IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 December 2023, accepted 25 January 2024, date of publication 8 March 2024, date of current version 14 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3359433

RESEARCH ARTICLE

Fused-IoU Loss: Efficient Learning for Accurate Bounding Box Regression

YONG SUN, JIANZHONG WANG, HONGFENG WANG, SHENG ZHANG, YU YOU, ZIBO YU, AND YIGUO PENG

School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Jianzhong Wang (cwjzwang@bit.edu.cn)

This work was supported by the Defense Industrial Technology Development Program under Grant JCKY2021602B029.

ABSTRACT The importance of the loss function in object detection algorithms based on deep learning has grown significantly technological progress. The accuracy of object detection is significantly affected by bounding box regression, which is a crucial factor. Since the introduction of the Intersection over Union (IoU) loss in 2016, many improvements have been proposed based on this loss function. These studies considered various geometric factors related to bounding boxes, and constructed penalty terms to address this issue. This paper summarizes these functions and introduces a new Fused IoU (FIoU) loss function that leads to superior performance. The FIoU loss function not only solves the problem of gradient vanishing during the backpropagation process of the IoU loss function but also solves the problem of some IoU-based loss functions degenerating into IoU loss functions under certain conditions. In addition, in the simulation experiments, the FIoU loss function resulted in faster convergence speed. In our ablation experiments across different datasets and algorithms, our aim was to compare the mAP metrics under different loss functions. On the test set of the Pascal VOC dataset, employing the Faster R-CNN algorithm, FIoU demonstrated improvements of 1.1% and 1.7% over GIoU and Smoothℓ_1, respectively. With the YOLOX algorithm, FIoU outperformed GIoU and IoU by 1.0% and 0.8%. Utilizing the YOLOv7 algorithm, we evaluated seven loss functions, achieving optimal results with FIoU. On the validation set of the MS-COCO 2017 dataset, using YOLOv7 and YOLOv8, FIoU exhibited gains of 0.4%, 0.2%, 0.2% over EIoU, DIoU, GIoU, and 0.3%, 0.5%, 0.3% over EIoU, DIoU, GIoU, respectively.

INDEX TERMS Loss function, IoU, object detection, bounding box regression.

I. INTRODUCTION

Object detection, which has received considerable research attention, is a key issue in computer vision tasks. Current state-of-the-art object detection methods involve two basic tasks: object classification and object localization. Owing to the deformable Part Model [1], bounding box regression has been widely adopted for localization in object detection. With the significant progress made in deep learning, numerous deep models based on bounding box regression have been developed, including the YOLO series [2], [3], [4], [5], [6], [7],

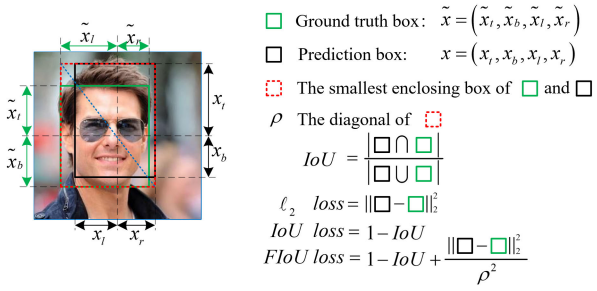


FIGURE 1. $\mathcal{L}_{l_2}$,$\mathcal{L}_{IoU}$ and $\mathcal{L}_{FIoU}$.

Faster R-CNN [8], Cascade R-CNN [9], and SSD [10]. Based on these models, a well-designed bounding box regression loss function is essential. Thus far, the majority of

The associate editor coordinating the review of this manuscript and approving it for publication was Tallha Akram.

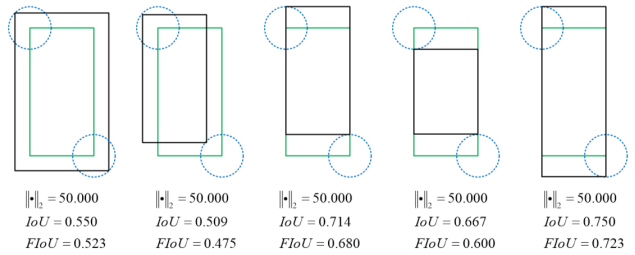


FIGURE 2. The examples with the bounding boxes represented by two corners(x_1, y_1, x_2, y_2).For all five cases in this set ℓ_2-norm distance,$||.||_2$, between the representation of two rectangles are the same value, but their IoU and FIoU values are very different.

bounding box regression loss functions can be divided into two main categories: ℓ_n loss functions and IoU(Intersection over Union)-based loss functions.

During the bounding box regression process, most of the existing loss functions have the same value in different situations. e.g. ℓ_2 loss, defined on the parametric representation of two bounding boxes in 2D/3D as shown in Fig. 1. When there is a different positional relationship between the predicted box and ground truth box, the same ℓ_2-norm distance is displayed. However, in this case, the intersection over the union is not the same. For example, consider the simple 2D scenario in Fig.2, where the predicted bounding box (black rectangle) and the ground truth box (green rectangle) are represented by their top-left and bottom-right corners, that is, (x_1, y_1, x_2, y_2). Any predicted bounding box where the corresponding corner lies on a circle with a fixed radius centered on the corner of the green rectangle (shown by a blue dashed line circle) will have the same ℓ_2-norm distance from the ground truth box; however, their IoU values are entirely different, as shown in Fig.2. In addition, we found that the IoU had scale invariance. Under the same overlap, regardless of how the two boxes were scaled, the IoU value is not affected. By contrast, the ℓ_2-norm is very sensitive to changes in scale. In Fig.3, the difference in ℓ_2-norm is significant for the same IoU. In addition, we selected a 4×4 ground truth box and an 8×8-sized ground truth box in the pixel dimension, as shown in Fig.4. When there is a pixel deviation in the width and height, although the value obtained by the ℓ_2-norm is the same, there are significant differences in the actual IoU, resulting in better prediction results for large targets and poorer prediction results for small targets.

Furthermore, some representations may suffer from a lack of regularization between the different parameter types used. For example, in Faster R-CNN [8], the width, height, and center point coordinates are used to represent the bounding box. The center point(x, y) coordinates are defined in the location space, whereas width w and height h are defined in the size space.

However, once proposed as a metric or loss function, the IoU faces two problems. If there is no overlap between the two objects or if there is an inclusion relationship between the two objects, IoU serves as the loss function with a derivative of zero. In this case, the IoU cannot further optimize the two objects and cannot reflect the positional relationship between the two objects, as shown in Fig.4.

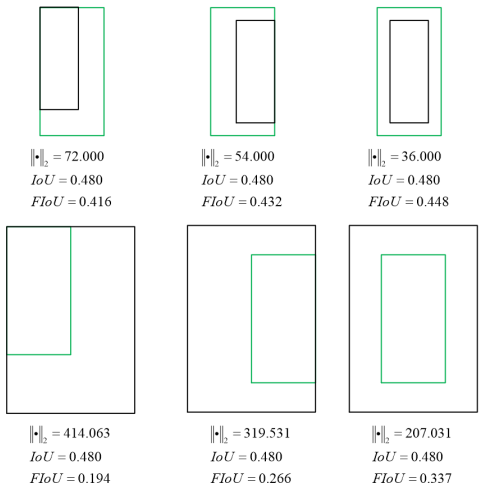


FIGURE 3. The examples with the bounding boxes represented by two corners(x_1, y_1, x_2, y_2).For all six cases in this set, IoU between the representation of two rectangles is the same value, but their FIoU values are very different.

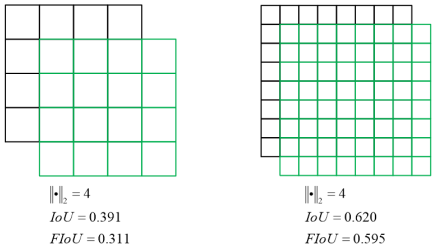


FIGURE 4. For ground truth boxes of different scales, the deviation between the predicted box and the ground truth box is only one pixel in the height and width directions. Although the l_2 loss is the same, the prediction effect is completely different.

With the concept of anchor boxes being proposed, anchor boxes obtained through clustering algorithms are pre-set with multiple aspect ratios. These can be represented as nonlinear using the truth box. An increasing number of IoU-based bounding box regression loss functions have been proposed; however, in the optimization process, there is still a significant gap between the predicted results and the truth box.

In this study, we address the weaknesses of several existing bounding box regression loss functions. Inspired by the geometric features of the horizontal rectangle, we explore a novel metric for bounding box regression called Fused Intersection over Union loss function $\mathcal{L}_{FIoU}$ based on ℓ_2 loss function and Intersection over Union loss function, which normalizes the ℓ_2-norm from each pair of predicted bounding box and ground truth bounding box by dividing the obtained ℓ_2-norm to the square of the diagonal of the smallest enclosing convex containing two boxes as shown in Fig. 1 and use the obtained result as the penalty term of the IoU loss function. We used FIoU as a new measure to compare the similarity between the predicted bounding box and ground truth bounding box in the bounding box regression process. During this process, we ensure that this metric maintains

the scale-invariant characteristics of the IoU and a strong correlation with the IoU in the case of overlapping objects. We also attempted to incorporate FIoU loss into state-of-the-art object detection algorithms such as Faster R-CNN [8], YOLOX [6], YOLOv7 [7], and YOLOv8, and tested it on mainstream object detection datasets Pascal Visual Object Classes (VOC) 2007 & 2012 [11], [12] and Microsoft Common Objects in Context (COCO) 2017 [13] to verify the performance of our proposed FIoU.

The main contribution of the paper is summarized as follows:

- We introduced *FIoU* as a metric that combines *IoU* with the normalized ℓ_2 norm to compare two arbitrary bounding boxes.
- We propose a new loss function, FIoU loss, which can solve the problem of gradient vanishing in traditional IoU loss in backpropagation. Through simulation experiments, We also verified that its convergence speed is faster than that of most existing IoU-based loss functions.
- We applied FIoU loss to popular object detection algorithms such as Faster R-CNN, YOLOX, YOLOv7, and YOLOv8, and demonstrated their performance improvement on standard object detection benchmarks Pascal VOC 2007 & 2012 and MS-COCO 2017.

II. RELATED WORK

A. OBJECT DETECTION

In recent years, deep learning-based object detection [14], [15], [16], [17] algorithms have emerged one after another, with the two most important issues being classification and localization. In many well-known classical algorithms, bounding box regression has become an essential component for defining localization loss functions [18]. In deep models for object detection, the R-CNN series [8], [9], [19] adopted two or three bounding box regression modules to obtain higher location accuracy, whereas the YOLO series [2], [3], [4], and SSD series [10], [20], [21] adopt one for faster inference speed. In RepPoints [22], a rectangular box is formed by predicting several points. FCOS [23] locates an object by predicting the distances from the sampling points to the top, bottom, left, and right sides of the ground-truth box.

B. LOSS FUNCTION FOR BOUNDING BOX REGRESSION

1) ℓ_N LOSS

- ℓ_1 loss

  ℓ_1 loss refers to the value of the absolute difference between the model predicted value x, and the ground truth value $\widetilde{x}$, and the formula is as follows:

$$\ell_1\ loss = \sum |x_i - \tilde{x}_i| \tag{1}$$

  In the process of calculating the ℓ_1 loss, regardless of whether the predicted value is close to the true value, the resulting gradient is constant, which can easily lead to solution divergence or missing extreme points. Therefore, Smoothℓ_1 loss is often used instead of traditionalℓ_1 loss.

- Smoothℓ_1 loss

$$Smooth\ell_1\ loss = \begin{cases} \dfrac{0.5(x_i - \tilde{x}_i)^2}{\beta} & if \quad |x_i - \tilde{x}_i| < \beta \\ |x_i - \tilde{x}_i| - 0.5\beta & otherwise \end{cases} \tag{2}$$

  β is usually set as 1. When the difference between the ground truth value and the predicted value is small (the absolute value difference is less than 1), the gradient will also be relatively small (the loss function is smoother than ordinaryℓ_1 loss here), The most famous Faster R-CNN network [8] uses this loss function in the process of bounding box regression.

- ℓ_2 loss

$$\ell_2\ loss = \sum (x_i - \tilde{x}_i)^2 \tag{3}$$

  Compared with the ℓ_1 loss, the ℓ_2 loss function is smooth and differentiable and can have a more stable solution without oscillation. Typical YOLOv1 [2], YOLOv2 [3], and YOLOv3 [4] algorithms use this loss function for bounding box regression. Because the parameters are not normalized, this regression method differs in sensitivity to large and small targets, leading to unsatisfactory results.

2) IOU LOSS

The intersection over union(IoU) [24], also known as the Jaccard similarity coefficient, is an index used to measure the degree of overlap between the predicted bounding box and ground truth bounding box in target detection tasks. It has two appealing features: IoU, as a type of distance, has a loss function (5), which can be used as a metric system and has attributes such as non-negativity, identity, symmetry, and triangle inequality; IoU is invariant to the scale, which means that regardless of the scale, two targets have in space, as long as their overlapping positional relationship remains unchanged, and their IoU is constant.

$$IoU = \frac{B^{gt} \cap B^{pred}}{B^{gt} \cup B^{pred}} \tag{4}$$

$$\mathcal{L}_{IoU} = 1 - IoU \tag{5}$$

where represented by green box B^{gt} denotes the ground truth bounding box, represented by the black box B^{pred} denotes the predicted bounding box in Fig. 5, and $\mathcal{L}$ represents loss function.

However, when the ground truth bounding box and predicted bounding box do not overlap, the value of IoU is zero. In this case, we cannot determine the relative position relationship between the predicted bounding box and the ground truth bounding box: adjacent or far apart. In addition, the IoU loss enters a plateau period, and there is no way to optimize it. Since then, variants based on IoU loss have emerged in an endless stream, such as DIoU, GIoU, and

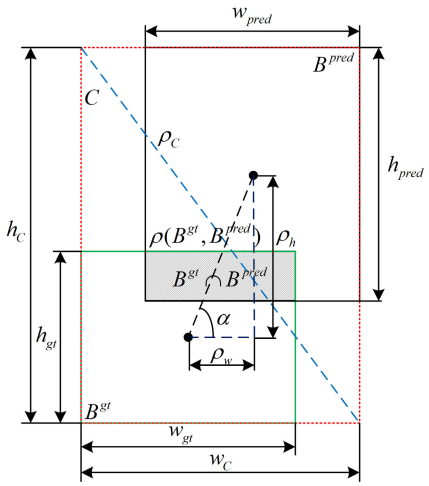


FIGURE 5. The relationship between ground truth bounding box and predicted bounding box, and the relevant parameters required to calculate the existing IoU-based metrics.

EIoU, which can be uniformly defined as penalty terms, and $\mathcal{R}$ is used to represent them in the function as follows:

$$\mathcal{L}_{IoU-based} = 1 - IoU + \mathcal{R}(x_i - \tilde{x}_i) \tag{6}$$

- GIoU loss

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|\ C - B^{gt} \cup B^{pred}\ |}{|\ C\ |} \tag{7}$$

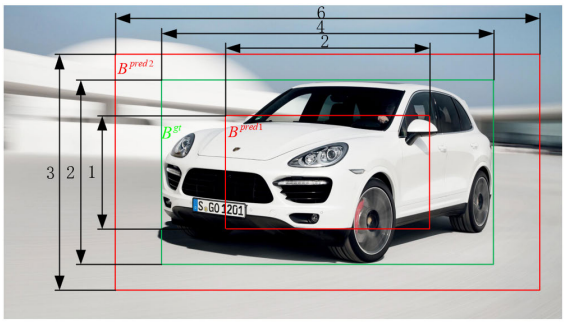
$IoU = 0$ is a fundamental issue that motivated the initial proposal of the GIoU [25]. (7), where C represents the smallest enclosing bounding box that includes both the predicted bounding box and ground truth bounding box, reflecting its penalty term. By calculating the ratio between the area occupied by C excluding B^{gt} and B^{pred} and dividing by the total area occupied by C, this penalty term primarily focuses on the parts that do not overlap between B^{gt} and B^{pred}. The goal was to reduce the non-overlapping areas between the two boxes by adjusting the predicted bounding box. However, GIoU degenerates into IoU if an inclusion relationship exists between the two boxes, such as $C = B^{gt}$ or $C = B^{pred}$.

- DIoU loss

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(B^{gt}, B^{pred})}{{\rho_C}^2} \tag{8}$$

According to DIoU [26], an important aspect in determining the overlap between two bounding boxes is the distance between their center points. The equation for the DIoU loss can be written as (8), where $\rho(B^{gt}, B^{pred})$ represent the Euclidean distance between the centers of the predicted bounding box and ground truth bounding box, and ρ_C is the diagonal length of the smallest enclosing rectangle, which corresponds to the black and blue dotted lines in Fig. 5, respectively.

We can see that the DIoU loss seeks to reduce the distance between the center points of the two bounding boxes; however, when the two points coincide, the penalty term is zero, and the loss function degenerates into IoU loss. The CIoU has emerged to address this issue.



$L^1_{IoU} = L^1_{GIoU} = L^1_{DIoU} = L^1_{CIoU} = 0.750,\ L^1_{FIoU} = 0.875$
$L^2_{IoU} = L^2_{GIoU} = L^2_{DIoU} = L^2_{CIoU} = 0.560,\ L^2_{FIoU} = 0.806$

FIGURE 6. CIoU, DIoU, and GIoU will all degenerate into IoU When the aspect ratio of the ground truth bounding box and the predicted bounding box is the same, and their centres overlap.

- CIoU loss

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(B^{gt}, B^{pred}))}{{\rho_C}^2} + \alpha V \tag{9}$$

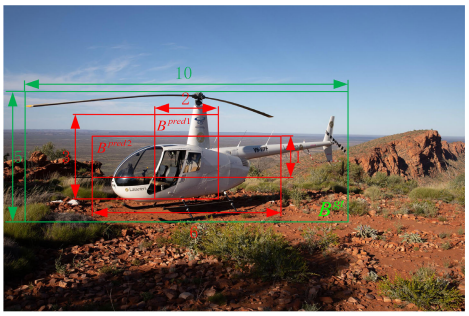
$$V = \frac{4}{\pi^2}(\arctan\frac{w_{gt}}{h_{gt}} - \arctan\frac{w_{pred}}{h_{pred}})^2 \tag{10}$$

$$\alpha = \begin{cases} 0, & \text{if} \quad IoU < 0.5, \\ \frac{V}{(1-IoU)+V}, & \text{if} \quad IoU \geq 0.5. \end{cases} \tag{11}$$

CIoU [27] introduced the concept of aspect ratio based on DIoU. However, we can analyze from formulas (9), (10), and (11) that when $IoU < 0.5$, CIoU degenerates into DIoU. The authors believe that it is reasonable that when two bounding boxes are not well matched, the consistency of the aspect ratio is less important, and when $IoU > 0.5$, the consistency of the aspect ratio becomes necessary. However, the definition of the aspect ratio from CIoU is a relative value rather than an absolute value. For example, as Fig. 6 shows, when the aspect ratio of the ground truth bounding box and the predicted bounding box are the same, and their centers overlap, we find that CIoU, DIoU, and GIoU degenerate into IoU. To address this issue, EIoU was proposed based on DIoU.

- EIoU loss

Although EIoU [28] used the ratio of the difference between the width and height of the predicted bounding box and the ground truth bounding box to the actual width and height of the smallest enclosing bounding box as the post-term of DIoU, which solve the problem of loss function degradation when CIoU utilizes the relative value of the aspect ratio in practical situations, as shown in Fig. 7, when the height difference and width difference meet a specific proportional relationship with the width and height of the minimum bounding box,



$L^1_{IoU} = L^2_{IoU} = 0.800,\ L^1_{GIoU} = L^2_{GIoU} = 0.800$
$L^1_{DIoU} = L^2_{DIoU} = 0.800,\ L^1_{EIoU} = L^2_{EIoU} = 1.600$
$L^1_{FIoU} = 1.072,\ L^2_{FIoU} = 0.928$

FIGURE 7. CIoU, DIoU, GIoU, and EIoU enjoy the same loss value in different situations, which leads to limited convergence speed and accuracy.

it will loss effectiveness.

$$\mathcal{L}_{EIoU} = 1 - IoU + \frac{\rho^2(B^{gt}, B^{pred})}{\rho_C{}^2} + \frac{\rho^2(w_{gt}, w_{pred})}{w_C{}^2} + \frac{\rho^2(h_{gt}, h_{pred})}{h_C{}^2} \tag{12}$$

- SIoU loss
The previous IoU-based loss function only considered factors such as the center distance and aspect ratio between the predicted bounding box and ground truth bounding box. Ignoring the matching of the direction between the two boxes leads to a slow convergence speed and low efficiency, because the predicted bounding box may "wandering" during the training process, ultimately generating a worse model. Therefore, Gevorgyan constructed SIoU [29] using the angle cost as (13), distance cost as (14) and (15), and shape cost as (16) and (17). These are combined in (18). The angle cost, which describes the minimum α shown in Fig. 5 between the $\rho(B^{gt}, B^{pred})$ and the (X or Y) axis, can quickly drift the prediction box to the nearest axis; then, only one coordinate (X or Y) needs to be regressed, effectively reducing the total number of degrees of freedom.

$$\Lambda = 1 - 2 \times \sin^2(\arcsin \frac{\min(\rho_h, \rho_w)}{\rho(B^{gt}, B^{pred})} - \frac{\pi}{4}) \tag{13}$$

$$\Delta = \sum_{t=w,h} (1 - e^{-\gamma k_t}) \tag{14}$$

where

$$\begin{cases} \gamma = 2 - \Lambda \\ k_w = \left(\frac{\rho_w}{w_C}\right)^2 \\ k_h = \left(\frac{\rho_h}{h_C}\right)^2 \end{cases} \tag{15}$$

$$\Omega = \frac{1}{2} \sum_{t=w,h} (1 - e^{\omega_t})^\theta \tag{16}$$

where

$$\begin{cases} \theta = 4 \\ \omega_w = \frac{|w^{pred} - w^{gt}|}{\max(w^{pred}, w^{gt})} \\ \omega_h = \frac{|h^{pred} - h^{gt}|}{\max(h^{pred}, h^{gt})} \end{cases} \tag{17}$$

$$\mathcal{L}_{SIoU} = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{18}$$

The distance cost indicates that the penalty of $\rho(B^{gt}, B^{pred})$ is positively correlated with angle cost. That is, the contribution of the distance cost is greatly reduced when α approaches 0 and increases when α approaches $\frac{\pi}{4}$. The shape cost describes the difference between the two bounding boxes, and is used to achieve the overall shape convergence effect by converging the length and width. Because the penalty of SIoU on the distance metric increases with an increase in the shape cost, the models trained by SIoU have a faster convergence speed and lower regression error.

- WIoU loss
Owing to the inevitable inclusion of low-quality examples in the training data, geometric factors such as distance and aspect ratio exacerbate the punishment for low-quality examples, thereby reducing the generalization performance of the model. When the predicted bounding box overlaps well with the ground truth bounding box, a good loss function should weaken the penalty of the geometric factors, whereas less training intervention will enable the model to achieve better generalization ability. The WIoU [30] is defined as (19).

$$\mathcal{L}_{WIoUv1} = \mathcal{R}_{WIoU} \mathcal{L}_{IoU}$$
$$\mathcal{R}_{WIoU} = \exp(\frac{\rho_w{}^2 + \rho_h{}^2}{(\rho_C{}^2)^*}) \tag{19}$$

III. FUSED INTERSECTION OVER UNION

A. FIOU AS A METRIC

Thus, we have gained an understanding of the advantages of ℓ_n loss, existing IoU-based loss functions, and problems exposed in different situations. We consider how to solve bounding box regression problems more robustly, accurately, and efficiently. Considering the advantages of existing algorithms, we were inspired by the geometric properties of ground truth bounding boxes, predicted bounding boxes, the smallest enclosing bounding boxes, and proposed FIoU. The calculation of the FIoU is summarized in Algorithm 1.

FIoU, as a new metric, has the following properties:

1) Similar to IoU, FIoU is a distance, for example, FIoU loss = 1-FIoU, holding all the properties of a metric, such as non-negativity, identity of indiscernibles, symmetry, and triangle inequality.
2) Similar to IoU, FIoU also exhibits scale invariance when two rectangular boxes have the same overlapping positional relationship.

Algorithm 1 Fused Intersection over Union

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

1. For A and B, (x_1^a, y_1^a), (x_2^a, y_2^a)denote the top-left and bottom-right point coordinates of A, (x_1^b, y_1^b), (x_2^b, y_2^b)denote the top-left and bottom-
right point coordinates of B.
2. Find the smallest enclosing convex object C, which can be denoted
by the top-left point coordinates(x_1^c, y_1^c), and bottom-right point coordinates(x_2^c, y_2^c):
$x_1^c = min(x_1^a, x_2^a), \quad y_1^c = min(y_1^a, y_2^a),$
$x_2^c = max(x_1^b, x_2^b), \quad y_2^c = max(y_1^b, y_2^b).$
3. Calculating the width and height of C:
$w = x_2^c - x_1^c, \quad h = y_2^c - y_1^c.$
4. Calculating the square of the diagonal of C:
$\rho^2 = w^2 + h^2.$
5. $l_2 = (x_1^b - x_1^a)^2 + (y_1^b - y_1^a)^2 + (x_2^b - x_2^a)^2 + (y_2^b - y_2^a)^2.$
6. $IoU = \frac{|A \cap B|}{|A \cup B|}.$
7. $FIoU = IoU - \frac{l_2}{\rho^2}.$

3) FIoU 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 more substantial shape similarity and proximity, *i.e.*, $\lim_{A \to B} FIoU(A, B) = IoU(A, B)$.
4) $\forall A, B \subseteq \mathbb{S}, 0 \leq IoU(A, B) \leq 1$, however, FIoU has a different range, *i.e.*, $\forall A, B \subseteq \mathbb{S}, -2 < FIoU(A, B) \leq 1$:
   - Similar to IoU, the value of 1 occurs only when two objects overlay perfectly, that is, if $|A \cup B| = |A \cap B|$, then FIoU = IoU = 1.
   - When C is much greater than A and B, *i.e.*, in this extreme case, the predicted bounding box and ground truth bounding box are located at the opposite corners of the image, and the height and width of both boxes are much smaller than the height and width of the image, we assume that the value of the ℓ_2-norm is approximate twice the square of the diagonal of image, with IoU=0, then FIoU=−2.

The proposed method retained the original IoU properties when combined with the ℓ_2-norm. Compared with many recently proposed IoU-based methods, our proposed method is more concise and efficient. Therefore, the FIoU can serve as a substitute for bounding box regression loss in the localization loss function in many 2D/3D object detection algorithms. In this study, we focused on 2D object detection. The extension to non-axis-aligned 3D cases is left for future work.

B. FIOU AS LOSS FOR BOUNDING BOX REGRESSION

In the early YOLO series [2], [3], [4], Faster R-CNN [8], *etc.*, each bounding box $B^{pred} = [x^{pred}, y^{pred}, w^{pred}, h^{pred}]^T$ predicted by these algorithms was forced to approach its ground truth bounding box $B^{gt} = [x^{gt}, y^{gt}, w^{gt}, h^{gt}]^T$ by minimizing the loss function as follows:

$$\mathcal{L} = \min_{\Theta} \sum_{B_{gt} \in \mathbb{B}_{gt}} \mathcal{L}(B_{gt}, B_{pred}|\Theta) \tag{20}$$

where $\mathbb{B}_{gt}$ is the set of ground truth bounding boxes and Θ is the parameter of the deep model for regression. Based on the definition of FIoU in the previous section, we define the loss function based on FIoU as follows:

$$\mathcal{L}_{FIoU} = 1 - IoU + \frac{l_2}{{\rho_C}^2} \tag{21}$$

Here, we used the following two sets of coordinates to represent the four regression parameters defined in the previous algorithm:

$$\begin{aligned}
w &= max(x_2^g, x_2^p) - min(x_1^g, x_1^p), \\
h &= max(y_2^g, y_2^p) - min(y_1^g, y_1^p), \\
x^{gt} &= \frac{x_1^g + x_2^g}{2}, y^{gt} = \frac{y_1^g + y_2^g}{2}, \\
w^{gt} &= x_2^g - x_1^g, h^{gt} = y_2^g - y_1^g, \\
x^{pred} &= \frac{x_1^p + x_2^p}{2}, y^{pred} = \frac{y_1^p + y_2^p}{2}, \\
w^{pred} &= x_2^p - x_1^p, h^{pred} = y_2^p - y_1^p.
\end{aligned} \tag{22}$$

where w and h represent the weight and height of the smallest enclosing bounding box covering B^{gt} and B^{pred}, (x^{gt}, y^{gt}) and (x^{pred}, y^{pred}) represent the coordinates of the central points of the ground truth bounding box and the predicted bounding box, respectively. w_{gt} and h_{gt} represent the width and height of the ground truth bounding box, respectively. w_{pred} and h_{pred} represent the width and height of the predicted bounding box, respectively.

From (22), we can see that all of the factors considered in the existing loss functions can be determined by the coordinates of the top-left points and the bottom-right points, such as the non-overlapping area, central point distance, and deviation of width and height, which means that our proposed FIoU not only considers but also simplifies the calculation process. Then the ℓ_2 loss, IoU loss, and FIoU loss can be obtained through Algorithm 2.

1) NORMALIZED MATHEMATICAL EXPRESSION

Compared to ℓ_2 loss, FIoU loss features a normalized mathematical expression. Initially, traditionalℓ_2 loss optimizes four mutually independent variables $(x^{pred}, y^{pred}, w^{pred}, h^{pred})$, distributed in both location and size spaces. This assumption contradicts the fact that the bounding box of an object is highly correlated, which leads to many failure cases where one or two boundaries of the predicted bounding box are very close to the ground truth bounding box, but the entire bounding box is unacceptable.

The situation depicted in Fig. 2 arises upon transforming these variables into diagonal coordinates representing rectangles. Specifically, when one corner of the predicted bounding box is fixed, the diagonal position can be any point on a circle with a fixed radius, resulting in identical ℓ_2 loss values.

Algorithm 2 l_2, *IoU* and *FIoU* as Bounding Box Losses

input: Predicted B^{pred} and ground truth B^{gt} bounding box coordinates:
$B^{pred} = (x_1^p, y_1^p, x_2^p, y_2^p)$, $B^{gt} = (x_1^g, y_1^g, x_2^g, y_2^g)$
output: $\mathcal{L}_{l_2}, \mathcal{L}_{IoU}, \mathcal{L}_{FIoU}$.
1. For the predicted box B^{pred}, ensuring $x_2^p > x_1^p$ and $y_2^p > y_1^p$:
$\hat{x}_1^p = \min(x_1^p, x_2^p)$, $\hat{x}_2^p = \max(x_1^p, x_2^p)$,
$\hat{y}_1^p = \min(y_1^p, y_2^p)$, $\hat{y}_2^p = \max(y_1^p, y_2^p)$.
2. Calculating area of B^{pred} and B^{gt}:
$A^p = (\hat{x}_2^p - \hat{x}_1^p) \times (\hat{y}_2^p - \hat{y}_1^p)$
$A^g = (x_2^g - x_1^g) \times (y_2^g - y_1^g)$.
3. Calculating intersection $\mathcal{I}$ between B^{pred} and B^{gt}:
$x_1^{\mathcal{I}} = \max(\hat{x}_1^p, x_1^g)$, $x_2^{\mathcal{I}} = \min(\hat{x}_2^p, x_2^g)$,
$y_1^{\mathcal{I}} = \max(\hat{y}_1^p, y_1^g)$, $y_2^{\mathcal{I}} = \min(\hat{y}_2^p, y_2^g)$,
$\mathcal{I}_w = x_2^{\mathcal{I}} - x_1^{\mathcal{I}}$, $\mathcal{I}_h = y_2^{\mathcal{I}} - y_1^{\mathcal{I}}$,
$\mathcal{I} = \begin{cases} \mathcal{I}_w \times \mathcal{I}_h & \text{if } x_2^{\mathcal{I}} > x_1^{\mathcal{I}}, y_2^{\mathcal{I}} > y_1^{\mathcal{I}} \\ 0 & \text{otherwise.} \end{cases}$
4. Finding the smallest enclosing box: $B^C(x_1^c, y_1^c, x_2^c, y_2^c)$, and its weight,
w, height, h, and the square of the diagonal, ρ^2, according to the steps
2-4 in Algorithm 1.
5. $l_2 = (x_1^g - \hat{x}_1^p)^2 + (y_1^g - \hat{y}_1^p)^2 + (x_2^g - \hat{x}_2^p)^2 + (y_2^g - \hat{y}_2^p)^2$.
6. $IoU = \frac{\mathcal{I}}{\mathcal{U}}$, where $\mathcal{U} = A^p + A^g - \mathcal{I}$.
7. $FIoU = IoU - \frac{l_2}{\rho^2}$.
8. $\mathcal{L}_{l_2} = l_2$, $\mathcal{L}_{IoU} = 1 - IoU$, $\mathcal{L}_{FIoU} = 1 - FIoU$.

However, this failed to adequately represent the prediction scenario. Additionally, $\ell_2$loss exhibits a strong correlation with scale variations, as illustrated in Fig. 4. We can see that given two pixels, one is located within the larger bounding box and the other within the smaller bounding box. The former will have a greater impact on the penalty than the latter, as the ℓ_2 loss is not standardized. This imbalance causes cellular neural networks to pay more attention to larger objects and ignore smaller ones.

In the case of IoU loss [24], although it exhibits scale invariance, inaccuracies arise when two predicted boxes exhibit an inclusion relationship, as shown in Fig. 3. In these situations, the IoU loss values may be identical; however, the representation of the prediction results is not accurate. Combining the strengths and weaknesses of both ℓ_2 and IoU losses, our proposed FIoU loss introduces normalization in the penalty term. This approach allows for the normalization of targets of different scales, effectively increasing the loss value when predicting smaller targets and enhancing the accuracy of small target detection.

2) THE SOLUTIONS OF GRADIENT VANISHING PROBLEM

To avoid losing generality during the derivation process, that is, to reduce the occurrence of negative signs, for each pixel (i, j) in an image, the bounding box of the ground truth can be defined as a 4-dimensional vector:

$$\widetilde{x}_{i,j} = (\widetilde{x}_{t_{i,j}}, \widetilde{x}_{b_{i,j}}, \widetilde{x}_{l_{i,j}}, \widetilde{x}_{r_{i,j}}) \tag{23}$$

where $\widetilde{x}_t, \widetilde{x}_b, \widetilde{x}_l, \widetilde{x}_r$ represent the distances between the current pixel location (i, j) and the top, bottom, left and right bounds of ground truth, respectively. Accordingly, the predicted bounding box is defined as $\boldsymbol{x} = (x_t, x_b, x_l, x_r)$, $I_h = min(x_t, \widetilde{x}_t) + min(x_b, \widetilde{x}_b)$, $I_w = min(x_l, \widetilde{x}_l) + min(x_r, \widetilde{x}_r)$, $I_H = max(x_t, \widetilde{x}_t) + max(x_b, \widetilde{x}_b)$, $I_W = max(x_l, \widetilde{x}_l) + max(x_r, \widetilde{x}_r)$, as shown in Fig. 1. According to Algorithm 2, we can deduce the backward algorithm of the IoU loss. First, we need to compute the partial derivative of A^p $w.r.t.$ x, marked as $\nabla_x A^p$ (for simplicity, we denote x for any of x_t, x_b, x_l, x_r if missing):

$$\begin{aligned} \frac{\partial \nabla_x A^p}{\partial x_t (\textbf{or } \partial x_b)} &= x_l + x_r, \\ \frac{\partial \nabla_x A^p}{\partial x_l (\textbf{or } \partial x_r)} &= x_t + x_b. \end{aligned} \tag{24}$$

then we need to compute the partial derivative of $\mathcal{I}$ $w.r.t.$ x, marked as $\nabla_x \mathcal{I}$

$$\begin{aligned} \frac{\partial \mathcal{I}}{\partial x_t (or\ \partial x_b)} &= \begin{cases} \mathcal{I}_w, & \text{if } x_t < \widetilde{x}_t \text{ (or } x_b < \widetilde{x}_b) \\ 0, & \text{otherwise,} \end{cases} \\ \frac{\partial \mathcal{I}}{\partial x_l (or\ \partial x_r)} &= \begin{cases} \mathcal{I}_h, & \text{if } x_l < \widetilde{x}_l \text{ (or } x_r < \widetilde{x}_r) \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{25}$$

Finally we can compute the gradient of localization $\mathcal{L}_{IoU}$ $w.r.t.$ x:

$$\begin{aligned} \frac{\partial \mathcal{L}_{IoU}}{\partial x} &= \frac{\partial (1 - \frac{\mathcal{I}}{\mathcal{U}})}{\partial x} \\ &= \frac{1}{\mathcal{U}^2} [\mathcal{U} \nabla_x A^p \mathcal{U} - (\mathcal{U} - \mathcal{I})(\nabla_x A^p - \nabla_x \mathcal{I})] \\ &= \frac{1}{\mathcal{U}^2} [(\mathcal{U} - \mathcal{I}) \nabla_x \mathcal{I} + \mathcal{I} \nabla_x A^p]. \end{aligned} \tag{26}$$

We found that when the union is zero, $\mathcal{I}$ and $\nabla_x \mathcal{I}$ are also zero, that is, the gradient of the IoU loss is zero, which leads to a gradient vanishing problem.

For the penalty term proposed in GIoU loss:

$$\mathcal{R}_{GIoU} = \frac{| C - B^{gt} \cup B^{pred} |}{| C |} \tag{27}$$

When the predicted bounding box and the ground truth box exhibit an inclusive or being-inclusive relationship, meaning C equals $B^{gt} \cup B^{pred}$, the Generalized Intersection over Union (GIoU) degenerates into IoU. In such cases, the backpropagation of the loss still results in gradient vanishing.

For the penalty term proposed in DIoU:

$$\mathcal{R}_{DIoU} = \frac{\rho^2(B^{gt}, B^{pred})}{{w_C}^2 + {h_C}^2} \tag{28}$$

we then compute the partial derivative of the $\mathcal{R}_{DIoU}$ with respect to the edge of the smallest enclosing bounding box:

$$\frac{\partial \mathcal{R}_{DIoU}}{\partial w_C} = -2w_C \frac{\rho^2(B^{gt}, B^{pred})}{{w_C}^2 + {h_C}^2} < 0$$
$$\frac{\partial \mathcal{R}_{DIoU}}{\partial h_C} = -2h_C \frac{\rho^2(B^{gt}, B^{pred})}{{w_C}^2 + {h_C}^2} < 0 \tag{29}$$

We found that $\mathcal{R}_{DIoU}$ provides a negative gradient for the size of the smallest enclosing box, which increases w_C and h_C and hinders the overlap between the predicted bounding box and the ground truth bounding box.

For the penalty term proposed in CIoU:

$$\mathcal{R}_{CIoU} = \frac{\rho^2(B^{gt}, B^{pred}))}{{\rho_C}^2} + \alpha V \tag{30}$$

where the values of α and V are given by (10) and (11), repectively. This penalty term considers the influence of the aspect ratio based on the $\mathcal{R}_{DIoU}$, and then takes the partial derivative of V:

$$\frac{\partial V}{\partial w_{pred}} = \frac{8}{\pi^2}\left(\arctan\frac{w_{gt}}{h_{gt}} - \arctan\frac{w_{pred}}{h_{pred}}\right)\frac{h}{h_{pred}^2 + {w_{pred}}^2}$$
$$\frac{\partial V}{\partial h_{pred}} = -\frac{8}{\pi^2}\left(\arctan\frac{w_{gt}}{h_{gt}} - \arctan\frac{w_{pred}}{h_{pred}}\right)\frac{w}{h_{pred}^2 + w_{pred}^2} \tag{31}$$

The drawback of CIoU is that $\frac{\partial V}{\partial w_{pred}} = -\frac{w_{pred}}{h_{pred}}\frac{\partial V}{\partial h_{pred}}$, meaning that V cannot provide gradients of the same sign for the width w_{pred} and height h_{pred} of the predicted bounding box. In the previous analysis of $\mathcal{R}_{DIoU}$, it was observed that $\mathcal{R}_{DIoU}$ could produce a negative gradient (29). When this negative gradient precisely counterbalances the gradient generated by the IoU loss on the predicted bounding box, the predicted bounding box is not optimized.

For the penalty term proposed in FIoU:

$$\mathcal{R}_{FIoU} = \frac{\sum_{i\in\{t,b,l,r\}}(x_i - \widetilde{x}_i)^2}{\mathcal{I}_W^2 + \mathcal{I}_H^2} = \frac{(x_t - \widetilde{x}_t)^2 + (x_b - \widetilde{x}_b)^2 + (x_l - \widetilde{x}_l)^2 + (x_r - \widetilde{x}_r)^2}{\mathcal{I}_W^2 + \mathcal{I}_H^2} \tag{32}$$

Taking x_t as an example, we compute the partial derivative of $\mathcal{R}_{FIoU}$ with respect to x_t.

$$\frac{\mathcal{R}_{FIoU}}{\partial x_t} = \begin{cases} 2(x_t - \tilde{x}_t)\cdot(\mathcal{I}_W^2 + \mathcal{I}_H^2) + [(x_t - \tilde{x}_t)^2 + E]\cdot 2\mathcal{I}_W, \\ \qquad\qquad\qquad\qquad \text{if } x_t > \widetilde{x}_t \\ 0, \qquad\qquad\qquad\qquad \text{if } x_t = \widetilde{x}_t \\ 2(x_t - \tilde{x}_t)\cdot(\mathcal{I}_W^2 + \mathcal{I}_H^2), \quad \text{if } x_t < \widetilde{x}_t \end{cases} \tag{33}$$

where E denotes $(x_b - \tilde{x}_b)^2 + (x_l - \tilde{x}_l)^2 + (x_r - \tilde{x}_r)^2$. We found that when $x_t > \widetilde{x}_t$, $\partial\mathcal{R}_{FIoU} > 0$, which provides a positive gradient for the x_t, that is, x_t becomes smaller and approaches $\widetilde{x}_t$, besides, $x_t < \widetilde{x}_t$, $\partial\mathcal{R}_{FIoU} < 0$, which provides a negative gradient for the x_t, that is, x_t will become larger and approach $\widetilde{x}_t$. Only when $x_t, \widetilde{x}_t$ are equal does the gradient equal to zero, which means that the value of the prediction box reaches its optimal state and no longer requires optimization. This can be extended to x_b, x_l, x_r.

In summary, this penalty term can provide a gradient regardless of x takes. If and only if the predicted value is equal to the ground truth value, the gradient is zero, which solves the problem of vanishing the gradient generated in the original IoU loss function.

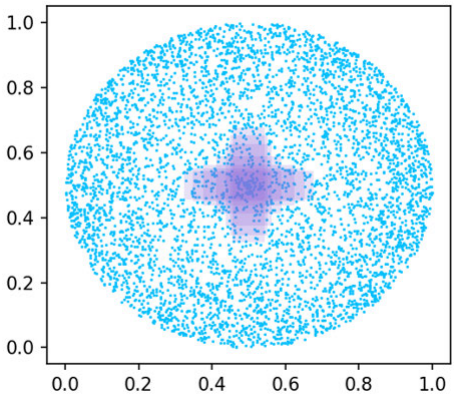


FIGURE 8. Anchor points (blue) and target boxes (purple) in simulation experiments.

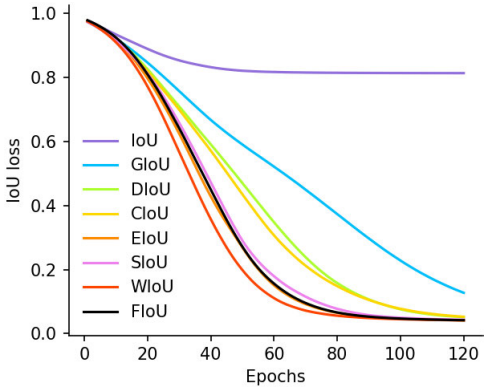


FIGURE 9. IoU loss curves of bounding box regression losses in simulation experiments.

3) SIMULATION EXPERIMENT

To compare the loss functions of several known bounding box regressions, we used the simulation experiment proposed by Zheng et al. [26] for evaluation. The initial conditions of the experiment were set as follows. First, seven different aspect ratios(*i.e.*,1:4, 1:3, 1:2, 1:1, 2:1, 3:1, and 4:1) were used to generate a ground truth bounding box centered at coordinate (0.5,0.5), with an area of 1/32. Then 5000 anchor points were generated at a radius of 0.5, and a center of (0.5, 0.5), each of which generated 49 anchor boxes, including seven different scales ((*i.e.*, 1/32, 1/24, 3/64, 1/16, 1/12, 3/32, 1/8) and seven different aspect ratios (*i.e.*,1:4, 1:3, 1:2, 1:1, 2:1, 3:1 and 4:1). Each anchor box must be matched with the ground truth bounding box, resulting in a total of $7\times7\text{x}7\times5000$=1,715,000 cases. We optimized the loss value by using a gradient descent algorithm with a learning rate

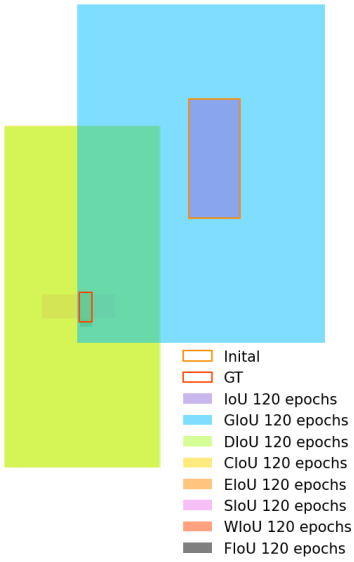


FIGURE 10. Regression results with different bounding box regression losses guiding.

of 0.01. The corresponding relationship between the anchor point and ground truth bounding box is shown in Fig. 8, and the comparison results are shown in Fig. 9.

From the above results, we can obtain the following observation: Because the $\mathcal{L}_{IoU}$ cannot converge under the condition that two bounding boxes do not intersect, other loss functions will cause the anchor box to converge towards the target box. In these IoU-based methods, the convergence speed of the $\mathcal{L}_{FIoU}$ we proposed is very fast, second only to that of $\mathcal{L}_{WIoU}$. Considering Fig. 10 as an example, after 120 epochs, the anchor box almost converges to the target box. This proves that our proposed loss function converges quickly and is also very accurate under simulation conditions.

IV. EXPERIMENTAL STUDIES

We evaluate our new bounding box regression loss $\mathcal{L}_{FIoU}$ by incorporating it into the most popular 2D object detectors such as Faster R-CNN [8], YOLOX [6], YOLOv7 [7] and YOLOv8. To this end, we replace their default regression losses with $\mathcal{L}_{FIoU}$, that is, we replace Smoothℓ_1 loss in Faster R-CNN, $\mathcal{L}_{GIoU}$ in YOLOX, $\mathcal{L}_{GIoU}$, $\mathcal{L}_{DIoU}$, $\mathcal{L}_{CIoU}$ in YOLOv7, and $\mathcal{L}_{GIoU}$, $\mathcal{L}_{DIoU}$ in YOLOv8. The experimental environment can be summarized as follows: the memory was 32GB, the operating system was Ubuntu20.04, the CPU was Intel i9-13900k, and the graphics card was an NVIDIA GeForce RTX 4090 with 24GB memory. All experiments were conducted using PyTorch for a fair comparison.

We trained all detection baselines and reported all the results on standard object detection benchmarks, that is, the PASCAL VOC 2007 & 2012 and MS-COCO 2017 datasets. The Pascal VOC benchmark is one of the most widely used datasets for classification, object detection, and semantic segmentation and consists of 20 classes. The train/val data contained 11,530 images containing 27,450 ROI annotated objects and 6,929 segmentations that were annotated with bounding boxes. The MS-COCO benchmark [13] is a large-scale dataset widely used in computer vision research, particularly for tasks such as object detection, segmentation, and image captioning. Created by Microsoft Research, the MS-COCO dataset was designed to address the limitations of existing datasets by providing a diverse range of object categories, capturing objects in complex scenes, and including high-quality annotations.

We report our experimental results by utilizing the performance measurements provided by the MS-COCO 2018 Challenge, which includes the calculation of mean Average Precision (mAP) over different class labels for a specific value of the IoU threshold 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 values of IoU thresholds, *i.e.*, $0.5 \leq IoU \leq 0.95$.

A. EXPERIMENTAL RESULTS OF FASTER R-CNN

1) TRANING PROTOCOL

We used the latest PyTorch implementations of Faster R-CNN (https://github.com/bubbliiiing/faster-rcnn-pytorch). For baseline results(trained using Smooth ℓ_1 loss), we used ResNet-50, the backbone network architecture for Faster R-CNN in all experiments, and followed their training protocol using the reported default parameters. We froze the backbone network and trained it for 50 epochs. After unfreezing, we continued to train for 150 epochs and selected the optimal weight to compare the results on the test set. Considering that Faster R-CNN is a two-stage object detection algorithm, it first performs RPN in the algorithm, which distinguishes the foreground and background. Subsequently, we fine-tuned and classified the detected results into the CNN network. In this experiment, we replaced the bounding box regression loss function of the RPN. To train this part using GIoU and FIoU losses, we replaced their Smooth ℓ_1 loss with $\mathcal{L}_{GIoU}$ and $\mathcal{L}_{FIoU}$ losses explained in (7) and Algorithm 2

2) RESULTS

Inspired by the concept of transfer learning, the features extracted from the main feature extraction part of the neural networks are universal. We used freeze training to accelerate the training efficiency and prevent weight damage. When the backbone network was frozen, the feature extraction network remained unchanged. The occupied graphics memory was small, but its impact on the network was minimal. However, after unfreezing, all the feature extraction networks changed. The occupied graphics memory was relatively large, and the training time was relatively long. From the results of the training, it can be seen that during the training process shown in Fig. 11, the map gradually stabilized with the increase in iteration times, while for the three loss functions we compared, the FIoU loss function we proposed, clearly performed better.

TABLE 1. Comparison between the performance of Faster R-CNN trained using its own loss ($\mathcal{L}_{Smoothl\ \ell_1}$ loss) as well as using $\mathcal{L}_{GIoU}$ and $\mathcal{L}_{FIoU}$ losses. The results are reported on the test set of **PASCAL VOC 2007 & 2012.**

Loss	*AP*	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
$\mathcal{L}_{Smooth\ \ell_1}$	0.443	0.714	0.477	0.116	0.282	0.517
$\mathcal{L}_{GIoU}$	0.449	0.713	0.490	0.118	0.273	0.524
$\mathcal{L}_{FIoU}$	**0.460**	**0.728**	**0.507**	**0.121**	**0.292**	**0.538**

TABLE 2. Comparison between the performance of YOLOX trained using its own loss ($\mathcal{L}_{IoU}$ loss) as well as using $\mathcal{L}_{GIoU}$ and $\mathcal{L}_{FIoU}$ losses. The results are reported on the test set of **PASCAL VOC 2007 & 2012.**

Loss	*AP*	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
$\mathcal{L}_{IoU}$	0.617	0.805	0.670	0.251	0.445	0.691
$\mathcal{L}_{GIoU}$	0.619	0.810	0.671	0.247	0.442	0.691
$\mathcal{L}_{FIoU}$	**0.627**	**0.812**	**0.679**	**0.253**	**0.451**	**0.706**

TABLE 3. Comparison between the proformance of YOLOv7 trained using its own loss ($\mathcal{L}_{CIoU}$ loss) as well as using $\mathcal{L}_{DIoU}$, $\mathcal{L}_{EIoU}$, $\mathcal{L}_{WIoU}$, $\mathcal{L}_{GIoU}$, $\mathcal{L}_{SIoU}$ and $\mathcal{L}_{FIoU}$ losses. The result are reported on the test set of **PASCAL VOC 2007 & 2012.**

Loss	*AP*	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
$\mathcal{L}_{GIoU}$	0.613	0.779	0.662	0.230	0.452	**0.719**
$\mathcal{L}_{DIoU}$	0.613	0.779	0.662	0.248	**0.457**	0.715
$\mathcal{L}_{CIoU}$	0.612	0.776	0.662	0.251	0.444	**0.719**
$\mathcal{L}_{EIoU}$	0.609	0.774	0.659	0.263	0.438	0.717
$\mathcal{L}_{SIoU}$	0.613	0.775	0.661	0.245	0.455	0.717
$\mathcal{L}_{WIoU}$	0.613	0.778	0.662	0.251	0.449	0.717
$\mathcal{L}_{FIoU}$	**0.615**	**0.780**	**0.665**	**0.266**	0.453	0.717

TABLE 4. Comparison between the performance of YOLOv7 trained using $\mathcal{L}_{FIoU}$ loss as well as using $\mathcal{L}_{DIoU}$, $\mathcal{L}_{EIoU}$ and $\mathcal{L}_{GIoU}$ losses. The results are reported on the val set of **COCO 2017.**

Loss	*AP*	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
$\mathcal{L}_{DIoU}$	0.504	0.689	0.548	0.345	0.552	**0.650**
$\mathcal{L}_{EIoU}$	0.506	0.690	0.550	0.344	0.554	0.648
$\mathcal{L}_{GIoU}$	0.506	0.689	**0.551**	0.348	**0.555**	0.646
$\mathcal{L}_{FIoU}$	**0.508**	**0.691**	**0.551**	**0.350**	0.552	**0.650**

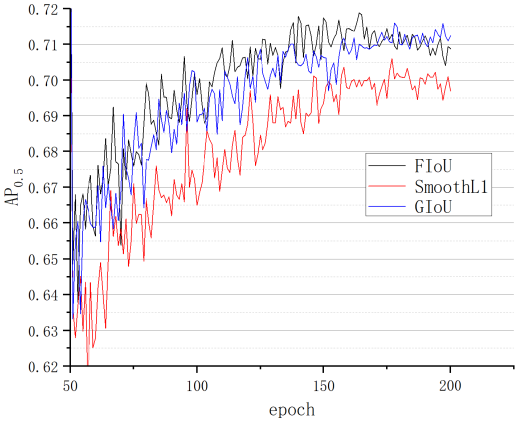


FIGURE 11. The map against training iterations when Faster R-CNN was trained using its standard loss ($\mathcal{L}_{Smoothl\ \ell_1}$ loss) as well as using $\mathcal{L}_{GIoU}$ and $\mathcal{L}_{FIoU}$ losses.

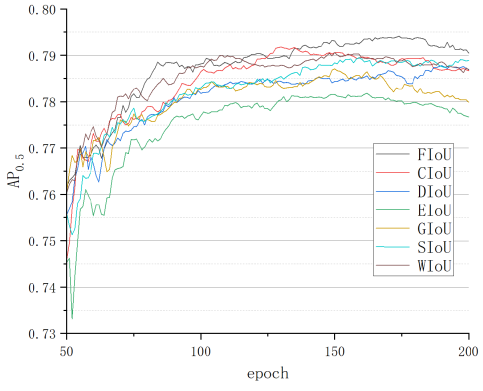


FIGURE 12. The $AP_{0.5}$ against training iterations when YOLOv7 was trained on PASCAL VOC 2007 & 2012 using $\mathcal{L}_{CIoU}$, $\mathcal{L}_{DIoU}$, $\mathcal{L}_{EIoU}$, $\mathcal{L}_{WIoU}$, $\mathcal{L}_{GIoU}$, $\mathcal{L}_{SIoU}$ and $\mathcal{L}_{FIoU}$ losses.

Their performance using the best network model for each loss was evaluated using the test set of PASCAL VOC 2007 & 2012, and the results are presented in Tab. 1. In the Fast R-CNN algorithm, the results trained using the FIoU loss function improved the map metrics by 1.1% and 1.7%, respectively, compared to the GIoU loss and the Smooth ℓ_1 loss used in the algorithm itself. In addition to the mAP indicator, the FIoU loss significantly improved the detection results in predicting multi-scale targets.

TABLE 5. Comparison between the performance of YOLOv8 trained using $\mathcal{L}_{FIoU}$ loss as well as using $\mathcal{L}_{DIoU}$, $\mathcal{L}_{EIoU}$ and $\mathcal{L}_{GIoU}$ losses. The results are reported on the val set of COCO 2017.

Loss	*AP*	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
$\mathcal{L}_{DIoU}$	0.363	0.515	**0.395**	0.177	0.397	**0.524**
$\mathcal{L}_{EIoU}$	0.360	0.513	0.392	0.177	0.389	0.508
$\mathcal{L}_{GIoU}$	0.362	0.514	0.393	0.186	**0.397**	0.512
$\mathcal{L}_{FIoU}$	**0.365**	**0.516**	0.393	**0.190**	0.396	0.522

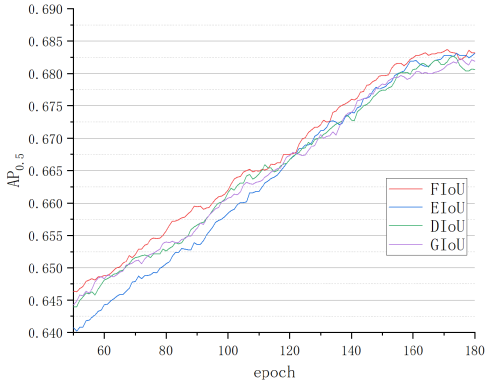


FIGURE 13. The $AP_{0.5}$ against training iterations when YOLOv7 was trained on COCO2017 using $\mathcal{L}_{FIoU}$, $\mathcal{L}_{DIoU}$, $\mathcal{L}_{EIoU}$ and $\mathcal{L}_{GIoU}$ losses.

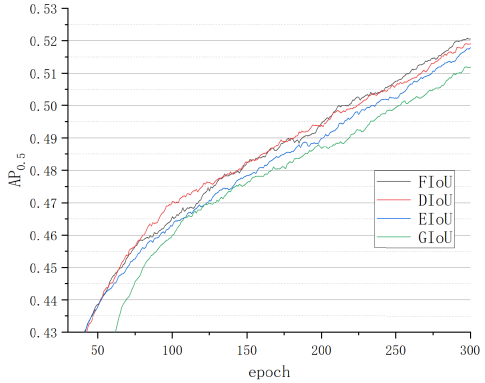


FIGURE 14. The $AP_{0.5}$ against training iterations when YOLOv8 was trained on COCO2017 using $\mathcal{L}_{FIoU}$, $\mathcal{L}_{DIoU}$, $\mathcal{L}_{EIoU}$ and $\mathcal{L}_{GIoU}$ losses.

B. EXPERIMENTAL RESULTS OF YOLOX

1) TRAINING PROTOCOL

We used the latest PyTorch implementations of YOLOX (https://github.com/bubbliiiing/yolox-pytorch). For the baseline results(trained using IoU loss), we used DarkNet-53, the backbone network architecture for YOLOX in all experiments, and followed their training protocol using the reported default parameters. We also froze the backbone network and trained this model for 50 epochs. After unfreezing, we continued to train for 150 epochs and selected the optimal weight to compare the results on the test set. To train YOLOX using GIoU and FIoU losses, we replaced IoU loss with $\mathcal{L}_{GIoU}$ and $\mathcal{L}_{FIoU}$.

2) RESULTS

Following the original training protocol, the network we trained using each loss in the training and validation sets of the dataset. Their performance using the best network model for each loss was evaluated using the PASCAL VOC 2007 & 2012 test, and the results are reported in Tab. 2, and it shows that in the YOLOX algorithm, the results trained using the FIoU loss function improved the mAP metrics by 1.0% and 0.8%, respectively, compared to the IoU loss and GIoU loss. FIoU performed better.

C. EXPERIMENTAL RESULTS OF YOLOV7

1) TRANING PROTOCOL

We used the latest PyTorch implementations of YOLOv7 (https://github.com/WongKinYiu/yolov7). For the baseline results(trained using $\mathcal{L}_{CIoU}$ loss), we used CSPDarkNet, the backbone network architecture for YOLOv7 in all experiments, and followed their training protocol using the reported default parameters.

We trained YOLOv7 using CIoU, DIoU, EIoU, WIoU, GIoU, SIoU, and FIoU losses and used each loss on the training set of the PASCAL VOC 2007 & 2012 dataset for up to 300 epochs. Their performance using the best weights for each loss was evaluated using the PASCAL VOC 2007 & 2012 test set. The results are presented in Tab. 3.

We trained YOLOv7 using DIoU, EIoU, GIoU, and FIoU losses and used each loss on the training set of the MS-COCO dataset for up to 180 epochs. Their performance using the best weights for each loss was evaluated using the MS-COCO 2017 val set. The results are presented in Tab. 4.

2) RESULTS

According to Fig. 12, the FIoU loss function performs better than other existing IoU-based loss functions during the training process. Although there are some shocks in the early stages of training, as the number of iterations increases, the advantages of the FIoU loss function become increasingly evident. In Fig. 15, we list some images from the test set of the PASCAL VOC 2007 & 2012 dataset. The optimal weights obtained by comparing the training of various loss functions were applied in the YOLOv7 algorithm to detect envoy images. We found that the results obtained by using FIoU loss can not only accurately identify the category of the target and the number of targets during the target detection process, but also accurately draw the prediction box. According to Fig. 13, the FIoU loss function also performs better than other existing IoU-based loss functions during the training process. Because we use pre-trained weights, during the iteration

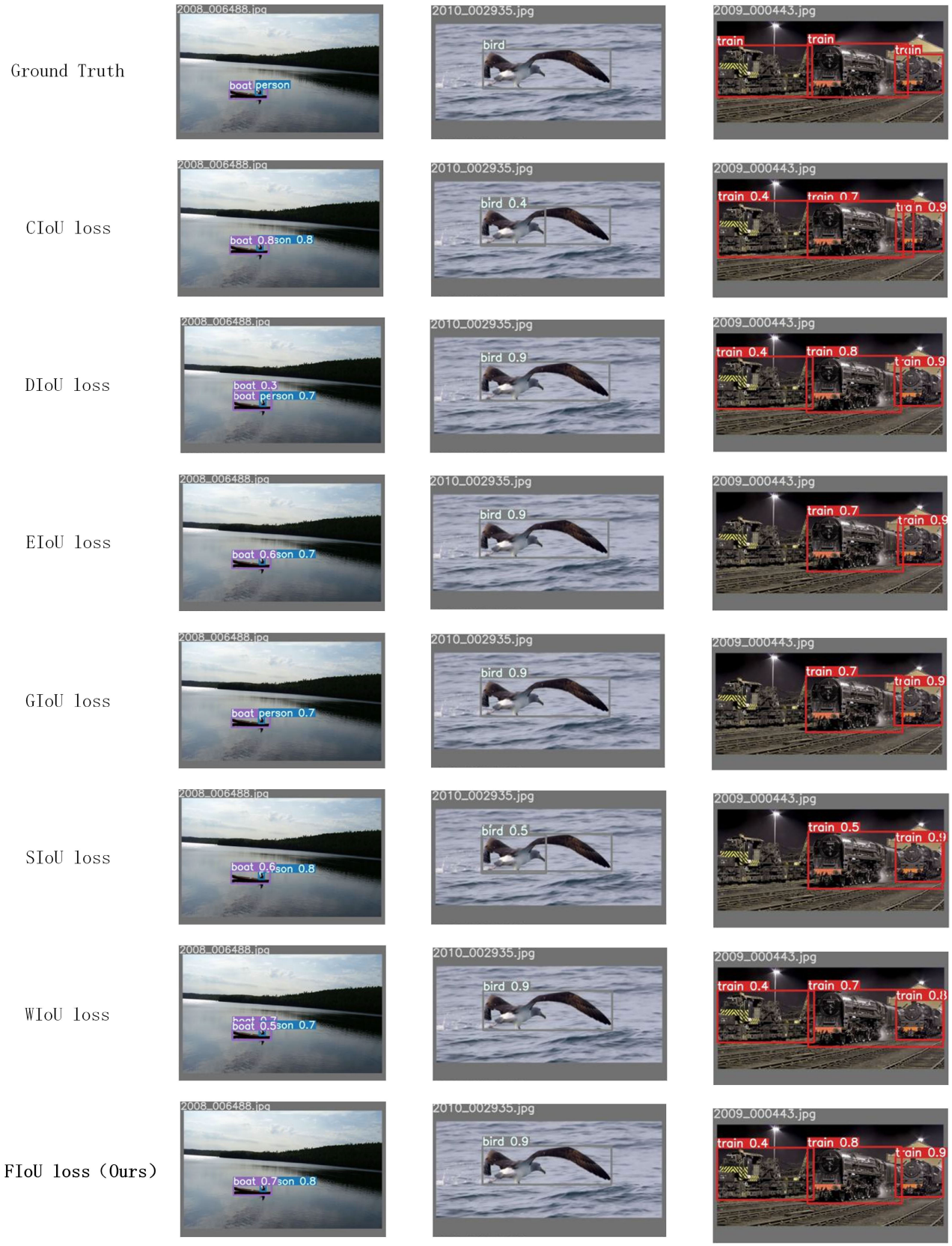


FIGURE 15. The comparison between the detection results obtained on the PASCAL VOC 2007 & 2012 test set using YOLOv7 algorithm with loss functions $\mathcal{L}_{CIoU}$, $\mathcal{L}_{DIoU}$, $\mathcal{L}_{EIoU}$, $\mathcal{L}_{GIoU}$, $\mathcal{L}_{SIoU}$, $\mathcal{L}_{WIoU}$, $\mathcal{L}_{FIoU}$, and the original images.

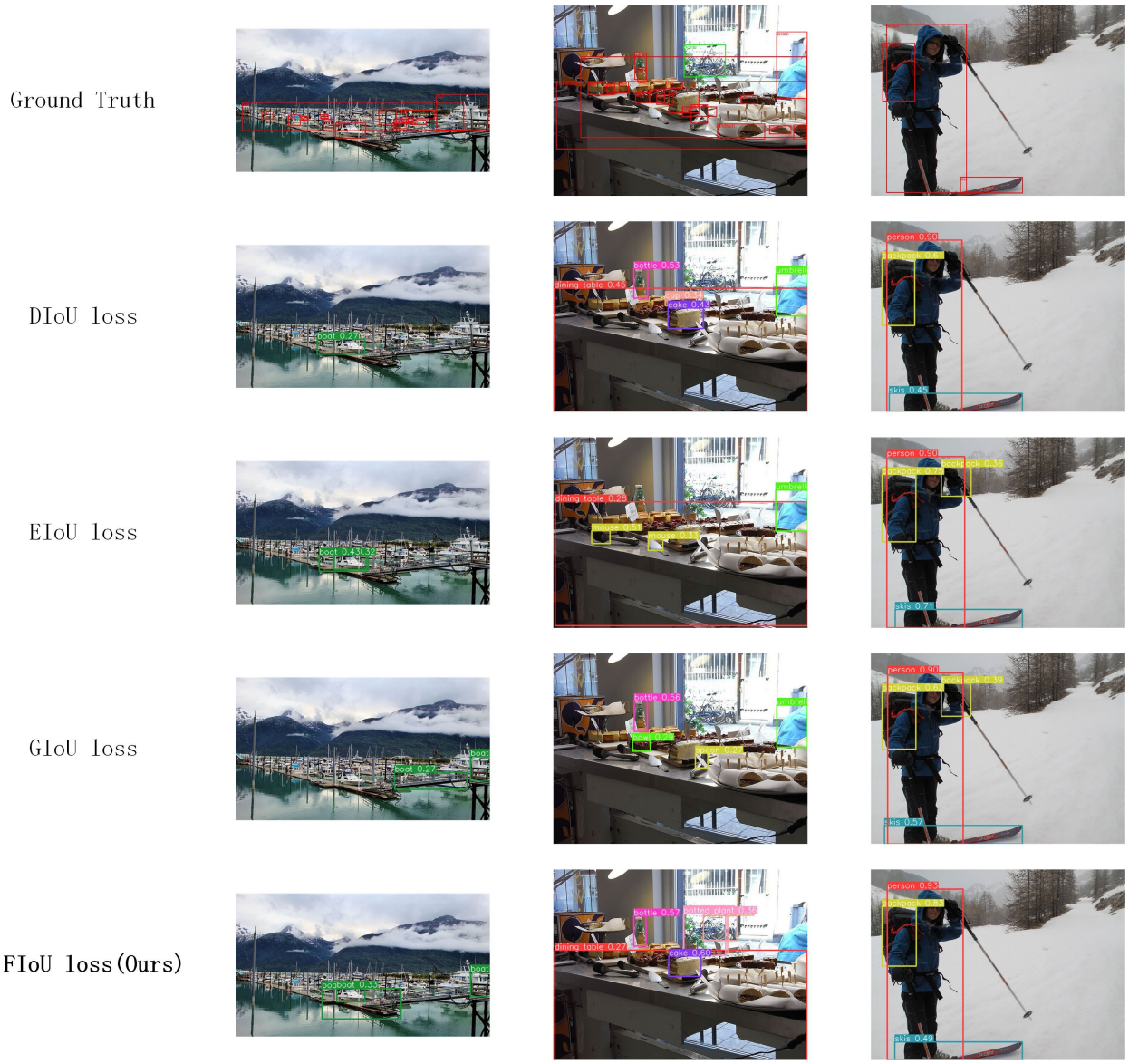


FIGURE 16. The comparison between the detection results obtained on the COCO 2017 val set using YOLOv8 algorithm with loss functions $\mathcal{L}_{DIoU}$, $\mathcal{L}_{EIoU}$, $\mathcal{L}_{FIoU}$, $\mathcal{L}_{GIoU}$, and the original images.

process, although there are small differences in the results obtained by each loss function training, the results of using the FIoU loss function are almost always better than those of other loss functions.

D. EXPERIMENTAL RESULTS OF YOLOV8

1) TRANING PROTOCOL

We used the latest PyTorch implementations of YOLOv8 (https://github.com/ultralytics/ultralytics). We trained YOLOv8 from scratch without using any other pre-trained weights and followed their training protocol using the reported default parameters.

We trained YOLOv8 using DIoU, EIoU, GIoU, and FIoU losses and used each loss on the training set of the MS-COCO dataset for up to 180 epochs. Their performance using the best weights for each loss was evaluated using the MS-COCO val set. The results are presented in Tab. 5.

2) RESULTS

According to Fig. 14, the FIoU loss function performs better than other existing IoU-based loss functions during the training process. In Fig. 16, we list some images from the val set of the MS-COCO 2017 dataset. This set of images includes images with very dense small targets of the same type, images with targets of different categories at different scales, and images with simple backgrounds. By comparing the original images and the prediction results, we found that our method can detect more targets more accurately, and its detection performance on small targets is also better. In the first set of images, the targets are mainly composed of boats of the same type. The proposed method can be used to detect

more boats. In the second set of images, there are many targets of different types and scales. Our method not only identifies nearby targets but also recognizes bicycles outside the window in the distance. In the third set of images, our proposed method can identify the targets more accurately.

E. DISCUSSION

To effectively evaluate the method proposed in this paper, we compared several indicators and conducted comparative experiments on popular object detection benchmarks, including the PASCAL VOC 2007 & 2012 and MS-COCO datasets. The experimental results showed that our proposed FIoU loss function not only addresses the issue of gradient vanishing in backpropagation by adding a normalized ℓ_2 norm as a penalty term but also exhibits a faster convergence speed compared to similar loss functions. In addition, our method has been verified to achieve more accurate detection accuracy on metrics such as mAP.

In the Faster R-CNN algorithm, we compared our proposed FIoU loss, Smooth ℓ_1 loss, and the widely used GIoU loss. The mAP indicators were increased by 1.7% and 1.1%, respectively. Particularly in the $AP_{0.75}$ indicator, improvements of 3.0% and 1.7%, respectively. For the YOLOX algorithm, we compared the proposed FIoU loss with IoU loss and GIoU loss. The results indicated that the mAP indicators were increased by 1.0% and 0.8%, respectively. For the YOLOv7 algorithm, we conducted ablation experiments on two datasets, comparing the detection results of various mainstream loss functions. Through comparative experiments, our proposed loss function performed better overall, particularly for small target detection.

Building upon this, in the newly proposed YOLOv8 algorithm, we compared FIoU loss with DIoU loss, GIoU loss, and EIoU loss, achieving improvements of 0.2%, 0.5%, and 0.3% in the mAP metrics, respectively. Despite these improvements, the accuracy of small object detection remains very low, and this issue needs to be addressed in future work.

V. CONCLUSION

In this paper, we first summarized the ℓ_n-norm-based and IoU-based bounding box regression loss functions and analyzed their advantages and disadvantages. On this basis, we proposed our own method, that is, a new metric named FIoU, to compare the similarity and overlap between the predicted bounding box and ground truth bounding box, which preserved all the attributes of IoU; thus, it could be applied as an optional loss function to existing target detection network frameworks to solve the problem of target localization.

In addition, we propose an FIoU loss function. It fused ℓ_2 loss and IoU loss functions, transforming x, y defined in the location space and w, h defined in the size space into the coordinates of the diagonal points of the bounding box and normalizing targets of different scales using the square the smallest enclosing bounding box diagonal, which solved the problem of significant differences in loss values for targets of different scales using the ℓ_2 loss function. In addition, we calculated the gradient of the FIoU as a loss function during the backpropagation process, which solved the problem of vanishing the IoU loss function gradient. Moreover, we compared the convergence speeds of the existing loss functions under simulation conditions. The experimental results indicate that the convergence speed of the FIoU as a loss function exceeds that of most IoU-based loss functions. Finally, we combined the mainstream algorithms Faster R-CNN, YOLOX, YOLOv7, and YOLOv8 on the general datasets PASCAL VOC 2007 & 2012 and MS-COCO 2017 for validation, and the results showed that our algorithm performed the best.

In the future, we hope to expand experiments on downstream tasks based on object detection to verify the generalization ability of our proposed loss function.

REFERENCES

[1] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Jun. 2010.
[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[3] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.
[4] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[5] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[6] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.
[7] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 7464–7475.
[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015.
[9] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.
[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.
[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[12] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.
[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 *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.
[14] L. Li, X. Yao, X. Wang, D. Hong, G. Cheng, and J. Han, "Robust few-shot aerial image object detection via unbiased proposals filtration," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5617011.
[15] X. Qian, B. Wu, G. Cheng, X. Yao, W. Wang, and J. Han, "Building a bridge of bounding box regression between oriented and horizontal object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5605209.
[16] X. Qian, Y. Huo, G. Cheng, C. Gao, X. Yao, and W. Wang, "Mining high-quality pseudoinstance soft labels for weakly supervised object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5607615.

[17] X. Qian, C. Li, W. Wang, X. Yao, and G. Cheng, "Semantic segmentation guided pseudo label mining and instance re-detection for weakly supervised object detection in remote sensing images," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 119, May 2023, Art. no. 103301.
[18] D. Forsyth, "Object detection with discriminatively trained part-based models," *Computer*, vol. 47, no. 2, pp. 6–7, Feb. 2014.
[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.
[20] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," 2017, *arXiv:1701.06659*.
[21] P. Zhou, B. Ni, C. Geng, J. Hu, and Y. Xu, "Scale-transferrable object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 528–537.
[22] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "RepPoints: Point set representation for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9657–9666.
[23] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9627–9636.
[24] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: An advanced object detection network," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 516–520.
[25] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.
[26] 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 *Proc. AAAI Conf. Artif. Intell.*, vol. 34, Feb. 2020, pp. 12993–13000.
[27] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing geometric factors in model learning and inference for object detection and instance segmentation," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8574–8586, Aug. 2022.
[28] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient IOU loss for accurate bounding box regression," *Neurocomputing*, vol. 506, pp. 146–157, Sep. 2022.
[29] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," 2022, *arXiv:2205.12740*.
[30] Z. Tong, Y. Chen, Z. Xu, and R. Yu, "Wise-IoU: Bounding box regression loss with dynamic focusing mechanism," 2023, *arXiv:2301.10051*.

YONG SUN received the B.E. degree from the Beijing Institute of Technology, Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include machine design, computer vision, and deep learning.

JIANZHONG WANG received the B.E., M.E., and Ph.D. degrees from the Nanjing University of Science and Technology, Nanjing, China. From 1990 to 2002, he was with the Wuhan University of Technology, Wuhan, China, where he is currently a Professor of mechanical and electrical engineering. Since 2002, he has been with the Beijing Institute of Technology, Beijing, China, where he is a Professor with the School of Mechatronics Engineering and the State Key Laboratory of Explosion Science and Technology. His current research interests include intelligent systems, unmanned ground vehicles, and multi-robot cooperative technology.

HONGFENG WANG was born in Shandong, China, in 1994. He received the B.E. degree in mechatronical engineering and the Ph.D. degree in armament science and technology from the Beijing Institute of Technology, China, in 2017 and 2023, respectively. His research interests include object detection and tracking and gaze estimation.

SHENG ZHANG received the B.E. degree in mechatronical engineering from the Beijing Institute of Technology, Beijing, China in 2018, where he is currently pursuing the Ph.D. degree. His current research interests include unmanned ground vehicle (UGV), deep reinforcement learning, and simulation technology.

YU YOU received the M.E. degree from the Beijing Institute of Technology, Beijing, China, in 2021, where he is currently pursuing the Ph.D. degree. His research interests include machine design, computer vision, and deep learning.

ZIBO YU was born in 1998. He received the bachelor's degree in mechanical and electronic engineering from the Beijing Institute of Technology, in 2020, where he is currently pursuing the Ph.D. degree in mechanical engineering. His research interests include target detection and tracking.

YIGUO PENG received the B.E. degree from the Beijing Institute of Technology, Beijing, China, in 2021, where he is currently pursuing the M.E. degree. His research interests include camouflage object detection, computer vision, and deep learning.