model on the validation set over different parameters such as the number of training iterations and bounding box regression loss regularizer. The final results on the test set of the dataset have been reported in Tab. 4.

According to both standard IoU based and new $GIoU$ based performance measure, the results in Tab. 4 show that training Faster R-CNN using $\mathcal{L}_{GIoU}$ as the bounding box regression loss can consistently improve its performance compared to its own regression loss (ℓ_1-smooth). Moreover, incorporating $\mathcal{L}_{IoU}$ as the regression loss can slightly improve the performance of Faster R-CNN on this benchmark. The improvement is inferior compared to the case where it is trained using $\mathcal{L}_{GIoU}$, see Fig. 4, where we visualized different values of mAP against different value of IoU thresholds, *i.e.* $.5 \leq IoU \leq .95$.

MS COCO. Similarly, we trained both Faster R-CNN and Mask R-CNN using each of the aforementioned bounding box regression losses on the MS COCO 2018 training dataset for 95K iterations. The results for the best model on the validation set of MS COCO 2018 for Faster R-CNN and Mask R-CNN have been reported in Tables 5 and 7 respectively. We have also compared them on the MS COCO 2018 Challenge by submitting their results to the COCO server. All results using the IoU based performance measure are also reported in Tables 6 and 8.

Table 4. Comparison between the performance of **Faster R-CNN** [22] trained using its own loss (ℓ_1-smooth) as well as $\mathcal{L}_{IoU}$ and $\mathcal{L}_{GIoU}$ losses. The results are reported on the **test set of PASCAL VOC 2007**.

Loss / Evaluation	AP		AP75	
	IoU	**GIoU**	**IoU**	**GIoU**
ℓ_1-smooth [22]	.370	.361	.358	.346
$\mathcal{L}_{IoU}$	.384	.375	.395	.382
Relative improv. %	3.78%	3.88%	10.34%	10.40%
$\mathcal{L}_{GIoU}$	**.392**	**.382**	**.404**	**.395**
Relative improv. %	**5.95%**	**5.82%**	**12.85%**	**14.16%**

Table 5. Comparison between the performance of **Faster R-CNN** [22] trained using its own loss (ℓ_1-smooth) as well as $\mathcal{L}_{IoU}$ and $\mathcal{L}_{GIoU}$ losses. The results are reported on the **validation set of MS COCO 2018.**

Loss / Evaluation	AP		AP75	
	IoU	**GIoU**	**IoU**	**GIoU**
ℓ_1-smooth [22]	.360	.351	.390	.379
$\mathcal{L}_{IoU}$	.368	.358	.396	.385
Relative improv.%	2.22%	1.99%	1.54%	1.58%
$\mathcal{L}_{GIoU}$	**.369**	**.360**	**.398**	**.388**
Relative improv. %	**2.50%**	**2.56%**	**2.05%**	**2.37%**

Table 6. Comparison between the performance of **Faster R-CNN** [22] trained using its own loss (ℓ_1-smooth) as well as $\mathcal{L}_{IoU}$ and $\mathcal{L}_{GIoU}$ losses. The results are reported on the **test set of MS COCO 2018**.

Loss / Metric	AP	AP75
ℓ_1-smooth [22]	.364	.392
$\mathcal{L}_{IoU}$	**.373**	.403
Relative improv.%	**2.47%**	2.81%
$\mathcal{L}_{GIoU}$	**.373**	**.404**
Relative improv.%	**2.47%**	**3.06%**

Table 7. Comparison between the performance of **Mask R-CNN** [6] trained using its own loss (ℓ_1-smooth) as well as $\mathcal{L}_{IoU}$ and $\mathcal{L}_{GIoU}$ losses. The results are reported on the **validation set of MS COCO 2018**.

Loss / Evaluation	AP		AP75	
	IoU	**GIoU**	**IoU**	**GIoU**
ℓ_1-smooth [6]	.366	.356	.397	.385
$\mathcal{L}_{IoU}$	.374	.364	.404	.393
Relative improv.%	2.19%	2.25%	1.76%	2.08%
$\mathcal{L}_{GIoU}$	**.376**	**.366**	**.405**	**.395**
Relative improv. %	**2.73%**	**2.81%**	**2.02%**	**2.60%**

Table 8. Comparison between the performance of **Mask R-CNN** [6] trained using its own loss (ℓ_1-smooth) as well as $\mathcal{L}_{IoU}$ and $\mathcal{L}_{GIoU}$ losses. The results are reported on the **test set of MS COCO 2018**.

Loss / Metric	AP	AP75
ℓ_1-smooth [6]	.368	.399
$\mathcal{L}_{IoU}$	**.377**	.408
Relative improv.%	**2.45%**	2.26%
$\mathcal{L}_{GIoU}$	**.377**	**.409**
Relative improv.%	**2.45%**	**2.51%**

Similar to the above experiments, detection accuracy improves by using $\mathcal{L}_{GIoU}$ as regression loss over ℓ_1-smooth [22, 6]. However, the amount of improvement between different losses is less than previous experiments. This may be due to several factors. First, the detection anchor boxes on Faster R-CNN [22] and Mask R-CNN [6] are more dense than YOLO v3 [21], resulting in less frequent scenarios where $\mathcal{L}_{GIoU}$ has an advantage over $\mathcal{L}_{IoU}$ such as non-overlapping bounding boxes. Second, the bounding box regularization parameter has been naively tuned on PASCAL VOC, leading to sub-optimal result on MS COCO [14].

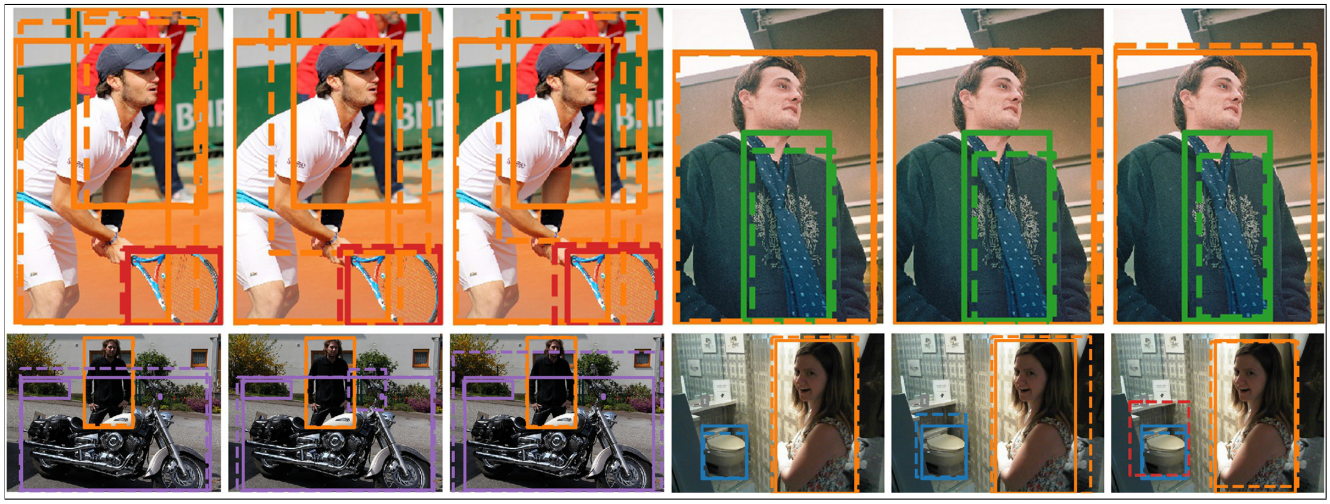


Figure 5. Example results from COCO validation using YOLO v3 [21] trained using (left to right) $\mathcal{L}_{GIoU}$, $\mathcal{L}_{IoU}$, and MSE losses. Ground truth is shown by a solid line and predictions are represented with dashed lines.

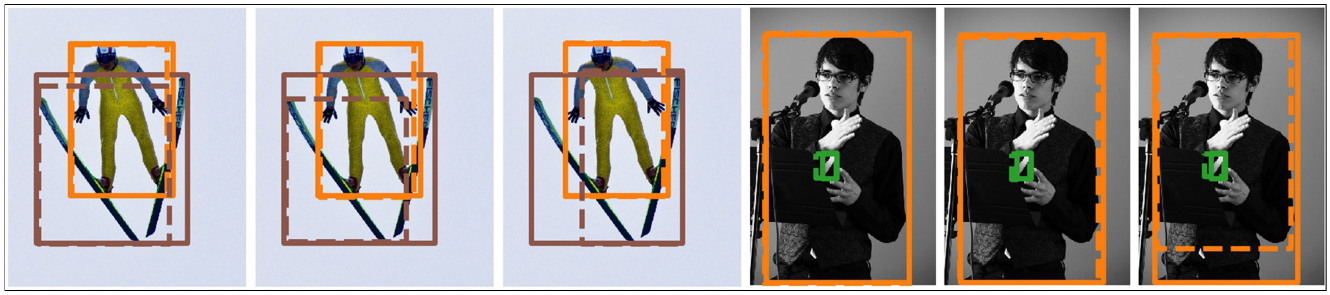


Figure 6. Two example results from COCO validation using Mask R-CNN [6] trained using (left to right) $\mathcal{L}_{GIoU}$, $\mathcal{L}_{IoU}$, ℓ_1-smooth losses. Ground truth is shown by a solid line and predictions are represented with dashed lines.

5. Conclusion

In this paper, we introduced a generalization to IoU as a new metric, namely $GIoU$, for comparing any two arbitrary shapes. We showed that this new metric has all of the appealing properties which IoU has while addressing its weakness. Therefore it can be a good alternative in all performance measures in 2D/3D vision tasks relying on the IoU metric.

We also provided an analytical solution for calculating $GIoU$ between two axis-aligned rectangles. We showed that the derivative of $GIoU$ as a distance can be computed and it can be used as a bounding box regression loss. By incorporating it into the state-of-the art object detection algorithms, we consistently improved their performance on popular object detection benchmarks such as PASCAL VOC and MS COCO using both the commonly used performance measures and also our new accuracy measure, *i.e.* $GIoU$ based average precision. Since the optimal loss for a metric is the metric itself, our $GIoU$ loss can be used as the optimal bounding box regression loss in all applications which require 2D bounding box regression.

In the future, we plan to investigate the feasibility of deriving an analytic solution for $GIoU$ in the case of two rotating rectangular cuboids. This extension and incorporating it as a loss could have great potential to improve the performance of 3D object detection frameworks.

References

[1] H. Alhaija, S. Mustikovela, L. Mescheder, A. Geiger, and C. Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision (IJCV)*, 2018. 2

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[3] 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*, pages 379–387, 2016. 3

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) chal-

lenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 2, 5

[5] 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*, pages 580–587, 2014. 2, 3

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 1, 5, 6, 7, 8

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[8] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings, European Conference on Computer Vision (ECCV) workshops*, 2018. 3

[9] S. Kosub. A note on the triangle inequality for the jaccard distance. *arXiv preprint arXiv:1612.02696*, 2016. 3

[10] M. Kristan and *et al.* The visual object tracking vot2016 challenge results. In *Proceedings, European Conference on Computer Vision (ECCV) workshops*, pages 777–823, 8Oct. 2016. 2

[11] L. Leal-Taixé, A. Milan, I. D. Reid, S. Roth, and K. Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *CoRR*, abs/1504.01942, 2015. 2

[12] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 936–944. IEEE, 2017. 3

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 1, 3

[14] 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, 7

[15] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018. 1

[16] 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. 3

[17] M. B. A. R. T. Matthew and B. Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[18] M. A. Rahman and Y. Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International Symposium on Visual Computing*, pages 234–244, 2016. 3

[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*, pages 779–788, 2016. 2, 3

[20] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016. 3

[21] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018. 3, 5, 6, 7, 8

[22] 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*, pages 91–99, 2015. 2, 3, 5, 6, 7

[23] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 3

[24] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang. Unitbox: An advanced object detection network. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 516–520. ACM, 2016. 2, 3

[25] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2