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

Hanyang Peng and Shiqi Yu[ID], *Member, IEEE*

Abstract—Four-variable-independent-regression localization losses, such as Smooth-ℓ_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-ℓ_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_{90}.

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]–[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.
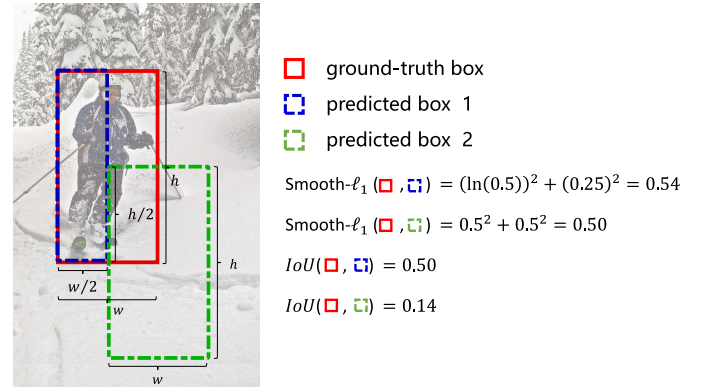
Fig. 1. Illustration of the *misalignment* between Smooth-ℓ_1 Loss and the metric IoU. Although the predicted box 1 matches the ground-truth box better than the predicted 2 (IoU: 0.50 vs 0.14), Smooth-ℓ_1 Loss of the ground-truth box and the predicted box 1 is larger than that of the ground-truth box and the predicted box 2 (Smooth-ℓ_1 Loss: 0.54 vs 0.50).

Constructing CNN architectures have made great strides in the past years [7]–[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 [18], 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-ℓ_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 oscillation 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 cannot 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 inferencing stage.

II. RELATED WORK

A. 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]. Reference [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.

B. 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 ℓ_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 ℓ_2 loss using in R-CNN with Smooth-ℓ_1 loss [7]. The localization loss of the latter CNN based detectors mostly follow Smooth-ell_1 loss with no or little change [8], [9], [13], [27], [30]. However, as illustrated in Section 1 and Fig 1, there is misalignment between Smooth-ℓ_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/focal 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

$$x_1 = \max\left(x_1^t,\ x_1^p\right), \tag{1}$$
$$y_1 = \max\left(y_1^t,\ y_1^p\right), \tag{2}$$
$$x_2 = \min\left(x_2^t,\ x_2^p\right), \tag{3}$$
$$y_2 = \min\left(y_2^t,\ y_2^p\right), \tag{4}$$
$$I_{\text{std}} = (x_2 - x_1)(y_2 - y_1), \tag{5}$$
$$S_t = (x_2^t - x_1^t)(y_2^t - y_1^t), \tag{6}$$
$$S_p = (x_2^p - x_1^p)(y_2^p - y_1^p), \tag{7}$$
$$U_{\text{std}} = S_t + S_p - I_{\text{std}}, \tag{8}$$
$$SIoU = \frac{I_{\text{std}}}{U_{\text{std}}}. \tag{9}$$

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.[1]

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.(1-4), the extended intersection (I_e) is

$$x_0 = \min\left(x_1^t,\ x_1^p\right) \tag{10}$$
$$y_0 = \min\left(y_1^t,\ y_1^p\right) \tag{11}$$
$$x_{\min} = \min\left(x_1,\ x_2\right) \tag{12}$$
$$y_{\min} = \min\left(y_1,\ y_2\right) \tag{13}$$
$$x_{\max} = \max\left(x_1,\ x_2\right) \tag{14}$$
$$y_{\max} = \max\left(y_1,\ y_2\right) \tag{15}$$
$$
\begin{aligned}
I_e &= S_1 + S_2 + S_3 + S_4 \\
&= (x_2 - x_0)(y_2 - y_0) + (x_{\min} - x_0)(y_{\min} - y_0) \\
&\quad -(x_1 - x_0)(y_{\max} - y_0) - (x_{\max} - x_0)(y_1 - y_0),
\end{aligned} \tag{16}
$$

where we define $I_e = S_1 + S_2 - S_3 - S_4$, 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 $(x_1, y_{\max})$; S_4 is area of the rectangle with top-left corner (x_0, y_0) and bottom-right $(x_{\max}, y_1)$.

---

[1] Actually, $SIoU$ 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 I.
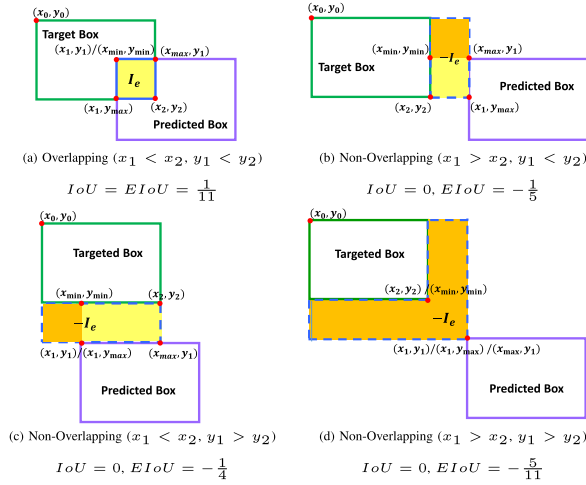
Fig. 2. Illustration the difference between $EIoU$ and $SIoU$. It is known $IoU = \frac{I}{S_t + S_p - I}$ and S_t and S_p are fixed, so the differences between I_e and I_{std} are the key. From Eq.(10-16), we know $I_e = S_1 + S_2 - S_3 - S_4$, where S_1 is area of the rectangle with top-left corner (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 $(x_{\min}, y_{\min})$; S_3 is area of the rectangle with top-left corner (x_0, y_0) and bottom-right $(x_1, y_{\max})$; S_4 is area of the rectangle with top-left corner (x_0, y_0) and bottom-right $(x_{\max}, y_1)$. It is known the standard $I_{std} = S_0$, where S_0 is the area of the rectangle with top-left corner (x_1, y_1) 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 exactly equivalent to the standard I_{std}. When two boxes are not overlapping, there are three situations shown in (b) with $x_1 > x_2$ and $y_1 < y_2$, (c) with $x_1 < x_2$ and $y_1 > y_2$ and (d) with $x_1 > x_2$ and $y_1 > y_2$. In this case, I_e become *negative*. Moreover, unlike I_{std} that keeps *constant* 0, the further two boxes are mutually separated, the smaller the value of I_e is, which conforms to human's intuition better and make gradients of I_e *non-zero*. Note that light yellow regions for I_e in (a), (b) and (c) are one-fold areas and deep yellow region in (b), (c) and (d) are two-fold areas.

We enumerate all the four situations whether two boxes are overlapping or not overlapping for the proposed I_e 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

$$
\begin{aligned}
I_e &= (x_{\max} - x_0)(y_{\max} - y_0) + (x_{\min} - x_0)(y_{\min} - y_0) \\
&\quad - (x_{\min} - x_0)(y_{\max} - y_0) - (x_{\max} - x_0)(y_{\min} - y_0) \\
&= (x_{\max} - x_{\min})(y_{\max} - y_{\min}) \\
&= (x_2 - x_1)(y_2 - y_1) \\
&> 0.
\end{aligned}
\tag{17}
$$

(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

$$
\begin{aligned}
I_e &= (x_{\min} - x_0)(y_{\max} - y_0) + (x_{\min} - x_0)(y_{\min} - y_0) \\
&\quad - (x_{\max} - x_0)(y_{\max} - y_0) - (x_{\max} - x_0)(y_{\min} - y_0) \\
&= (x_{\min} - x_{\max})(y_{\max} - y_0) + (x_{\min} - x_{\max})(y_{\min} - y_0) \\
&< 0.
\end{aligned}
\tag{18}
$$

(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 $x_{\min} = x_1$,
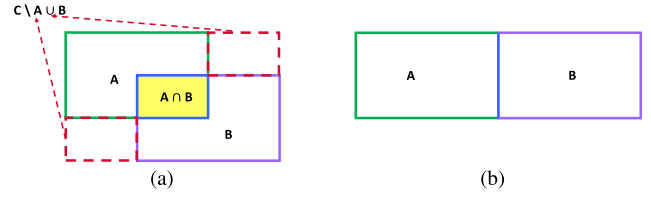

Fig. 3. A *counter-intuitive* case for $GIoU$. (a) We first compute the standard IoU: $SIoU = \frac{A \bigcap B}{A \bigcup B} = \frac{0.25}{2 - 0.25} = \frac{1}{7}$, and then compute $GIoU = SIoU - \frac{C \setminus (A \bigcup B)}{C} = \frac{1}{7} - \frac{0.5}{2.25} = -\frac{5}{63}$, while (b) two boxes are just attached, $GIoU = SIoU = 0$. The value of $GIoU$ in (a) is *smaller* than that in (b), which is inconsistent with the fact that two boxes match better in (a) than in (b).

$x_{\max} = x_2$, $y_{\min} = y_2$ and $y_{\max} = y_1$, and then

$$
\begin{aligned}
I_e &= (x_{\max} - x_0)(y_{\min} - y_0) + (x_{\min} - x_0)(y_{\min} - y_0) \\
&\quad - (x_{\min} - x_0)(y_{\max} - y_0) - (x_{\max} - x_0)(y_{\max} - y_0) \\
&= (x_{\max} - x_0)(y_{\min} - y_{\max}) + (x_{\min} - x_0)(y_{\min} - y_{\max}) \\
&< 0.
\end{aligned}
\tag{19}
$$

(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

$$
\begin{aligned}
I_e &= (x_{\min} - x_0)(y_{\min} - y_0) + (x_{\min} - x_0)(y_{\min} - y_0) \\
&\quad - (x_{\max} - x_0)(y_{\max} - y_0) - (x_{\max} - x_0)(y_{\max} - y_0) \\
&= 2\left((x_{\min} - x_0)(y_{\min} - y_0) - (x_{\max} - x_0)(y_{\max} - y_0)\right) \\
&< 0.
\end{aligned}
\tag{20}
$$

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.

Analogous to $SIoU$ in Eq.(6-9), we obtain $EIoU$ based on I_e in Eq. (16):

$$
U_e = S_t + S_p - I_e,
\tag{21}
$$

$$
EIoU = \frac{I_e}{U_e},
\tag{22}
$$

U_e is a function of I_e and always larger than zero, so characteristics of $EIoU$ are similar to that of I_e, which are summarized as follows:

• When two boxes are attached, $EIoU$ *exactly equivalent* to the standard IoU and always larger than zero.
• When the two boxes are detached, $EIoU$ is smaller than zero and decreases with distance of two boxes, so that gradient descent algorithms can be employed to train the predicted box to approach the targeted box until matched.

1) Differences From GIoU: Both $GIoU$ [19] 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

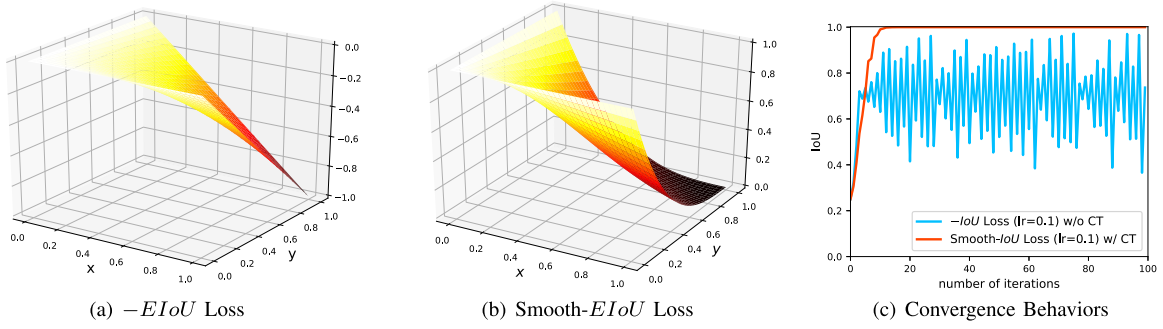|		
(a) $-EIoU$ Loss	(b) Smooth-$EIoU$ Loss	(c) Convergence Behaviors

Fig. 4. Visualization of $-EIoU$ and Smooth-$EIoU$ Loss constructed from $-EIoU$ with CT in Eq. (23) when the targeted box is fixed with $(0, 0, 1, 1)$ and the predicted box varies with $(0, 0, x, y)$. (a) shows the value space of $-EIoU$. $-EIoU$ is smaller than zero and not smooth in the neighbourhood of the minimum, and its gradient at the minimum is *non-zero*; (b) shows the value space of Smooth-$EIoU$ Loss. After employing CT, Smooth-$EIoU$ Loss is larger than zeros and smooth everywhere, and its gradients are gradually close to zero when it approaches the minimum. (c) Convergence behaviours of $-EIoU$ and Smooth-$EIoU$ Loss when the targeted box is fixed with $(0, 0, 1, 1)$ and the predicted box is initialized with $(0, 0, 0.5, 0.5)$. Due to the non-zero gradients of $-EIoU$ at the minimum and non-smoothness in its neighbourhood, $-EIoU$ Loss *oscillates* dramatically and cannot converge. In contrast, Smooth-$EIoU$ Loss quickly converges to the optimum.

Algorithm 1 $GIoU$ in [19]

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: $SIoU = \frac{A \bigcap B}{A \bigcup B}$
3. Compute $GIoU$: $GIoU = SIoU - \frac{C \backslash (A \bigcup B)}{C}$

---

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.

C. Covexification Technique (CT)

Loosely speaking, any a decreasing function w.r.t. IoU can be treated as a localization loss, such as $\frac{1}{IoU}$, $-IoU$ 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 of them at the minimum 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 train. 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 the 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 follows,[2] *i.e.*,

$$\mathcal{L}_{\text{Smooth}-\text{EIoU}} = \left(1 - \frac{I_e}{U_e}\right)^2 \tag{23}$$

CT scan 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.*,

$$\frac{\partial \mathcal{L}_{\text{Smooth}-\text{EIoU}}}{\partial z} = -\left(1 - \frac{I_e}{U_e}\right) \frac{\partial \left(\frac{I_e}{U_e}\right)}{\partial z} \tag{24}$$

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, $\frac{\partial \mathcal{L}_{\text{Smooth}-\text{EIoU}}}{\partial z}$ 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 train.

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 4 (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 4 (c). 4 (c) shows the convergence behavior of $-EIoU$ and Smooth-$EIoU$ Loss when the predicted bounding box starts with the initial value $(0, 0, 0.5, 0.5)$.

---

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

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)

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.,

$$\frac{\partial I_e}{\partial x_1^p} = \begin{cases} y_{\min} - y_{\max}, & \text{if } x_1^p \geq x_1^t \text{ and } x_1 \leq x_2, \\ 2y_0 - y_{\max} - y_1, & \text{if } x_1^p \geq x_1^t \text{ and } x_1 > x_2, \\ 0, & \text{if } x_1^p < x_1^t. \end{cases} \quad (25)$$

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

$$\frac{\partial U_e}{\partial x_1^p} = (y_1^p - y_2^p) - \frac{\partial I_e}{\partial x_1^p}. \quad (26)$$

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

$$\frac{\partial L_{\text{Smooth}-\text{EIoU}}}{\partial x_1^p} = 2\left(1 - \frac{I_e}{U_e}\right) \frac{I_e \frac{\partial U_e}{\partial x_1^p} - \frac{\partial I_e}{\partial x_1^p} U_e}{U_e^2}. \quad (27)$$

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 $I_e \propto s$, $U_e \propto s^2$, $\frac{\partial I_e}{\partial x_1^p} \propto s$ and $\frac{\partial U_e}{\partial x_1^p} \propto s$ 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 $\frac{\partial L_{\text{Smooth}-\text{EIoU}}}{\partial x_1^p} \propto \frac{1}{s}$, 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 targeted 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 ℓ_2 Loss. To achieve this goal, we change the update rule for variables of $EIoU$. We take x_1^p for example, i.e.,

$$x_{1_k}^p = x_{1_{k-1}}^p - 2\alpha \frac{\partial L_{\text{GI}-\text{IoU}}}{\partial x_1^p} U_e$$

$$= x_{1_{k-1}}^p - 2\alpha \left(1 - \frac{I_e}{U_e}\right) \frac{I_e \frac{\partial U_e}{\partial x_1^p} - \frac{\partial I_e}{\partial x_1^p} U_e}{U_e} \quad (28)$$

where k is the number of iterations and α 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.
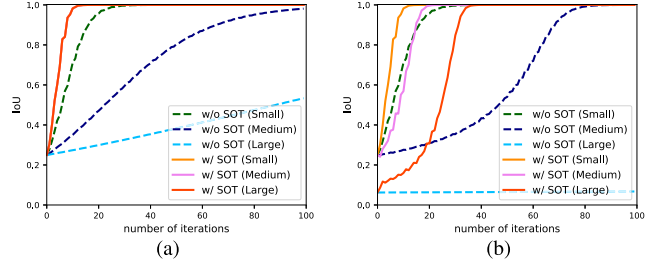

Fig. 5. Convergence of Smooth-$EIoU$ Loss optimized with/without SOT: (a) comparisons when the size of the targeted box and predicted box proportionally varies: the targeted box are fixed with $(0, 0, 1, 1)$ (small), $(0, 0, 2, 2)$ (medium) and $(0, 0, 4, 4)$ (large), and the initial value of the predicted is proportionally 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), $(0, 0, 2, 2)$ (medium) and $(0, 0, 4, 4)$ (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 set as $(0, 0, 4, 4)$, it is even *trapped* and cannot move to the target box.

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

$$\|\nabla f(x_1) - \nabla f(x_2)\| \leq L\|x_1 - x_2\|_2, \quad (29)$$

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,

$$x_{k+1} = x_k - \alpha g(x_k)\nabla f(x_k), \quad (30)$$

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. (23) 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 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 proportionally 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 above. When only the size of the predicted box varies, the targeted box are fixed with $(0, 0, 1, 1)$, and the

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-l_1	$SIoU$	$GIoU$	$EIoU$	CT	SOT	IoU Head	AP
①	✓	-	-	-	-	-	-	45.5 (Smooth-l_1 Loss, Baseline)
②	-	✓	-	-	-	-	-	46.6 (Standard IoU Loss)
③	-	-	✓	-	-	-	-	46.9 ($GIoU$ Loss)
④	✓	-	-	-	-	-	✓	46.2 (Smooth-l_1 Loss with IoU Head)
⑤	-	-	-	✓	-	-	-	47.5 ($EIoU$ Loss)
⑥	-	-	-	✓	✓	-	-	47.9 ($EIoU$ Loss with CT)
⑦	-	-	-	✓	✓	✓	-	48.2 ($EIoU$ Loss with CT, SOT)
⑧	-	-	-	✓	✓	✓	✓	49.7 ($EIoU$ Loss with CT, SOT and IoU Head)

initial value of the predicted is set as $(0, 0, 0.5, 0.5)$ (small), $(0, 0, 2, 2)$ (medium) and $(0, 0, 4, 4)$ (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 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-$EIoU$ 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.*,

$$q_p(x) = \text{Sigmoid}(x), \tag{31}$$

$$\mathcal{L}_{KL} = q_g \log \frac{q_g}{q_p(x)} + (1 - q_g) \log \frac{1 - q_g}{1 - q_p(x)}, \tag{32}$$

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-$EIoU$ 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.

1) Differences From IoU-Net: [27] pioneeringly 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
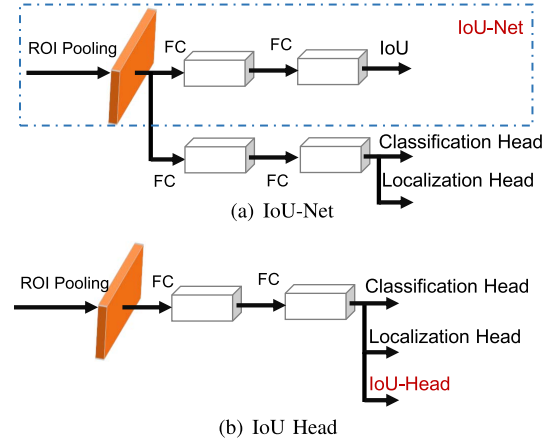

(a) IoU-Net

(b) IoU Head

Fig. 6.   The network architectures of IoU-Net and IoU Head.

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 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 2.

Algorithm 2 Training $EIoU$ Loss

Input: the anchor box $(x_1^a, y_1^a, x_2^a, y_2^a)$, the target box $(x_1^t, y_1^t, x_2^t, y_2^t)$ and the CNN predicted normalized difference value $(x_{1_0}^d, y_{1_0}^d, x_{2_0}^d, y_{2_0}^d)$

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

Compute $S = \sqrt{(x_2^a - x_1^a)(x_2^a - x_1^a)}$

Compute $(x_1^{an}, y_1^{an}, x_2^{an}, y_2^{an}) = (\frac{x_1^a}{S}, \frac{y_1^a}{S}, \frac{x_2^a}{S}, \frac{y_2^a}{S})$

Compute $(x_1^{tn}, y_1^{tn}, x_2^{tn}, y_2^{tn}) = (\frac{x_1^t}{S}, \frac{y_1^t}{S}, \frac{x_2^t}{S}, \frac{y_2^t}{S})$

while *not convergence* **do**

    Compute $(x_{1_k}^p, y_{1_k}^p, x_{2_k}^p, y_{2_k}^p) = (x_{1_k}^d, y_{1_k}^d, x_{2_k}^d, y_{2_k}^d)$
    $+(x_1^{an}, y_1^{an}, x_2^{an}, y_2^{an})$

    Using Eq. (1)- (4), (10)- (23) to compute $\mathcal{L}_{\text{GI-IoU}}$

    Using Eq. (25)- (27) to compute $(\frac{\partial \mathcal{L}}{\partial x_{1_k}^p}, \frac{\partial \mathcal{L}}{\partial y_{1_k}^p}, \frac{\partial \mathcal{L}}{\partial x_{2_k}^p}, \frac{\partial \mathcal{L}}{\partial y_{2_k}^p})$

    $(\frac{\partial \mathcal{L}}{\partial x_{1_k}^d}, \frac{\partial \mathcal{L}}{\partial y_{1_k}^d}, \frac{\partial \mathcal{L}}{\partial x_{2_k}^d}, \frac{\partial \mathcal{L}}{\partial y_{2_k}^d}) = (\frac{\partial \mathcal{L}}{\partial x_{1_k}^p}, \frac{\partial \mathcal{L}}{\partial y_{1_k}^p}, \frac{\partial \mathcal{L}}{\partial x_{2_k}^p}, \frac{\partial \mathcal{L}}{\partial y_{2_k}^p})$

    Using Eq. (28) to update $(x_{1_{k+1}}^d, y_{1_{k+1}}^d, x_{2_{k+1}}^d, y_{2_{k+1}}^d)$

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. No other data augmentation except of 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 $EIoU$, CT, SOT and IoU Head, and the results are reported in Table I.

As shown in Table I, with the standard $SIoU$ based loss replacing the baseline Smooth-ℓ_1 Loss, the performance is improved to some extents (+1.1% mAP, comparing ① and ②). Substituting $SIoU$ with $GIoU$ further boost the performance with scores +0.3% mAP (comparing ② and ③), which is consistence with the results in Table 5 in [19]. Comparing with $GIoU$, individually equipping $EIoU$ can bring more substantial improvement(+0.9 % mAP, ② and ⑤), which indicates $EIoU$ may be more piratically powerful than $GIoU$. 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, $EIoU$ Loss with IoU Head can generate better cooperativity than Smooth-l_1 Loss with IoU Head(+1.5% mAP vs +0.7% mAP, comparing ⑦, ⑧ and comparing ①, 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 $EIoU$, CT, SOT and IoU Head yields significant gains, which is 4.2% higher than the baseline Smooth-l_1 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-ℓ_1 Loss and the related $GIoU$ Loss [19] and $CIoU$ [20] Loss in this subsection. For simplicity, our systematical method is referred to as $EIoU$ 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-III. 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-ℓ_1 Loss and $GIoU$ Loss, $EIoU$ Loss in one-stage and two-stage detectors can steadily yield gains on PASCAL VOC and MS COCO. Specifically, for the baseline Smooth-ℓ_1 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 $GIoU$ Loss, $EIoU$ 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-ℓ_1 Loss. However, when the threshold grows higher, $EIoU$ Loss gradually outperforms Smooth-ℓ_1 Loss and $GIOU$ Loss,. Especially at AP$_{90}$, comparing with Smooth-ℓ_1 Loss, $EIoU$ Loss improves 8.2% on PASCAL VOC dataset and 5.3% on MS COCO dataset . The reason for it is $EIoU$ Loss can help a detector to predict more accurate bounds than Smooth-ℓ_1 Loss. It is known there is a gap between Smooth-ℓ_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-ℓ_1 Loss decreases quicker than $EIoU$ Loss as two boxes are gradually matched, so during training Smooth-ℓ_1 Loss commonly gives less attention to better matched pair-boxes. Therefore, comparing to Smooth-ℓ_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-ℓ_1 Loss

TABLE II

COMPARISONS OF AVERAGE PRECISION(AP) OF SMOOTH-ℓ_1 LOSS, $GIoU$ LOSS AND $EIoU$ LOSS ATTACHING TO RETINANET AND FASTER-RCNN 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

loss	Net	mAP	AP_{50}	AP_{75}	AP_{90}	AP_S	AP_M	AP_L
Smooth-ℓ_1 Loss (Baseline) [9]	RetinaNet	44.2	70.5	47.5	25.0	9.8	28.3	54.3
$GIoU$ Loss [19]	RetinaNet	45.2	70.0	48.2	28.8	9.9	29.1	55.5
$CIoU$ Loss [20]	RetinaNet	45.9	70.3	50.1	30.3	10.1	30.2	55.9
Ours	RetinaNet	**46.4**	**71.1**	**49.1**	**32.2**	**10.5**	**31.4**	**56.3**
Smooth-ℓ_1 Loss (Baseline) [9]	Faster-RCNN+FPN	45.5	72.6	49.8	25.2	10.0	29.5	55.6
$GIoU$ Loss [19]	Faster-RCNN+FPN	46.9	73.1	50.8	28.6	9.6	31.0	57.2
$CIoU$ Loss [20]	Faster-RCNN+FPN	48.0	73.5	51.2	28.8	9.8	31.8	57.9
Ours	Faster-RCNN+FPN	**49.7**	**73.7**	**54.1**	**33.4**	**11.9**	**32.8**	**59.8**

TABLE III

COMPARISONS OF AVERAGE PRECISION(AP) OF OF SMOOTH-ℓ_1 LOSS, $GIoU$ LOSS AND $EIoU$ LOSS ATTACHING TO RETINANET AND FASTER-RCNN WITH RES50 + FPN AS THE BACKBONE. MODELS ARE TRAINED ON THE UNION SET OF COCO_2017_TRAIN. THE RESULTS ARE REPORTED ON THE SET OF COCO_2017_VAL

loss	Net	mAP	AP_{50}	AP_{75}	AP_{90}	AP_S	AP_M	AP_L
Smooth-ℓ_1 Loss (Baseline) [9]	RetinaNet	35.7	54.7	38.5	22.8	19.5	39.9	47.5
$GIoU$ Loss [19]	RetinaNet	36.2	54.5	38.9	24.4	19.6	40.3	48.3
$CIoU$ Loss [20]	RetinaNet	36.7	54.7	39.3	25.0	20.1	40.9	48.9
Ours	RetinaNet	**37.1**	**54.8**	**39.6**	**25.7**	**20.6**	**41.3**	**49.2**
Smooth-ℓ_1 Loss (Baseline) [9]	Faster-RCNN+FPN	36.7	58.5	39.6	21.2	21.1	39.8	48.1
$GIoU$ Loss [19]	Faster-RCNN+FPN	37.8	58.4	41.0	24.8	21.2	40.9	49.8
$CIoU$ Loss [20]	Faster-RCNN+FPN	38.4	58.3	41.5	25.2	21.2	41.3	50.2
Ours	Faster-RCNN+FPN	**39.0**	**58.7**	**42.1**	**26.5**	**22.3**	**41.7**	**50.7**

on PASCAL VOC dataset with Faster-RCNN, but Smooth-ℓ_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 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-ℓ_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

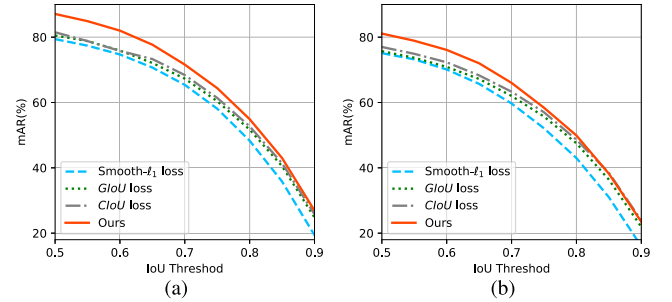We evaluate $EIoU$ Loss attached to FPN on the MS COCO 2019 $test - dev$ set with 180k iterations and compare the

Fig. 7. IoU threshold against Average Recall(AR) of Faster RCNN + FPN with Smooth-ℓ_1 Loss, $GIoU$ Loss and our approach on (a) PASCAL VOC and (b)MS COCO. Models are trained on (a) the union set of VOC 2007 trainval and VOC 2012 trainval and (b) the set of COCO 2017 train. The results are demonstrated on (a) the set of VOC 2007 test and (b) the COCO 2017 val.

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 ResNeX-64 × 4d-101+FPN achieves 44.1% mAP, which surpasses the counterparts in the Table IV 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.

TABLE IV
PERFORMANCE OF STATE-OF-THE-ART DETECTORS ON THE SET OF COCO TEST-DEV. OUR MODEL
IS TRAINED ON THE SET OF COCO_2017_train

Method	Backbone	Year	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv3 [11]	DarkNet-53	2018	33.0	57.9	34.4	18.3	35.4	41.9
SSD513 [12]	ResNet-101	2016	31.2	50.4	33.3	10.2	34.5	49.8
RetinaNet800[13]	ResNeXt-101	2017	39.1	59.1	42.3	21.8	42.7	50.2
DSSD513 [31]	ResNet-101	2017	33.2	53.3	35.2	13.0	35.4	51.1
RefineDet512[33]	ResNet-101	2018	36.4	57.5	39.5	16.6	39.9	51.4
CornerNet511[34]	Hourglass-104	2018	40.5	56.5	43.1	19.4	42.7	53.9
CenterNet [34]	Hourglass-104	2019	42.1	61.1	45.9	24.1	45.5	52.8
FCOS[36]	ResNeXt-101	2019	43.2	62.8	46.6	26.5	46.2	53.3
Faster-R-CNN +++ [6]	ResNet-101	2016	34.9	55.7	37.4	15.6	38.7	50.9
Faster-RCNN w FPN [9]	ResNet-101	2016	36.2	59.1	39.0	18.2	39.0	48.2
Mask R-CNN [1]	ResNeXt-101	2017	39.8	62.3	43.4	22.1	43.2	51.2
DetNet [24]	DetNet	2018	40.3	62.1	43.8	23.6	42.6	50.0
IoU-Net [27]	ResNet-101	2018	40.6	59.0	-	-	-	-
TridenNet w *2fc* [29]	ResNet-101	2019	42.0	63.5	45.5	24.9	47.0	56.9
Grid R-CNN [28]	ResNeXt-101	2019	43.2	63.0	46.6	25.1	46.5	55.2
Ours	ResNet-101	2019	42.2	61.8	46.1	24.4	45.2	55.4
Ours	ResNeXt-101	2019	**44.1**	**63.7**	**47.6**	**26.8**	**47.6**	**57.1**

VI. CONCLUSION AND DISCUSSION

Smooth-ℓ_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]–[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 $\frac{1}{IoU}$, $-\ln(IoU)$ 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 (factional 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.

APPENDIX A
GRADIENTS OF SMOOTH-EIOU LOSS

In this section, we will deduce all the partial derivative of Smooth-$EIoU$ loss in Eq.(23) w.r.t. x_1^p, y_1^p, x_2^p and y_2^p. We first compute the partial derivative, *i.e.*,

$$\frac{\partial I_e}{\partial x_1^p} = \begin{cases} y_{\min} - y_{\max}, & \text{if } x_1^p \geq x_1^t \text{ and } x_1 \leq x_2, \\ 2y_0 - y_{\max} - y_1, & \text{if } x_1^p \geq x_1^t \text{ and } x_1 > x_2, \\ 0, & \text{if } x_1^p < x_1^t. \end{cases} \quad (33)$$

$$\frac{\partial I_e}{\partial y_1^p} = \begin{cases} x_{\min} - x_{\max}, & \text{if } y_1^p \geq y_1^t \text{ and } y_1 \leq y_2, \\ 2x_0 - x_{\max} - x_1, & \text{if } y_1^p \geq y_1^t \text{ and } y_1 > y_2, \\ 0, & \text{if } y_1^p < y_1^t. \end{cases} \quad (34)$$

$$\frac{\partial I_e}{\partial x_2^p} = \begin{cases} y_2 - y_1, & \text{if } x_2^p \leq x_2^t \text{ and } x_1 \leq x_2, \\ y_2 + y_{\min} - 2y_0, & \text{if } x_2^p \leq x_2^t \text{ and } x_1 > x_2, \\ 0, & \text{if } x_2^p > x_2^t. \end{cases} \quad (35)$$

$$\frac{\partial I_e}{\partial y_2^p} = \begin{cases} x_2 - x_1, & \text{if } y_2^p \leq y_2^t \text{ and } y_1 \leq y_2, \\ x_2 + x_{\min} - 2x_0, & \text{if } y_2^p \leq y_2^t \text{ and } y_1 > y_2, \\ 0, & \text{if } y_2^p > y_2^t. \end{cases} \quad (36)$$

And then we compute the the partial derivative of U_e

$$\frac{\partial U_e}{\partial x_1^p} = (y_1^p - y_2^p) - \frac{\partial I_e}{\partial x_1^p}, \quad (37)$$

$$\frac{\partial U_e}{\partial y_1^p} = (x_1^p - x_2^p) - \frac{\partial I_e}{\partial y_1^p}, \quad (38)$$

$$\frac{\partial U_e}{\partial x_2^p} = (y_2^p - y_1^p) - \frac{\partial I_e}{\partial x_2^p}, \quad (39)$$

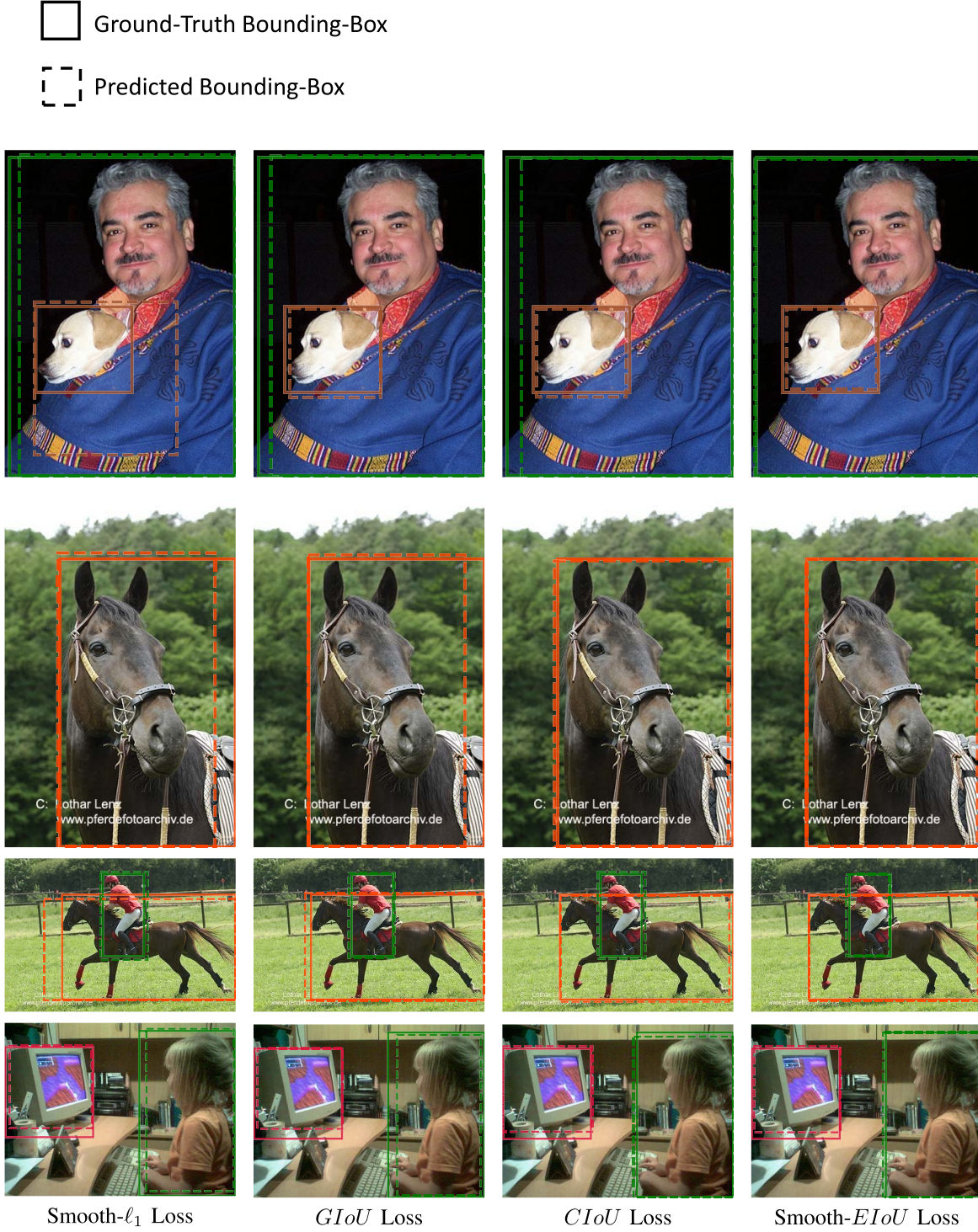$$\frac{\partial U_e}{\partial y_2^p} = (x_2^p - x_1^p) - \frac{\partial I_e}{\partial y_2^p}. \quad (40)$$

□ Ground-Truth Bounding-Box

⌐¬ Predicted Bounding-Box

| Smooth-ℓ_1 Loss | $GIoU$ Loss | $CIoU$ Loss | Smooth-$EIoU$ Loss |

Fig. 8. Some test examples of the VOC2007_test_set with Faster-RCNN with Res50 backbone and FPN architecture trained using Smooth-ℓ_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.

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

$$\frac{\partial L_{\text{Smooth}-\text{EIoU}}}{\partial x_1^p} = 2\left(1 - \frac{I_e}{U_e}\right)\frac{I_e\frac{\partial U_e}{\partial x_1^p} - \frac{\partial I_e}{\partial x_1^p}U_e}{U_e^2}, \quad (41)$$

$$\frac{\partial L_{\text{Smooth}-\text{EIoU}}}{\partial y_1^p} = 2\left(1 - \frac{I_e}{U_e}\right)\frac{I_e\frac{\partial U_e}{\partial y_1^p} - \frac{\partial I_e}{\partial y_1^p}U_e}{U_e^2}, \quad (42)$$

$$\frac{\partial L_{\text{Smooth}-\text{EIoU}}}{\partial x_2^p} = 2\left(1 - \frac{I_e}{U_e}\right)\frac{I_e\frac{\partial U_e}{\partial x_2^p} - \frac{\partial I_e}{\partial x_2^p}U_e}{U_e^2},$$

$$(43)$$

$$\frac{\partial L_{\text{Smooth}-\text{EIoU}}}{\partial y_2^p} = 2\left(1 - \frac{I_e}{U_e}\right)\frac{I_e\frac{\partial U_e}{\partial y_2^p} - \frac{\partial I_e}{\partial y_2^p}U_g}{U_g^2}.$$

$$(44)$$

APPENDIX B
PROOF OF THEOREM 1

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

$$\|\nabla f(x_1) - \nabla f(x_2)\| \le L\|x_1 - x_2\|_2, \tag{45}$$

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

$$x_{k+1} = x_k - \alpha g(x_k)\nabla f(x_k), \tag{46}$$

will make $f(x)$ steadily decrease.

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

$$f(x) \le f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2}\|x - x_k\|_2^2. \tag{47}$$

It is known $\alpha < \frac{1}{LM}$ and $0 < g(x_k) \le M$, hence we have

$$
\begin{aligned}
f(x) &\le f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2\alpha M}\|x - x_k\|_2^2 \\
&\le f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2\alpha g(x_k)}\|x - x_k\|_2^2.
\end{aligned} \tag{48}
$$

The right side of Eq. (48) can be further equivalently reformulated as

$$
\begin{aligned}
P(x; x_k) &= f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2\alpha g(x_k)}\|x - x_k\|_2^2 \\
&= f(x_k) + \frac{1}{2\alpha g(x)}\|x - (x_k - \alpha g(x_k)\nabla f(x_k))\|_2^2 \\
&\quad - \frac{\alpha g(x_k)}{2}\|\nabla f(x_k)\|_2^2
\end{aligned} \tag{49}
$$

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

$$f(x_{k+1}) \le P(x_{k+1}; x_k) \le P(x_k; x_k) = f(x_k). \tag{50}$$

it inidicates $f(x)$ will decrease monotonicly via the update rule in Eq. (30), and $f(x)$ we arrive the conclusion. $\square$

APPENDIX C
EXPERIMENTAL EXAMPLES

Fig. 8 shows test examples of the VOC2007_test_set trained using Smooth-ℓ_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-ℓ_1 Loss, $GIoU$ Loss is more helpful to better localization, which is consistent with the results in [19].
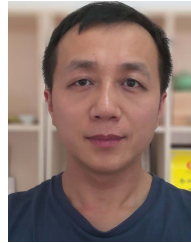
REFERENCES

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jul. 2017, pp. 2980–2988.

[2] P. O. Pinheiro, T. Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *Proc. Eur. Conf. Comput. Vis.*, 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 *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 153–164.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[5] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[7] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 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 Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[9] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[10] 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.

[11] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[12] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[14] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 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 Trans. Image Process.*, vol. 28, no. 1, pp. 265–278, Jan. 2019.

[16] G. Cheng, J. Yang, D. Gao, L. Guo, and J. Han, "High-quality proposals for weakly supervised object detection," *IEEE Trans. Image Process.*, 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 J. Photogramm. Remote Sens.*, vol. 159, pp. 296–307, Jan. 2020.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[19] 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.

[20] 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. (AAAI)*, 2020, pp. 12993–13000.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[22] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

[23] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[24] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "DetNet: A backbone network for object detection," 2018, *arXiv:1804.06215*. [Online]. Available: http://arxiv.org/abs/1804.06215

[25] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," 2016, *arXiv:1603.06937*. [Online]. Available: http://arxiv.org/abs/1603.06937

[26] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[27] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 784–799.

[28] X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan, "Grid R-CNN," 2018, *arXiv:1811.12030*. [Online]. Available: http://arxiv.org/abs/1811.12030

[29] Y. Li, Y. Chen, N. Wang, and Z.-X. Zhang, "Scale-aware trident networks for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6054–6063.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015, *arXiv:1506.02640*. [Online]. Available: http://arxiv.org/abs/1506.02640

[31] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," 2017, *arXiv:1701.06659*. [Online]. Available: http://arxiv.org/abs/1701.06659

[32] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "RON: Reverse connection with objectness prior networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5244–5252.

[33] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4203–4212.

[34] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.

[35] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," 2019, *arXiv:1903.00621*. [Online]. Available: http://arxiv.org/abs/1903.00621

[36] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," 2019, *arXiv:1904.01355*. [Online]. Available: http://arxiv.org/abs/1904.01355

[37] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*. [Online]. Available: http://arxiv.org/abs/1904.07850

[38] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," 2019, *arXiv:1904.08189*. [Online]. Available: http://arxiv.org/abs/1904.08189

[39] 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.

[40] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. (2018). *Detectron*. [Online]. Available: https://github.com/facebookresearch/detectron

Hanyang Peng received the B.S. degree in measurement and control technology from the Northeast University of China, Shenyang, China, in 2008, the M.E. degree in detection technology and automatic equipment from the Tianjin University of China, Tianjin, China, in 2010, and the 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 the Southern University of Science and Technology, Shenzhen. His current research interests include computer vision, machine learning, deep learning, and optimization.

Shiqi Yu (Member, IEEE) received the B.E. degree in computer science and engineering from the Chu Kochen Honors College, Zhejiang University, in 2002, and the 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 with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, from 2007 to 2010, and an Associate Professor with Shenzhen University from 2010 to 2019. He is currently an Associate Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. His research interests include computer vision, pattern recognition, and artificial intelligence.