A Systematic IoU-Related Method: Beyond Simplified Regression for Better Localization

Hanyang Peng, Shiqi Yu Member, IEEE

Abstract—Four-variable-independent-regression localization losses, such as Smooth-$\ell_1$ Loss, are used by default in modern detectors. Nevertheless, this kind of loss is oversimplified so that it is inconsistent with the final evaluation metric, intersection over union (IoU). Directly employing the standard IoU is also not infeasible, since the constant-zero plateau in the case of non-overlapping boxes and the non-zero gradient at the minimum may make it not trainable. Accordingly, we propose a systematic method to address these problems. Firstly, we propose a new metric, the extended IoU (EIoU), which is well-defined when two boxes are not overlapping and reduced to the standard IoU when overlapping. Secondly, we present the convexification technique (CT) to construct a loss on the basis of EIoU, which can guarantee the gradient at the minimum to be zero. Thirdly, we propose a steady optimization technique (SOT) to make the fractional EIoU loss approaching the minimum more steadily and smoothly. Fourthly, to fully exploit the capability of the EIoU based loss, we introduce an interrelated IoU-predicting head to further boost localization accuracy. With the proposed contributions, the new method incorporated into Faster R-CNN with ResNet50+FPN as the backbone yields 4.2 mAP gain on VOC2007 and 2.3 mAP gain on COCO2017 over the baseline Smooth-$\ell_1$ Loss, at almost no training and inferencing computational cost. Specifically, the stricter the metric is, the more notable the gain is, improving 8.2 mAP on VOC2007 and 5.4 mAP on COCO2017 at metric AP$_{50}$.

Index Terms—Object Detection, Loss Function, IoU, Optimization

I. INTRODUCTION

Object detection is a heavily-investigated topic in the computer vision community, because it is fundamental and the prerequisite for many other vision tasks, such as instance segmentation [1], [2] and high-level object-based reasoning [3].

With the advent of deep CNNs [4], [5], [6] in recent years, the performance of object detection has progressed substantially. There are generally two possible approaches to improve detection accuracy besides increasing samples: constructing ingenious architectures and devising better losses. Constructing CNN architectures have made great strides in the past years [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. One tendency is to design more and more sophisticated architectures for better performance, but this way commonly will increase the computational cost. In contrast, devising better losses is more economical, since we can obtain the improvement at little cost of extra training and inferencing time. However, research on devising losses, especially localization losses, received much less attention in past years. Since R-CNN introduced a four-variable-independent-regression loss for localization in 2013 [13], the localization loss in modern deep detectors changed little. Although the four-variable-independent-regression loss is simple and straightforward, it is not consistent with the final detection performance metric, IoU. The gap between the four-variable-independent-regression loss and IoU will inevitably result in some misaligned cases – the loss is small, but IoU is also small, which means the predicted box and the ground-truth box overlap little, and vice versa. An example in Fig 1 visually illustrates this gap between Smooth-$\ell_1$ Loss (the most widely-used four-variable-independent-regression loss) and IoU. It is intuitive that equipping an IoU related loss can address this problem.

However, the standard IoU based losses did not popularize in the past years, since there are two intrinsic deficiencies in the standard IoU. (i) When the predicted box and the ground-truth box do not overlap, the standard IoU itself is ill-defined since the value is constant zero. Then, the gradient of any standard IoU based loss will also become zero, so the backpropagation cannot pull the predicted box close to the ground-truth in this non-overlapping case. (ii) The gradient of a simple standard IoU loss at the minimum where two boxes completely overlap is non-zero, which will bring about oscil-
lution and slow convergence when applying gradient descent algorithms. Very recently, [19] pioneeringly proposed $GIOU$ that adds a regularization term after a standard $IOU$ loss, and then the new loss has non-zero gradients when two boxes are not overlapping. However, the regularization term also makes $GIOU$ not equivalent to the standard $IOU$ when two boxes are overlapping. Hence the performance of $GIOU$ might be suboptimal as the standard $IOU$ is the final evaluation metric. Moreover, $GIOU$ Loss still does not overcome problems of oscillation and slow convergence. [20] presented $CIoU$ loss by incorporating the normalized distance between two boxes into the standard $IOU$. Actually, $CIoU$ can be considered as the combination of a four-variable-independent-regression loss and a standard $IOU$ loss. $CIoU$ converges much faster than $GIOU$, but it still can not avoid oscillation due to non-zero gradients at the minimum.

In this paper, we will propose a systematic method to tackle all the problems above and introduce some new techniques to further improve the detection accuracy.

- We propose a more generalized and well-defined $IOU$, namely $EIOU$. In the case of overlapping bounding boxes, $EIOU$ is identical to the standard $IOU$, while in the case of non-overlapping boxes, $EIOU$ is smaller as two boxes separate further, which will make $EIOU$ trainable.
- We present a convexification technique (CT) to construct a new loss. It will lead the gradient to become zero at the minimum. So it is possible to achieve the minimum through gradient descent algorithms. Moreover, just like Focal Loss, the convexification technique will adaptively assign higher weight on hard examples.
- We introduce a steady optimizing technique (SOT) to make the loss approach the minimum steadily and smoothly. The convergence of the steady optimization technique is theoretically ensured.
- Harnessing the computed ground-truth $IOU$ score in the new loss above, we add a single-layer head to be trained to predict this $IOU$ score. Then, we can utilize the predicted $IOU$ score to help non-maximum suppression (NMS) select more precise bounding boxes in the inferring stage.

II. RELATED WORK

Architectures of CNN based detectors. The architecture of modern CNN based detectors can be generally divided into two parts: the backbone network and the detection-specific network. The backbones are commonly borrowed from the networks designed for categorization, of which VGG [21], ResNet [6], ResNeXt [22] are often leveraged. Besides, some specially-designed backbones for detection were also proposed in past years, such as DarkNet [23] and DetNet [24], and Hourglass Net [25] are also frequently adopted.

There are two different logics to design a detection-specific network. The first one is the two-stage network, and it consists of two sub-networks, where the first one is to generate a sparse set of candidate proposers, and the other is to determine the accurate location and categories based on the proposals. R-CNN [18], fast R-CNN [7] and Faster R-CNN [8] shaped the basic network architecture of two-stage detectors, and then R-FCN [26] replaced the fully-connected sub-network with a convolution sub-network to improve efficiency. FPN [9] introduces a lateral network to produce object proposals at multiple scales with more contextual information. Cascade R-CNN devised a cascade structure and it improves performance substantially [10], [27] proposed IoU-Net and IoU guided NMS to acquire location confidence for accurate detection. Grid R-CNN [28] can capture the spatial information explicitly and enjoys the position-sensitive property of fully convolutional architecture. Very recently, TridentNet [29] constructed a parallel multi-branch architecture aiming to generate scale-specific feature maps with a uniform representational power.

Another one is the one-stage network, which directly predicts the locations and categories of the object instance. YOLO [30] and SSD [12] first popularized the one-stage methods by much reducing the computational cost but still maintaining competitive performance. Then, DSSD [31] and RON [32] introduced a network similar to the hourglass network to combine low-level and high-level information. RetinaNet [13] with Focal loss as the one-stage detectors first outperformed the two-stage detectors. RefineDet [33] designed the anchor refinement module and the object detection module to reduce negative boxes and improve detection. CornerNet [34] is an anchor-free framework and adopts two subnetworks to detect the top-left and bottom-right key points and then employs a grouping subnetwork to pair them. Later some other competitive anchor-free detectors, such as FSAF [35], FCOS [36] and CenterNet [37], [38], were further developed.

These ingenious architectures significantly promoted the evolution of object detection. It is worth noting that the improvement of detection performance is partly attributed to sophisticated backbones and detection-specific networks that will commonly bring extra computational cost.

Losses of CNN based detectors. Compared with the design of architectures, the exploration of losses is more economical, because a well-devised loss can obtain performance gain with little additional train time cost and no extra test time cost. However, research on losses for detection has been underestimated for a long time.

Modern CNN based detectors were popularized by R-CNN in 2013 [18], and it introduced the softmax loss and a four-variable-independent-regression loss for classification and localization. Since then, this type of classification loss and localization loss became mainstream and were applied to most detectors. As for the classification loss, YOLO [30] used to employ the $\ell_2$ loss for categorization, but the later improved YOLO9000 [23] gets back on track to reuse the softmax loss. Afterwards, Focal Loss [13] was specially developed to address extreme foreground-background ratio problem in one-stage detectors. It can adaptively down-weight overwhelming well-classified background examples to enjoy better detection performance. Recently, [15] exploits new losses to address the object rotation problem and the within-class diversity problem.

In terms of localization loss, Fast R-CNN substitutes the four-variable-independent-regression $\ell_2$ loss using in R-CNN with Smooth-$\ell_1$ loss [7]. The localization loss of the latter CNN based detectors mostly follow Smooth-$\ell_1$ loss with no
or little change [8], [30], [13], [9], [27]. However, as illustrated in Section 1 and Fig 1, there is misalignment between Smooth-$\ell_1$ loss and the evaluation metric of IoU. So [39] tried to introduce a standard IoU based loss to address this problem. Nevertheless, the standard IoU also has its own defect. As long as two boxes are mutually detached no matter how far the distance is, the standard IoU will become constant zero, so that the gradient of a standard IoU based loss will also become zero and the loss is not trainable in this case. GIoU [19] introduced a well-designed term added after a standard IoU based loss, and then the new loss becomes non-zero when two boxes are separated. This pioneering work made great progress to make IoU based loss feasible. But just the adding term makes this new loss no longer equal the standard IoU. Hence it may lead to an unexpected result that GIoU Loss in some cases of overlapping boxes is larger than that in some cases of non-overlapping boxes.

In this work, we will propose a systematical method to tackle the problems above of existing localization losses.

III. THE PROPOSED APPROACH

In this section, we will present this systematic approach. We first introduce the standard IoU, and interpret its plight for handling the situation that two boxes are non-overlapping. Next, we will show how we devise a new extended IoU that can overcome the difficulty above. Then, we will use a convexification technique/local technique to construct an extended IoU based loss. Afterward, we will provide a steady optimization technique to make the training process steadily and smoothly. Finally, we will present an interrelated IoU-predicting head to select more precise predicted bounding boxes.

A. Standard IoU

Constructing an IoU based loss is an intuitive way to tackle the unappealing problems that the four-variable-independent-regression losses bring. However, the standard IoU (SIoU) has some deficiencies that hinder the prevalence of IoU based losses, and we will elaborate it in the following.

Given the targeted bounding boxes with a tuple $(x_1^t, y_1^t, x_2^t, y_2^t)$ and the predicted box with a tuple $(x_1^p, y_1^p, x_2^p, y_2^p)$, where $x_1, y_1$ and $x_2, y_2$ are the coordinate value of the top-left and bottom-right corners of the bounding boxes, respectively. When two boxes are overlapping, the definition of the standard SIoU is

$$x_1 = \max(x_1^t, x_1^p),$$

$$y_1 = \max(y_1^t, y_1^p),$$

$$x_2 = \min(x_2^t, x_2^p),$$

$$y_2 = \min(y_2^t, y_2^p),$$

$$I_{std} = (x_2 - x_1)(y_2 - y_1),$$

$$S_t = (x_2^t - x_1^t)(y_2^t - y_1^t),$$

$$S_p = (x_2^p - x_1^p)(y_2^p - y_1^p),$$

$$U_{std} = S_t + S_p - I_{std},$$

$$SIoU = \frac{I_{std}}{U_{std}}.$$  \hspace{1cm} (10)

However, when two boxes are not overlapping, the value of the intersection $I_{std}$ and SIoU is constant 0, which will bring two drawbacks.

- SIoU cannot distinguish whether the two boxes are just in the vicinity or they are separated remotely.
- The gradient of the SIoU for backpropagation will also become zero.

Hence SIoU is not trainable in this case.

B. Extended IoU

In this subsection, we introduce our extended IoU (EIoU) that is accurately equivalent to the standard IoU in the case of overlapping boxes and has non-zero gradients in the case of non-overlapping boxes.

Conserving the definition of Eq.(10), the extended intersection $(I_c)$ is

$$x_0 = \min(x_1^t, x_1^p)$$

$$y_0 = \min(y_1^t, y_1^p)$$

$$x_{\min} = \min(x_1, x_2)$$

$$y_{\min} = \min(y_1, y_2)$$

$$x_{\max} = \max(x_1, x_2)$$

$$y_{\max} = \max(y_1, y_2)$$

$$I_c = S_1 + S_2 + S_3 + S_4$$

$$= (x_2 - x_0)(y_2 - y_0) + (x_{\min} - x_0)(y_{\min} - y_0)$$

$$- (x_1 - x_0)(y_{\max} - y_0) - (x_{\max} - x_0)(y_1 - y_0),$$

where we define $I_c = S_1 + S_2 - S_3 - S_4$, in which $S_1$ is area of the rectangle with top-left corner $(x_0, y_0)$ and bottom-right $(x_2, y_2)$; $S_2$ is area of the rectangle with top-left corner $(x_0, y_0)$ and bottom-right $(x_{\min}, y_{\min})$; $S_3$ is area of the rectangle with top-left corner $(x_0, y_0)$ and bottom-right $(x_{\max}, y_1)$; $S_4$ is area of the rectangle with top-left corner $(x_0, y_0)$ and bottom-right $(x_{\max}, y_1)$.

We enumerate all the four situations whether two boxes are overlapping or not overlapping for the proposed $I_c$ in the following.

(i) As shown in Fig.2(a), when two boxes are overlapping with $x_1 < x_2$ and $y_1 < y_2$, we have $x_{\min} = x_1$, $x_{\max} = x_2$, $y_{\min} = y_1$ and $y_{\max} = y_2$, and then

$$I_c = (x_{\max} - x_0)(y_{\max} - y_0) + (x_{\min} - x_0)(y_{\min} - y_0)$$

$$- (x_1 - x_0)(y_{\max} - y_0) - (x_{\max} - x_0)(y_1 - y_0)$$

$$= (x_{\max} - x_{\min})(y_{\max} - y_{\min})$$

$$= (x_2 - x_1)(y_2 - y_1) > 0.$$  \hspace{1cm} (17)

1Actually, SIoU is not trainable only when all the pair boxes are non-overlapping. In practice, it is common there are overlapping pair boxes and non-overlapping pair boxes in a batch. Hence the total gradient of a batch might not be zero. However, the exist of non-overlapping boxes in a batch will still make the performance for SIoU degrade which can be seen in Table 1.
Fig. 2: Illustration the difference between $EIou$ and $SIou$. It is known $IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2}$ and $S_t$ and $S_p$ are fixed, so the differences between $I_e$ and $I_{std}$ are the key. From Eq.\ref{6-9}, we know $I_e = S_1 + S_2 - S_3 - S_4$, where $S_1$ is area of the rectangle with top-left corner $(x_0, y_0)$ and bottom-right $(x_2, y_2)$; $S_2$ is area of the rectangle with top-left corner $(x_0, y_0)$ and bottom-right $(x_1, y_{max})$; $S_3$ is area of the rectangle with top-left corner $(x_0, y_0)$ and bottom-right $(x_{max}, y_1)$; $S_4$ is area of the rectangle with top-left corner $(x_1, y_{max})$ and bottom-right $(x_{2}, y_{2})$. Thus, when two boxes are overlapping as shown in (a) with $x_1 < x_2$ and $y_1 < y_2$, $I_e$ is always positive and equivalent to the standard $IoU$.

(ii) As shown in Fig 2(b), when two boxes are non-overlapping with $x_1 > x_2$ and $y_1 < y_2$, we have $x_{min} = x_2$, $x_{max} = x_1$, $y_{min} = y_1$ and $y_{max} = y_2$, and then

$$I_e = (x_{min} - x_0)(y_{min} - y_0) + (x_{min} - x_0)(y_{min} - y_0) - (x_{max} - x_0)(y_{max} - y_0) - (x_{max} - x_0)(y_{max} - y_0) = 0$$

(iii) As shown in Fig 2(c), when two boxes are non-overlapping with $x_1 < x_2$ and $y_1 > y_2$, we have $x_{min} = x_1$, $x_{max} = x_2$, $y_{min} = y_2$ and $y_{max} = y_1$, and then

$$I_e = (x_{max} - x_0)(y_{min} - y_0) + (x_{min} - x_0)(y_{min} - y_0) - (x_{max} - x_0)(y_{max} - y_0) - (x_{max} - x_0)(y_{max} - y_0) = 0$$

(iv) As shown in Fig 2(d), when two boxes are non-overlapping with $x_1 > x_2$ and $y_1 > y_2$, we have $x_{min} = x_2$, $x_{max} = x_1$, $y_{min} = y_2$ and $y_{max} = y_1$, and then

$$I_e = (x_{min} - x_0)(y_{min} - y_0) + (x_{min} - x_0)(y_{min} - y_0) - (x_{max} - x_0)(y_{max} - y_0) - (x_{max} - x_0)(y_{max} - y_0) = 0$$

Therefore, $I_e$ is positive and reduced to $I_{std}$ in the case of overlapping and $I_e$ is negative and decreases with the distance of two boxes in the case of non-overlapping.

Differences From GIoU. Both $GIoU$ \cite{c19} and the proposed $EIou$ aim to address the problem of zero gradients when two boxes do not overlap, but there are still some significant distinctions between them. As shown in Algorithm 1, $GIoU$ adds an extra term after $SIou$, which can be considered as a regularization metric. The new term indeed makes $GIoU$ have non-zero gradients when two boxes are detached, but it also leads $GIoU$ to be not equivalent to $SIou$ any more when two boxes are attached. This change will cause new problems. First, it brings some counter-intuitive and unreasonable cases, and one example is visually illustrated in Fig 4. Second, the performance of $GIoU$ might be suboptimal as $SIou$ is the final evaluation metric. As for $EIou$ is not a regularization method and an incremental modification of $GIoU$. We fundamentally address the root of the problem by redefining $IoU$, so that it is trainable in the case of non-overlapping and equivalent(reduced) to $SIou$ in the case of overlapping. Accordingly, $EIou$ will never encounter similar plights shown in Fig 4.

Algorithm 1: $GIoU$ in \cite{c19}

Input: Two arbitrary bounding boxes: $A$ and $B$
Output: $GIoU$
1. Find the smallest bounding box $C$ that encloses $A$ and $B$
2. Compute the standard $IoU$: $SIou = \frac{A \cap B}{A \cup B}$
3. Compute $GIoU$: $GIou = SIou - \frac{C \cap (A \cup B)}{C}$

C. Covexification Technique (CT)

Loosely speaking, any a decreasing function w.r.t. $IoU$ can be treated as a localization loss, such as $\frac{1}{1+IoU}$, $-IoU$ and $-\ln(IoU)$, but there are two problems in these simple $IoU$
based losses. First, they are not ensured to be always non-negative. Second, the gradients at the minimum of them are not zero. It is well known that (stochastic) gradient methods ideally achieve a minimal point of which the gradient must be zero. Thus, theoretically, it cannot achieve the minimum if we use these losses in training. To make matters worse, non-zero gradients at the minimum are more likely to make the training process oscillating/non-convergent and even collapsed in practice. To tackle these problems, we present the convexification technique (CT) to modify the loss and make it practical during training. It needs two steps:

(i) Add the opposite number of the minimum of the original loss.
(ii) Square the sum above.

Adopting CT, any a decreasing functions w.r.t. $Iou$ will become a well-defined loss, so that it is always non-negative and the gradient at the minimum is zero. Note that CT is general, which can be employed to modify any loss not limited the localization loss and make it possess appealing characteristics. In this paper, we present a new loss based on the simplest decreasing function $-EIoU$ w.r.t. $EIoU$. The minimal value of $-EIoU$ is $-1$, so the loss is obtained through CT as follow\(\text{3}\), i.e.,

$$L_{\text{Smooth-EIoU}} = \left(1 - \frac{I_e}{U_e}\right)^2$$

CT can smooth loss functions, so the new loss is referred to as Smooth-$EIoU$ Loss. The new loss is also like Focal Loss. CT leads Smooth-$EIoU$ Loss to possess focal capability that down-weights the gradient of well-localized predicted boxes, i.e.,

$$\frac{\partial L_{\text{Smooth-EIoU}}}{\partial z} = - \left(1 - \frac{I_e}{U_e}\right) \frac{\partial \left(\frac{I_e}{U_e}\right)}{\partial z}$$

where $z$ is any one of $\{x_1^p, y_1^p, x_2^p, y_2^p\}$. It is known $EIoU$ between a well-localized box and the ground-truth box is close to 1, and then $\left(1 - \frac{I_e}{U_e}\right)$ will be close to 0. Thus, $\frac{\partial L_{\text{Smooth-EIoU}}}{\partial z}$ will also become very small, which means Smooth-$EIoU$ Loss will down-weight easy pair boxes and pay more attention to hard pair boxes in training.

\(\text{3}\)For a more general method, the power order is not limited to 2 but can be any number more than 1, such as $L_{\text{Smooth-EIoU}} = \left(1 - \frac{I_e}{U_e}\right)^{1.5}$.

The following example illustrates the importance of CT. Given the targeted bounding box with a tuple $(0,0,1,1)$ and the predicted bounding box with a tuple $(0,0,x,y)$, the value space of $-EIoU$ and Smooth-$EIoU$ Loss constructed from $-EIoU$ with CT are shown in (a)-(b). Smooth-$EIoU$ Loss becomes smooth after employing CT, and then the gradients of the loss are gradually close to zero when approaching minimum, so CT makes Smooth-$EIoU$ Loss achieves the minimum when applying a gradient descent algorithm, which can be observed in (c). (c) shows the convergence behavior of $-EIoU$ and Smooth-$EIoU$ Loss when the predicted bounding box starts with the initial value $(0,0,0.5,0.5)$. Not surprisingly, $-EIoU$ oscillates severely and there is no tendency to be converged. In contrast, Smooth-$EIoU$ Loss quickly and smoothly converges to the optimum. Notably, the steady optimization technique (SOT) that we will elaborate in the next subsection is adopted for $-EIoU$ and Smooth-$EIoU$ Loss in this experiment.

D. Steady Optimization Technique (SOT)\n
For simplicity, we only deduce the partial derivative of Smooth-$EIoU$ Loss in Eq. (23) w.r.t. $x_1^p$ here, and others are similar and presented in the appendix. We first compute the gradient of $I_e$ w.r.t. $x_1^p$, i.e.,

$$\frac{\partial I_e}{\partial x_1^p} = \begin{cases} y_{\min} - y_{\max}, & \text{if } x_1^p \geq x_1^* \text{ and } x_1 \leq x_2, \\ 2y_0 - y_{\min} - y_1, & \text{if } x_1^p \geq x_1^* \text{ and } x_1 > x_2, \\ 0, & \text{if } x_1^p < x_1^*. \end{cases}$$

(25)

And then we compute the gradient of $U_e$ w.r.t. $x_1^p$

$$\frac{\partial U_e}{\partial x_1^p} = (y_0^p - y_2^p) - \frac{\partial I_e}{\partial x_1^p}$$

(26)

Finally we obtain the gradient of Smooth-$EIoU$ Loss w.r.t. $x_1^p$

$$\frac{\partial L_{\text{Smooth-EIoU}}}{\partial x_1^p} = 2 \left(1 - \frac{I_e}{U_e}\right) \frac{I_e \frac{\partial I_e}{\partial x_1^p} - \frac{\partial L_{\text{Smooth-EIoU}}}{\partial x_1^p}}{U_e^2}$$

(27)

The partial derivative of Smooth-$EIoU$ Loss w.r.t. $y_1^p$, $x_2^p$, and $y_2^p$ are similar to Eq. (27) (details please see Appendix.A). From Eq. (16), Eq. (21), and Eq. (26), we know $I_e \propto s^2$, $U_e \propto s$ and $\frac{\partial I_e}{\partial x_1^p} \propto s$ where $s$ is the size (height or width) of the predicted box $(x_1^p, y_1^p, x_2^p, y_2^p)$. Hence, we analyze Eq. (27) and find $\frac{\partial L_{\text{Smooth-EIoU}}}{\partial x_1^p} \propto 1$, which means the gradient of Smooth-$EIoU$ Loss w.r.t. $x_1^p$ is inverse proportional to the size of the predicted box. This inverse proportion will make the loss difficult to converge in train, since when the size of the predict box is large, it means the absolute difference between the targetted box and the predicted box is also large, and then it needs to update with a relatively large step, but if applying gradient in Eq. (27) to update the variables, the update is small instead. When the size of the boxes is small, it will encounter a similar dilemma. A good iteratively update for variables should be proportional to the size, just like the gradients of $\ell_2$ Loss.
To achieve this goal, we change the update rule for variables of $EIoU$. We take $x_1^p$ for example, i.e.,

$$x_{1k} = x_{1k-1} - 2\alpha \frac{\partial L_{G1IoU}}{\partial x_1^p} U_e$$

$$= x_{1k-1} - 2\alpha \left(1 - \frac{L}{U_e}\right) \frac{I_e \frac{\partial U}{\partial x_1^p} - \frac{\partial L}{\partial x_1^p} U_e}{U_e}$$

(28)

where $k$ is the number of iterations and $\alpha$ is the learning rate. Compared with Eq. (27), Eq. (28) multiplies $U_e$ to make sure the new gradient update is proportional to the scale of the boxes.

We call this method as the steady optimization technique (SOT). This technique seems to be heuristic, but we will theoretically prove it reasonable in the following.

**Theorem 1:** If the gradient of $f(x)$, denoted as $\nabla f(x)$, is Lipschitz continuous, i.e.,

$$\|\nabla f(x_1) - \nabla f(x_2)\| \leq L \|x_1 - x_2\|_2,$$  

(29)

the function $g(x)$ is positive and bounded, i.e., $0 < g(x) \leq M$, and the learning rate satisfies $\alpha < \frac{1}{LM}$, the update update rule,

$$x_{k+1} = x_k - \alpha g(x_k) \nabla f(x_k),$$

(30)

will make $f(x)$ steadily decrease.

We provide the proof in the appendix. $U_e$ in our Smooth-$EIoU$ Loss is always greater than zero. Therefore if we set the learning rate properly, SOT can ensure Smooth-$EIoU$ Loss steadily decreases. From Eq. (28) we know the Smooth-$EIoU$ Loss is nonnegative and bounded, hence SOT will further guarantee it to be convergent according to the bounded monotonic principle.

According to Theorem 1, SOT is very general and can be applied to optimize many types of losses for steady convergence, including but not limited to fractional losses of which its gradients is not linearly proportional to the size.

We design two examples in 5 to further demonstrate the superiority of SOT. As shown in 5 SOT will make the convergence of the loss steady, and it is robust to the size of the initial predicted box and the targeted box. When the size of the target box and predicted box proportionally varies, the targeted box are fixed with $(0, 0, 1, 1)$ (small), $(0, 0, 2, 2)$ (medium) and $(0, 0, 4, 4)$ (large), and the initial value of the predicted is set as $(0, 0, 0.5, 0.5)$ (small), $(0, 0, 1, 1)$ (medium) and $(0, 0, 2, 2)$ (large). The converges tendency of Smooth-$EIoU$ Loss with SOT is completely the same regardless of the size, while Smooth-$EIoU$ Loss without SOT is very sensitive to the varied size. The larger the size is, the slower the convergence rate is. Just like what we analyzed. (b) comparison when only the size of the predicted box varies: the targeted box are fixed with $(0, 0, 1, 1)$, and the initial value of the predicted is set as $(0, 0, 0.5, 0.5)$ (small), $(0, 0, 1, 1)$ (medium) and $(0, 0, 2, 2)$ (large). Smooth-$EIoU$ Loss optimized with/without SOT: (a) comparisons when the size of the targeted box and predicted box proportionally varies: the targeted box are fixed with $(0, 0, 1, 1)$ (small), $(0, 0, 2, 2)$ (medium) and $(0, 0, 4, 4)$ (large), and the initial value of the predicted is set as $(0, 0, 0.5, 0.5)$ (small), $(0, 0, 1, 1)$ (medium) and $(0, 0, 2, 2)$ (large). The converges tendency of Smooth-$EIoU$ Loss with SOT is completely the same regardless of the size, while Smooth-$EIoU$ Loss without SOT is very sensitive to the varied size. The larger the size is, the slower the convergence rate is, just like what we analyzed. (b) comparison when only the size of the predicted box varies: the targeted box are fixed with $(0, 0, 1, 1)$, and the initial value of the predicted is set as $(0, 0, 0.5, 0.5)$ (small), $(0, 0, 1, 1)$ (medium) and $(0, 0, 2, 2)$ (large). Smooth-$EIoU$ Loss with SOT still can quickly converge, but Smooth-$EIoU$ Loss without SOT is more sensitive to the size under this circumstance. When the initial value of the predicted box is set as $(0, 0, 4, 4)$, it is even trapped and cannot move to the target box.
(0, 0, 2, 2) (medium) and (0, 0, 4, 4) (large). Smooth-$E\text{IoU}$ Loss with SOT still can quickly converge, but Smooth-$E\text{IoU}$ Loss without SOT is more sensitive to the size under this circumstance. When the initial value of the predicted box set as (0, 0, 4, 4), it is even trapped and cannot move to the target box.

E. IoU Head

In [27] it has demonstrated that there is a misalignment between classification confidence and localization accuracy, and utilizing precisely predicted IoU scores of bounding boxes to guide NMS will largely alleviate this problem. Taking advantage of the existing ground-truth IoU calculated in Smooth-$E\text{IoU}$ Loss, we add IoU Head and train it to predict accurate IoU scores. It is known IoU distributes over [0, 1], so we first utilize the sigmoid function to compress the predicted IoU score to [0, 1], and then a Kullback-Leibler (KL) divergence loss is employed in train, i.e.,

$$q_p(x) = \text{Sigmoid}(x),$$

$$\mathcal{L}_{KL} = q_g \log \frac{q_g}{q_p(x)} + (1 - q_g) \log \frac{1 - q_g}{1 - q_p(x)},$$

where $x$ is the output of IoU Head, $q_p(x)$ is the predicted IoU score and $q_g$ is the ground-truth IoU score that is generated in Smooth-$E\text{IoU}$ Loss.

Note that IoU Head is a single layer, and it shares most parameters with the classification head and the bounding-box head. Hence, it will increase little computational cost in train and test.

**Differences From IoU-Net.** [27] pionereingly proposed IoU-Net learning to predict IoU to promote the localization accuracy. However, there are still some significant differences between our IoU Head and the IoU-Net. Firstly, we used a KL loss that is widely proven to be effective for deep neural networks rather than a squared loss. Secondly, it needs to manually construct synthetical bounding-box sets to train IoU-Net individually besides training the main branches of classification and localization, while our IoU Head can seamlessly embed to the existing network and be trained end-to-end. Thirdly, IoU Head is much lighter than IoU-Net. IoU-Net is an individual subnet and works parallelly with the classification subnet and the localization subnet, while IoU Head is a single-layer branch and shares most layers with the main branches. Architectures of IoU-Net and IoU Head are visually illustrated in Fig 6. Fourthly, the ground-truth IoU used in IoU Head is generated by the localization head, so IoU Head and Localization Head are closely interrelated with better cooperativity, and the effect of "$1 + 1 > 2$" between them is shown in Table 1. But IoU-Net has little relation with the localization head.

IV. IMPLEMENTATION

In modern deep CNN based detectors, the neural network does not directly estimate the coordinates of the bounding box, and instead it predicts the normalized difference value between the corresponding coordinates of the anchor or proposal box (henceforth, we only use anchor box for simplicity) and the targeted box, and the normalization value is the width and height of anchor box. We adopt a similar strategy to generate the predicted box, but we uniformly employ the square root of the area of the anchor box to normalize all the corresponding coordinates rather than independently normalize them with the corresponding coordinate of the anchor, since the former will keep the width-height ratio of the predicted box and targeted box. Implementation details please see Algorithm 1.

**Algorithm 1. Training EIoU Loss**

**Input:** the anchor box $(x^a_1, y^a_1, x^a_2, y^a_2)$, the target box $(x^t_1, y^t_1, x^t_2, y^t_2)$ and the CNN predicted normalized difference value $(x^p_1, y^p_1, x^p_2, y^p_2)$

**Output:** the EIoU Loss $\mathcal{L}_{\text{GI-EIoU}}$

Compute $S = \sqrt{(x^t_2 - x^t_1)(x^t_2 - x^t_1)}$

Compute $(x^{p,n}_1, y^{p,n}, x^{p,n}_2, y^{p,n}) = \left(\frac{x^p_1 + x^p_2}{2}, \frac{y^p_1 + y^p_2}{2}, \frac{x^t_1 + x^t_2}{2}, \frac{y^t_1 + y^t_2}{2}\right)$

Compute $(x^{p,n}_1, y^{p,n}, x^{p,n}_2, y^{p,n}) = \left(\frac{x^t_1 + x^t_2}{2}, \frac{y^t_1 + y^t_2}{2}, \frac{x^p_1 + x^p_2}{2}, \frac{y^p_1 + y^p_2}{2}\right)$

**while not convergence do**

Compute $(x^{k,n}_1, y^{k,n}, x^{k,n}_2, y^{k,n}) = (x^{k-1}_1, y^{k-1}_1, x^{k-1}_2, y^{k-1}_2)$

Using Eq. (11) and (10) to compute $\mathcal{L}_{\text{GI-EIoU}}$

Using Eq. (25) and (27) to compute

$$\left(\frac{\partial \mathcal{L}_{\text{GI-EIoU}}}{\partial y^{k,n}_1}, \frac{\partial \mathcal{L}_{\text{GI-EIoU}}}{\partial y^{k,n}_2}, \frac{\partial \mathcal{L}_{\text{GI-EIoU}}}{\partial x^{k,n}_1}, \frac{\partial \mathcal{L}_{\text{GI-EIoU}}}{\partial x^{k,n}_2}\right) = \left(\frac{\partial g_{EIoU_1}}{\partial y^{k,n}_1}, \frac{\partial g_{EIoU_2}}{\partial y^{k,n}_2}, \frac{\partial g_{EIoU_1}}{\partial x^{k,n}_1}, \frac{\partial g_{EIoU_2}}{\partial x^{k,n}_2}\right)$

Using Eq. (28) to update $(x^{k+1}_1, y^{k+1}_1, x^{k+1}_2, y^{k+1}_2)$

**end while**

V. EXPERIMENT

A. Experimental Setting

All the experiments are conducted on the benchmark datasets – PASCAL VOC and MS COCO. Detectors are implemented in Facebook AI Research’s Detectron system [40]. Following the default settings in Detectron, we trained all the detectors on 8 NVIDIA P100 GPUs. Each mini-batch totally contains 16 images which are uniformly distributed to 8 GPUs. Input images are resized to 500 and 800 pixels along the short side on PASCAL VOC and MS COCO, respectively.
TABLE I: Ablation study by using Faster R-CNN with ResNet50 + FPN as the backbone. Models are trained on the union set of VOC_2007_trainval and VOC_2012_trainval. The results are reported on the set of VOC_2007_test.

| Smooth-ℓ₁ | S1oU | G1oU | E1oU | CT | SOT | IoU Head | AP          |
|-----------|------|------|------|----|-----|----------|-------------|
| (1)       | ✓    |      | -    | -  | -   | -        | 45.5 (Smooth-ℓ₁ Loss, Baseline) |
| (2)       |      | ✓    | -    | -  | -   | -        | 46.6 (Standard IoU Loss)       |
| (3)       |      |      | ✓    | -  | -   | -        | 46.9 (G1oU Loss)               |
| (4)       | ✓    |      | -    | -  | -   | -        | 46.2 (Smooth-ℓ₁ Loss with IoU Head) |
| (5)       |      |      | -    | ✓  | -   | -        | 47.5 (E1oU Loss)               |
| (6)       |      |      | -    | -  | ✓   | -        | 47.9 (E1oU Loss with CT)       |
| (7)       |      |      | -    | ✓  | ✓   | -        | 48.2 (E1oU Loss with CT, SOT)  |
| (8)       |      |      | -    | -  | ✓   | ✓        | 49.7 (E1oU Loss with CT, SOT and IoU Head) |

No other data augmentation except from the standard horizontal image flipping is employed. Standard SGD with weight decay of 0.0001 and momentum of 0.9 is adopted. We train the detectors with 20k iterations for PASCAL VOC and 90k (180k) iterations for MS COCO, and the learning rate is set to 0.02 at the begin and then decreased by a factor of 0.1 after 12k and 18k for PASCAL VOC and 60k (120k) and 80k (160k) iterations for MS COCO, respectively. We comply with the MS COCO evaluation protocol to report the experimental results.

B. Ablation Study

We implement ablation experiments on PASCAL VOC to clarify the contributions of the proposed E1oU, CT, SOT and IoU Head, and the results are reported in Table I.

As shown in Table I with the standard S1oU based loss replacing the baseline Smooth-ℓ₁ Loss, the performance is improved to some extents (+1.1% mAP, comparing (1) and (2)). Substituting S1oU with G1oU further boost the performance with scores +0.3% mAP (comparing (2) and (3)), which is consistent with the results in Table 5 in [19]. Comparing with G1oU, individually equipping E1oU can bring more substantial improvement (±0.9 % mAP, (2) and (5)), which indicates E1oU may be more pitorially powerful than G1oU. With the help of CT, the performance is continually promoted (±0.4 % mAP, (5) and (6)). Exploiting SOT in train further receives a gain of ±0.3 % mAP scores (comparing (6) and (7)). Adding IoU Head to the net significantly improves the performance (+1.5% mAP, comparing (7) and (8)). Interestingly, E1oU Loss with IoU Head can generate better cooperativity than Smooth-ℓ₁ Loss with IoU Head (+1.5% mAP vs ±0.7% mAP, comparing (7) and (8) and comparing (1) and (4)). The reason for it is that IoU Head has close relation to a IoU related loss, so they can receive the effect of ”1 + 1 > 2”. Totally, the proposed systematical method including E1oU, CT, SOT and IoU Head yields significant gains, which is 4.2% higher than the baseline Smooth-ℓ₁ Loss that is overwhelmingly used in popular detectors (comparing (1) and (8)).

C. Comparison to the Related Localization Losses

The proposed systematical method is mainly built on localization loss, so we will extensively compare the proposed method to widely-used Smooth-ℓ₁ Loss and the related G1oU Loss [19] and CIoU [20] Loss in this subsection. For simplicity, our systematical method is referred to as E1oU Loss henceforth. All the losses are attached to RetinaNet (that is a typical one-stage model) and Faster-RCNN (that is a typical two-stage detection model) during training. Overall Mean average Precision(mAP) for all the three losses is reported in Table II. Besides, the results of Average Precision (AP) at IoU thresholds: [0.5, 0.75, 0.90] and for individual small-size, medium-size and large-size objects are also listed for detailed comparison.

As shown in Table II and III compared with Smooth-ℓ₁ Loss and G1oU Loss, E1oU Loss in one-stage and two-stage detectors can steadily yield gains on PASCAL VOC and MS COCO. Specifically, for the baseline Smooth-ℓ₁ Loss that is dominant in popular detectors, our approach combining Faster R-CNN substantially boosts 4.2% AP and 1.2% mAP on PASCAL VOC and COCO, respectively. When comparing with G1oU Loss, E1oU loss can still consistently surpass it by a more than 2.0% margin on PASCAL VOC and an 1.0% margin on COCO.

There is an interesting phenomenon that when the IoU threshold is set to 0.5 , the performance of our approach is close to Smooth-ℓ₁ Loss. However, when the threshold grows higher, E1oU Loss gradually outperforms Smooth-ℓ₁ Loss and G1oU Loss.. Especially at AP₉₀, comparing with Smooth-ℓ₁ Loss, E1oU Loss improves 8.2% on PASCAL VOC dataset and 5.3% on MS COCO dataset . The reason for it is E1oU Loss can help a detector to predict more accurate bounds than Smooth-ℓ₁ Loss. It is known there is a gap...
between Smooth-$\ell_1$ Loss and the final evaluation IoU, and the relative gap is enlarging as two boxes are gradually matched, while $EIoU$ is exactly equivalent to $IoU$ when two boxes are overlapping. Moreover, Smooth-$\ell_1$ Loss decreases quicker than $EIoU$ Loss as two boxes are gradually matched, so during training Smooth-$\ell_1$ Loss commonly gives less attention to better matched pair-boxes. Therefore, comparing to Smooth-$\ell_1$ Loss, $EIoU$ Loss will receive more gains when the final evaluation metric (IoU) is stricter.

Another phenomenon observed from Table II and III is that $EIoU$ Loss seems to be superior to detect small-size objects, comparing to $GIoU$ Loss. Although the overall performance of $GIoU$ Loss is 1.4% higher than Smooth-$\ell_1$ Loss on PASCAL VOC dataset with Faster-RCNN, but Smooth-$\ell_1$ Loss and $GIoU$ Loss obtain similar scores (10.0% and 9.6%) for small-size objects, which means $GIoU$ Loss is still weak to detect small-size objects.

$EIoU$ Loss achieves 11.9 % under the same conditions. The superiority of $EIoU$ loss to detect small-size objects stems from the IoU predict head. In post-processing, conventionally we use classification loss to detect small-size objects stems from the IoU predict head. In post-processing, conventionally we use classification
confidence to guide non-maximum suppression (NMS) to filter redundant bounding boxes. Commonly, the correlation of classification confidence and localization confidence is weaker when detecting smaller objects. In our method, we use the predicted IoU confidence to correct the bias of classification confidence and localization confidence. Hence, our method has a better capacity for finding smaller objects.

Additionally, in terms of improvement, Faster-RCNN + FPN with $EIoU$ Loss performs better than RetinaNet with $EIoU$ Loss. It may be due to that there are denser anchor boxes in RetinaNet. Hence it is not so difficult to exactly regress the targeted boxes for Smooth-$\ell_1$ Loss.

As shown in Fig 7, the superior performance of $EIoU$ Loss for Average Recall (AR) are more obvious than that for AP across the different value of IoU threshold, which means $EIoU$ Loss is more powerful to find more objects, comparing with the popular localization losses.

D. Comparisons to State-of-the-Art Detectors

We evaluate $EIoU$ Loss attached to FPN on the MS COCO 2019 test-dev set with 180k iterations and compare the results to state-of-the-art one-stage and two-stage detectors. The experimental results are presented in Table IV. For fair comparison, we only list the results of competitors of a single model with no sophisticate data argumentation in the training and testing. Without bells and whistle, our method with ResNeXt-64x4d-101+FPN achieves 44.1% mAP, which surpasses the counterparts in the Table [V] by a large margin. Compared to the closest competitor Grid R-CNN [28], the superiority of the proposed approach is more substantial at the higher IoU threshold (0.75), improving more than 1.0% (47.6% vs 46.6%), which is consistent with that our method can predict more precise bounding boxes.

VI. CONCLUSION AND DISCUSSION

Smooth-$\ell_1$ Loss and its variants dominate the localization loss in modern CNN based detectors. Nevertheless, their oversimplified assumption that four coordinate variables of a bounding box are independent does not accord with the fact. Therefore the localization performance of these detectors might suffer degradation. In light of this, we propose a generalized $EIoU$ to address this problem. To make the $EIoU$ based loss not oscillated in the neighbourhood of the minimum and steadily optimized in train, we introduce CT and SOT. Moreover, we present IoU Head to further improve localization accuracy.

Very Recently, a wide variety of anchor-free detectors [34], [35], [36], [37], [38] were developed and receive more and more attention. We think the proposed $EIoU$ Loss may be more applicable to these detection models, because there may exist more non-overlapping box pairs due to no anchors.

We provide a new route to design IoU based losses, and all the decreasing functions of IoU can be modified and become an applicable localization loss through CT. We just tried the simplest $-IoU$, and many other functions not limited to $\frac{1}{1-IoU}$, $-\ln(IoU)$ might be more appropriate. Therefore there is great potential to further the performance by exploiting these techniques.

More importantly, CT and SOT are so general that they can beyond the field of detection. CT can help any loss to have zero-gradient at the minimum and make it possible to achieve the minimum through gradient descend algorithms. SOT can help many types of losses, including but not limited fractional losses (fractional losses are common in machine learning tasks, since we usually need to minimize an objective function and maximize another simultaneously ), to steadily and smoothly arrive at the minimum. Therefore, CT and SOT may find more applications in other fields.

REFERENCES

[1] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask r-cnn,” in IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
[2] P. O. Pinheiro, T. Y. Lin, R. Collobert, and P. Dollár, “Learning to refine object segments,” in European Computer Conference on Computer Vision, 2016, pp. 75–91.
[3] J. Wu, E. Lu, P. Kohli, B. Freeman, and J. Tenenbaum, “Learning to see physics via visual de-animation,” in Advances in Neural Information Processing Systems, 2017, pp. 153–164.
[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in International Conference on Neural Information Processing Systems, 2012, pp. 1097–1105.
[5] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” arXiv preprint:1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167
[6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[7] R. Girshick, “Fast r-cnn,” in IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[8] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017.
[9] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 936–944.
[10] Z. Cai and N. Vasconcelos, “Cascade r-cnn: Delving into high quality object detection,” in IEEE Conference on Computer Vision and Pattern Recognition, 2018.
[11] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” arXiv preprint:1804.02767, 2018.
[12] 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, 2016, pp. 21–37.
[13] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PP, no. 99, pp. 2999–3007, 2017.
[14] W. He, X. Y. Zhang, F. Yin, and C. L. Liu, “Deep direct regression for multi-oriented scene text detection,” in IEEE International Conference on Computer Vision, 2017, pp. 745–753.
[15] G. Cheng, J. Han, P. Zhou, and D. Xu, “Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection,” IEEE Transactions on Image Processing, vol. 28, no. 1, pp. 265–278, 2019.
[16] G. Cheng, J. Yang, D. Gao, L. Guo, and J. Han, “High-quality proposals for weakly supervised object detection,” IEEE Transactions on Image Processing, vol. 29, pp. 5794–5804, 2020.
[17] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, “Object detection in optical remote sensing images: A survey and a new benchmark,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 159, pp. 296 – 307, 2020.
[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in IEEE International Conference on Computer Vision, 2014, pp. 580–587.
In this section, we will deduce all the partial derivative of Smooth-\textit{EIoU} in Eq.(23) w.r.t. $x_p^1$, $y_p^1$, $x_p^2$ and $y_p^2$. We first compute the partial derivative, i.e.,

$$
\frac{\partial L_e}{\partial x_1^p} = \begin{cases} 
    y_{\text{min}} - y_{\text{max}}, & \text{if } x_1^p < x_1^t \\
    2y_0 - y_{\text{max}} - y_1, & \text{if } x_1^p \geq x_1^t \text{ and } x_1 > x_2, \\
    2x_0 - x_{\text{max}} - x_1, & \text{if } x_1^p \geq x_1^t \text{ and } y_1 \geq y_2, \\
    0, & \text{if } x_1^p < x_1^t.
\end{cases}
$$

(33)

Finally we obtain the the partial derivative of Smooth-\textit{EIoU} Loss w.r.t. $x_1^p$

$$
\frac{\partial L_{\text{Smooth-EIoU}}}{\partial x_1^p} = 2 \left( 1 - \frac{I_e}{U_e} \right) \frac{I_e \frac{\partial U_e}{\partial x_1^p} - \frac{\partial L_e}{\partial x_1^p} U_e}{U_e^2}.
$$

(41)

B. Proof of Theorem 1

Theorem 2: If the gradient of $f(x)$, denoted as $\nabla f(x)$, is Lipschitz continuous, i.e.,

$$
\|\nabla f(x_1) - \nabla f(x_2)\| \leq L \|x_1 - x_2\|_2,
$$

(45)

the function $g(x)$ is positive and bounded, i.e., $0 < g(x) \leq M$, and the learning rate satisfies $\alpha < \frac{1}{\nabla f(x)\cdot g(x)}$, the update update rule,

$$
x_{k+1} = x_k - \alpha g(x_k) \nabla f(x_k),
$$

(46)

will make $f(x)$ steadily decrease.

Proof 1: From Eq. (45), we can deduce that

$$
f(x) \leq f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \|x - x_k\|_2^2.
$$

(47)
It is known $\alpha < \frac{1}{2M}$ and $0 < g(x_k) \leq M$, hence we have
\[
f(x) \leq f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2\alpha M} \| x - x_k \|_2^2 \\
\leq f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2\alpha g(x_k)} \| x - x_k \|_2^2.
\]

The right side of Eq. (48) can be further equivalently reformulated as
\[
P(x; x_k) = f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2\alpha g(x_k)} \| x - x_k \|_2^2 \\
= f(x_k) + \frac{1}{2\alpha g(x)} \| x - (x_k - \alpha g(x_k) \nabla f(x_k)) \|_2^2 \\
- \frac{\alpha g(x_k)}{2} \| \nabla f(x_k) \|_2^2.
\]

It is easy to know $x_{k+1} = x_k - \alpha g(x_k) \nabla f(x_k)$ in Eq. (46) is the minimal point of Eq. (49), and then we obtain
\[
f(x_{k+1}) \leq P(x_{k+1}; x_k) \leq P(x_k; x_k) = f(x_k).
\]

it indicates $f(x)$ will decrease monotonically via the update rule in Eq. (30), and $f(x)$ we arrive the conclusion. □

C. Experimental Examples

Figure 8 shows test examples of the VOC2007 test set trained using Smooth-$\ell_1$ Loss, GIoU Loss and the proposed Smooth-EIoU Loss with Faster-RCNN with Res50 backbone and FPN architecture. The visual results indicate that the Smooth-IoU Loss can help to generate more precise bounding boxes, which verifies the conclusions in Section 2. Additionally, comparing with Smooth-$\ell_1$ Loss, GIoU Loss is more helpful to better localization, which is consistent with the results in [19].

Hanyang Peng received a B.S. degree in measurement and control technology from the Northeast University of China, Shenyang, China, in 2008, an M.E. degree in detection technology and automatic equipment from the Tianjin University of China, Tianjin, China, in 2010, and a Ph.D. degree in pattern recognition and intelligence systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2017. He is currently with Southern University of Science and Technology, Shenzhen, China. His current research interests include computer vision, machine learning, deep learning and optimization.

Shiqi Yu is currently an associate professor in the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. He received his B.E. degree in computer science and engineering from the Chu Kochen Honors College, Zhejiang University in 2002, and Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences in 2007. He worked as an assistant professor and an associate professor at Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences from 2007 to 2010, and as an associate professor at Shenzhen University from 2010 to 2019. His research interests include computer vision, pattern recognition and artificial intelligence.
Fig. 8: Some test examples of the VOC2007 test set with Faster-RCNN with Res50 backbone and FPN architecture trained using Smooth-$\ell_1$ Loss, GIoU Loss and Smooth-EIoU Loss (left to right). Ground-truth boxes are shown with solid lines and the predicted boxes are displayed with dashed lines.