IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received July 13, 2021, accepted July 22, 2021, date of publication July 26, 2021, date of current version August 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3100414

ICIoU: Improved Loss Based on Complete Intersection Over Union for Bounding Box Regression

XUFEI WANG[1,2] AND JEONGYOUNG SONG[2], (Member, IEEE)

[1]Key Laboratory of Industrial Automation, School of Mechanical Engineering, Shaanxi University of Technology, Hanzhong 723000, China
[2]Department of Computer Engineering, Pai Chai University, Daejeon 35345, South Korea

Corresponding author: Jeongyoung Song (jysong@pcu.ac.kr)

This work was supported in part by the Shaanxi Provincial Key Laboratory of Industrial Automation Research Program under Grant 18JS020.

ABSTRACT An object detector based on convolutional neural network (CNN) has been widely used in the field of computer vision because of its simplicity and efficiency. The average accuracy of CNN model detection results in the object detector is greatly affected by the loss function. The precision of the localization algorithm in the loss function is the main factor affecting the result. Based on the complete intersection over union (CIoU) loss function, an improved penalty function is proposed to improve the localization accuracy. Specifically, the algorithm more comprehensively considers matching bounding boxes between prediction with ground truth, using the proportional relationship of the aspect ratio from both bounding boxes. Under the same aspect ratio of the two bounding boxes, the influence factors of the prediction box on localization accuracy were considered. In this way, the function of the penalty function is strengthened, and localization accuracy of the network model improved. This loss function is called Improved CIoU (ICIoU). Experiments on the Udacity, PASCAL VOC, and MS COCO datasets have demonstrated the effectiveness of ICIoU in improving localization accuracy of network models by using the one-stage object detector YOLOv4. Compared with CIoU, the proposed ICIoU improved average precision (AP) by 0.57% and AP75 by 0.12% on Udacity, AP by 0.26% and AP75 by 1.28% on PASCAL VOC, and AP by 0.06% and AP75 by 0.65% on MS COCO.

INDEX TERMS Bounding box regression, localization accuracy, loss function, object detection.

I. INTRODUCTION

Object detection is one of the key problems in computer vision tasks. In recent years, convolutional neural networks (CNNs) have been increasingly applied in the field of computer vision [1]–[11]. When using CNNs to solve the problem of object detection, no matter whether a regression or classification problem, a loss function is indispensable. Loss functions are used to estimate the degree of inconsistency between the predicted value of a model and the real value. The main task of model training in the present work is to use the optimization method to find the model parameters corresponding to the minimization of the loss function. The loss function determines what the optimal value of the model is, so the performance of different object detectors is affected by the loss function. The loss function generally consists of bounding box regression and classification. The loss calculation of bounding box regression is the key step of object location, multiobject detection, target tracking, and instance-level segmentation. In terms of multiobject detection, compared with the traditional region proposal methods, a deep CNN has better performance advantages in predicting the bounding box of candidate objects. These networks include one-stage object detectors such as the YOLO series [3]–[6] and single shot multibox detector (SSD) [9], two-stage object detectors, such as series of the regions with CNN features (R-CNN) [11]–[14], and even multistage object detectors, such as cascade R-CNN [15]. In these networks, intersection over union (IoU) loss has become the most popular evaluation measurement algorithm for bounding box regression compared with focal loss and $\mathcal{L}_n$-norm (e.g. $\mathcal{L}_1$, $\mathcal{L}_2$) loss [16], [17]. However, IoU algorithm cannot detect the bounding box

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

regression problem when the bounding boxes of ground truth and prediction do not overlap. In order to compensate for the defects of IoU algorithm, generalized IoU (GIoU) [17], distance-IoU (DIoU), and complete IoU (CIoU) were proposed successively by increasing penalty terms [18]. The GIoU algorithm solved the bounding box regression problem when the bounding boxes of ground truth and prediction do not overlap, and the DIoU algorithm improved the convergence speed by directly minimizing the center distance between the bounding boxes of ground truth and prediction. The CIoU algorithm achieved the best state-of-the-art performance by simultaneously considering the three geometric measures including the center distance between the bounding boxes of ground truth and prediction. However, when the CIoU algorithm calculates the consistency of the aspect ratio of the bounding box for both ground truth and prediction, it will degenerate to DIoU when the aspect ratio of the bounding box of ground truth is equal to the aspect ratio of the bounding box of prediction. Based on this problem, in this paper an improved CIoU algorithm is proposed to avoid the aforementioned degradation.

The main contributions of this paper are the following.

(1) A method is proposed to calculate the penalty function using the ratio of the corresponding width and height of the bounding box for both ground truth and prediction, as a geometric metric that could better describe the regression of the boundary box and improve localization accuracy.

(2) Like CIoU, ICIoU could also be incorporated into the most advanced detection algorithms to achieve performance gains.

II. RELATED WORK

A. OBJECT DETECTION

Object detection algorithms using CNNs are mainly divided into two categories: one based on regression (Also known as one-stage) and the other on candidate region (Also known as two-stage). In terms of candidate-region-based research, Krizhevsky *et al.* proposed the AlexNet network and applied a CNN to image classification for the first time [1]. Girshick *et al.* proposed a R-CNN network, extracted candidate regions from input images through a selective search algorithm, and then extracted features from candidate boxes using AlexNet. Finally, support vector machine (SVM) classifiers and regressors were used to determine the category and location of objects [11]. Aiming at the problem of high complexity and long training time of the R-CNN algorithm, He *et al.* proposed a spatial pyramid pooling (SPP) network with ZF-5 as the backbone network, and the accuracy rate on the VOC2007 dataset increased to 59.2% [2]. Absorbing the ideas of SPP, Girshick proposed a Fast R-CNN network, and the mean average precision (mAP) of the FAST R-CNN with VGG-16 as the backbone network on VOC2007 dataset increased to 70.0% [12]. Ren *et al.* proposed the Faster R-CNN network, which reached a mAP of 73.2% on the VOC2007 dataset using VGG-16 as the backbone network, and the detection speed was increased from the 3 frames per second (FPS) of the Fast R-CNN to 7 FPS [13]. Dai *et al.* proposed a R-FCN network and realized a mAP of 79.5% on VOC2007 dataset [19]. He *et al.* proposed the Mask R-CNN network, which reached a mAP of 39.8% on the MS COCO dataset using Resnext-101 as the backbone network [14]. Pang *et al.* proposed Libra R-CNN network, it integrates IoU-balanced sampling, balanced feature pyramid, and balanced $\mathcal{L}_1$ loss, respectively for reducing the imbalance at sample, feature, and objective level, and reached an AP of 43.0 on the MS COCO dataset [20]. Li *et al.* constructed a parallel multibranch architecture in which each branch shares the same transformation parameters but with different receptive fields, and reached a mAP of 48.4% on the MS COCO dataset [21]. Zhu *et al.* proposed calibrating point-guided misalignment (CPM) R-CNN which contains three efficient modules to optimize anchor-based point-guided method. The CPM R-CNN could substantially improve detection mAP by 3.3% and 1.5% respectively compared with Faster R-CNN and Grid R-CNN [22]. The above detection algorithms based on candidate regions have a relatively slow detection speed because the model size is generally large in the process of accuracy improvement. In the research of object detection algorithms based on regression, because the generation stage of the candidate region is omitted, only one end-to-end processing cycle of the input image is needed to obtain the position and category of the object at the same time, so the detection speed is generally faster than that of the two-stage object detector. Redmon *et al.* proposed the first one-stage network, YOLOv1, the detection speed of which reached 45 FPS [3]. Liu *et al.*, based on YOLOv1, proposed the SSD, which improved the detection accuracy to 76.8% on the VOC2007 dataset and increased the detection speed to 59 FPS [9]. Redmon *et al.* proposed YOLOv2, and the accuracy rate on the VOC2007 dataset was 78.6% [4]. Redmon and Farhadi next proposed YOLOv3, and the accuracy rate on the MS COCO dataset increased to 33.0% [5]. Lin *et al.* proposed RetinaNet, and the accuracy rate on the MS COCO dataset increased to 40.8% [23]. Law and Deng proposed a CornerNet detection, and the accuracy rate on the MS COCO dataset increased to 42.1% [24]. Duan *et al.* proposed a CenterNet, which achieved an accuracy of 47.0% on MS COCO dataset [25]. Tan *et al.* presented EfficientDet, which achieved a test accuracy of 51.0% on the MS COCO dataset with a lower model scale [8]. Bochkovskiy *et al.* proposed YOLOv4, which combines a series of techniques with both detection accuracy and detection speed, and obtained 43.5% AP for the MS COCO dataset at a real-time speed of $\sim$ 65 FPS on a GPU of the Tesla V100 [7].

B. BOUNDING BOX REGRESSION

$\mathcal{L}_n$-norm losses (e.g., $\mathcal{L}_1$ and $\mathcal{L}_2$) are the most commonly used losses in deep learning [5], [9], and [13]. Yu *et al.* proposed IoU loss in the Unitbox network. The IoU loss took into account the overlapping area of bounding boxes of ground truth and prediction, and realized accurate and efficient localization through the Unitbox network. It overcame the shortcoming of $\mathcal{L}_n$-norm that the loss function is too simple and sensitive to different scales, and it was robust to objects of various shapes and scales [15]. However, IoU optimization in the case of non-overlapping bounding boxes is not feasible. Tychsen-Smith and Petersson derived a novel bounding box regression based on a set of IoU upper bounds that better matches the goal of IoU maximization while still providing good convergence properties [26]. To improve the accuracy of the bounding box regression, it is necessary to calculate not only the case in which the bounding boxes of ground truth and prediction overlap but also the case in which they do not overlap. On the basis of IoU, Rezatofighi *et al.* proposed GIoU to solve the problem of loss calculation when the bounding boxes do not overlap by adding penalty terms, and achieved better performance [17]. Although GIoU could alleviate the problem of gradient disappearance in the case of non-overlap, heavy reliance on IoU led to poor convergence and inaccurate detection. Qian *et al.* proposed a novel bounding box regression loss named IGIoU loss, which can optimize the GIoU directly and solve the aforementioned constant gradient problem involving GIoU loss[27]. Sun *et al.* proposed scale-balanced loss ($\mathcal{L}_{SB}$), which is asymmetric, position-sensitive, and scale-invariant. $\mathcal{L}_{SB}$ has improved average precisions at different IoU thresholds and scales [28]. Wu *et al.* proposed an IoU-aware one-stage object detector that predicts the IoU for each detected box, and the predicted IoU is multiplied by the classification score to compute the final detection confidence, which is more correlated with localization accuracy [29]. Zheng *et al.* proposed that DIoU could directly minimize the normalized distance between the centers of two bounding boxes of ground truth and prediction by changing the penalty term, and could converge faster than GIoU. However, any error is caused by many factors. Based on DIoU, Zheng *et al.* concurrently considered three important geometric measures (i.e., overlap area, central point distance and aspect ratio), and CIoU was proposed to achieve faster convergence and better performance than GIoU and DIoU [18].

III. METHODOLOGY

A. THE RELATIONSHIP BETWEEN BOUNDING BOXES

During the bounding box regression calculation, CIoU considered three geometric factors concurrently, including overlap area, central point distance, and aspect ratio between the two bounding boxes of ground truth and prediction, so as to obtain improvement in object detection performance. These three measures were defined in the loss function $\mathcal{L}_{CIoU}$, as shown in (1) [18].

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha v \tag{1}$$

where b and b^{gt} denote the central points of B and B^{gt}, $\rho(\cdot)$ is the Euclidean distance, and c is the diagonal length of the smallest enclosing box covering the two boxes. where α is a positive trade-off parameter defined as (2), and v measures the consistency of aspect ratio defined as (3).

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{2}$$

$$v = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2 \tag{3}$$

The w^{gt} and h^{gt} are the width and height of the bounding boxes of ground truth, while w and h are the width and height of the bounding boxes of prediction. It is because of aspect ratios of $\mathcal{L}_{CIoU}$ that the performance of state of the art was higher than $\mathcal{L}_{DIoU}$ [18].

In (3), when $w^{gt}/h^{gt} \neq w/h$, then $v > 0$, $\alpha v > 0$, penalty term αv has a positive role in the calculation of loss. However, if $w^{gt}/h^{gt} = w/h$ in (3), then $v = 0$ and $\alpha v = 0$, so $\mathcal{L}_{CIoU}$ degenerates into $\mathcal{L}_{DIoU}$, and the convergence rate will be reduced.

Hence, when $w^{gt}/h^{gt} = w/h$, it is necessary to analyze the size relationship between the loss values corresponding to the IoU, GIoU, DIoU, and CIoU algorithms. First, the box of ground truth and box of prediction were defined, as shown in Fig. 1, in which the blue rectangle is the bounding box of ground truth, B^{gt}, and the red rectangle is the bounding box of prediction, B^p. The centers of B^{gt} and B^p are C^{gt} and C^p, respectively.

In Fig. 1, there are three positional relationships between B^{gt} and B^p, including intersection, separation, and inclusion. When $w^{gt}/h^{gt} = w^p/h^p$, there are three positional relationships between B^{gt} and B^p, as shown in Fig. 2: intersection (a), separation (b), and inclusion (c). B^c is represented by the yellow dotted rectangle, which represents the minimum box that contains both B^{gt} and B^p in Fig. 2.

According to the method of calculating loss values described by the IoU, GIoU, DIoU, and CIoU algorithms, and combined with the three position relationships in Fig. 2, the magnitude relationship of each loss value was obtained, as shown in Tab. 1.

As can be seen from Tab. 1, the value range of $\mathcal{L}_{IoU}$ in the second row, when B^{gt} and B^p are separated, is shown in Fig. 2(b), $\mathcal{L}_{IoU} = 0$; when the corresponding dimensions of B^{gt} and B^p are equal and overlap, as shown in Fig. 2(c), $\mathcal{L}_{IoU} = 1$; in other cases, as shown in Figs. 2(a) and (c), $0 < \mathcal{L}_{IoU} < 1$. In the next line, the relationship between $\mathcal{L}_{GIoU}$ and $\mathcal{L}_{IoU}$ is presented, when $B^c = B^p$, as shown in Fig. 2(c), $\mathcal{L}_{GIoU}$ degenerates to $\mathcal{L}_{IoU}$; other situations are shown in Figs. 2(a) and (b), $\mathcal{L}_{GIoU} > \mathcal{L}_{IoU}$. The next line shows the relationship between $\mathcal{L}_{DIoU}$ and $\mathcal{L}_{IoU}$, in Fig. 2(c); if C^{gt} and C^p overlap, then $\mathcal{L}_{DIoU}$ degenerates to $\mathcal{L}_{IoU}$. For other cases, $\mathcal{L}_{DIoU} > \mathcal{L}_{IoU}$. The fifth line shows the relation between $\mathcal{L}_{CIoU}$ and $\mathcal{L}_{DIoU}$, as shown in Figs. 2(a)–(c), and $\mathcal{L}_{CIoU}$ degenerates into $\mathcal{L}_{DIoU}$ at three position relationships, which indicates that the $\mathcal{L}_{CIoU}$ algorithm still has shortcomings.

Based on the CIoU algorithm, an improved CIoU (ICIoU) algorithm is proposed in the present paper that takes the ratio of the corresponding width of the two bounding boxes for both ground truth and prediction as the factor of geometric

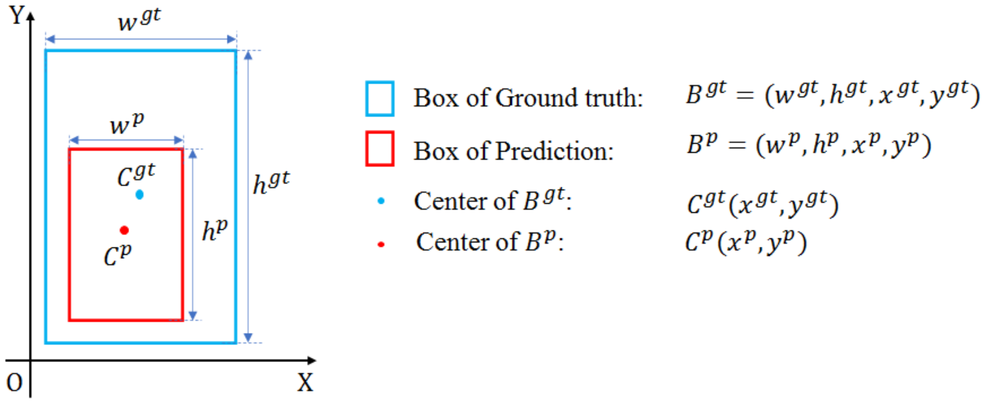
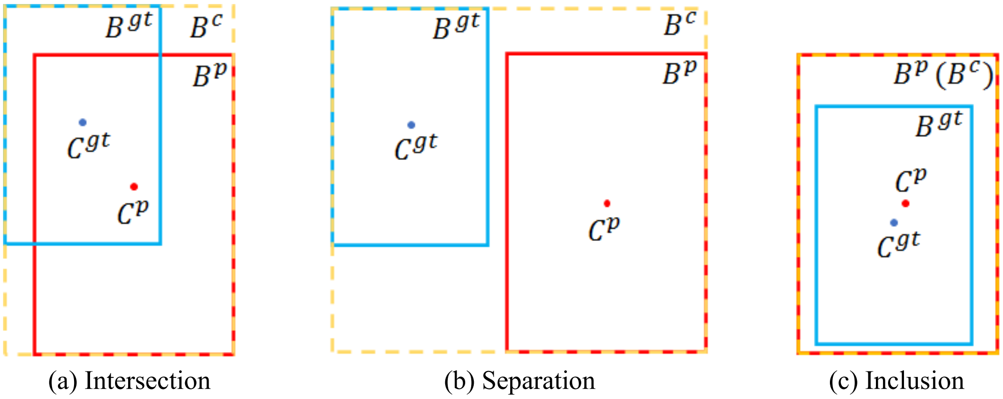

Y
w^{gt}
w^p
C^{gt}
C^p
h^p
h^{gt}
O
X
Box of Ground truth: $B^{gt} = (w^{gt}, h^{gt}, x^{gt}, y^{gt})$
Box of Prediction: $B^p = (w^p, h^p, x^p, y^p)$
Center of B^{gt}: $C^{gt}(x^{gt}, y^{gt})$
Center of B^p: $C^p(x^p, y^p)$

FIGURE 1. Bounding boxes of ground truth and prediction.



(a) Intersection (b) Separation (c) Inclusion

FIGURE 2. Position relationship between bounding boxes of ground truth and prediction.

measurement. Similarly, the ratio of the corresponding height of the two bounding boxes for both ground truth and prediction was also used as a factor in the geometric measure. This algorithm effectively avoids the situation that the CIoU algorithm degenerates into the DIoU algorithm when the aspect ratio of two bounding boxes for both ground truth and prediction is equal, thus improving localization accuracy of the improved algorithm.

The last line in Tab. 1 shows the relationship between the loss values $\mathcal{L}_{ICIoU}$ and $\mathcal{L}_{CIoU}$ calculated by the proposed algorithm. Among the three position relationships shown in Fig. 2, $\mathcal{L}_{ICIoU}$ can promote convergence better than $\mathcal{L}_{CIoU}$. The improved algorithm, ICIoU, is described in detail below.

B. THE PROPOSED METHOD

To change the penalty function degradation problem generated by the CIoU algorithm when $w^{gt}/h^{gt} = w^p/h^p$, it is proposed that the penalty function be calculated according to the variance of the corresponding side ratios of B^{gt} and B^p on the basis of the CIoU algorithm. This method improved the comprehensiveness of the calculation of the loss function when considering the geometric factors of the aspect ratio, effectively avoids the degradation of the penalty function, and enhances the robustness of calculating the loss function of different box sizes. The loss function designed in the present work is $\mathcal{L}_{ICIoU}$, and its expression is given in (4). The definitions of the two parameters α and v^v in the penalty term αv^v are given in (5) and (6), respectively.

$$\mathcal{L}_{ICIoU} = 1 - IoU + \frac{\rho^2\left(b^p, b^{gt}\right)}{c^2} + \alpha v^v \tag{4}$$

$$\alpha = \frac{v^v}{(1 - IoU) + v^v} \tag{5}$$

$$v^v = \frac{8}{\pi^2}[(arctan\frac{w^{gt}}{w^p} - \frac{\pi}{4})^2 + (arctan\frac{h^{gt}}{h^p} - \frac{\pi}{4})^2] \tag{6}$$

In particular, Eq. (6) is an improved expression for calculating the parameter v^v. In (6), when the size of B^{gt} and B^p is exactly the same, that is, $w^{gt} = w^p$ and $h^{gt} = h^p$, then $v^v = 0$, and the value of $\mathcal{L}_{ICIoU}$ is only related to IoU and the center distance of the two boxes, which is the same as $\mathcal{L}_{DIoU}$; that is to say, $\mathcal{L}_{DIoU}$ is a special case of $\mathcal{L}_{ICIoU}$. When the size of B^{gt} is not equal to that of B^p, then $v^v \neq 0$ and $\mathcal{L}_{ICIoU} > \mathcal{L}_{DIoU}$. Moreover, the larger the size difference between B^{gt} and B^p, the larger the penalty term αv^v and the larger the loss $\mathcal{L}_{ICIoU}$, which are more conducive to the bounding box regression calculation. Therefore, v^v can more

TABLE 1. **Relationship between loss values at three position relationships.**

Loss / Relationship	Intersection	Separation	Inclusion
$\mathcal{L}_{IoU}$	$0 < \mathcal{L}_{IoU} < 1$	$\mathcal{L}_{IoU} = 0$	$0 < \mathcal{L}_{IoU} \leq 1$
$\mathcal{L}_{GIoU}$	$\mathcal{L}_{GIoU} > \mathcal{L}_{IoU}$	$\mathcal{L}_{GIoU} > \mathcal{L}_{IoU}$	$\mathcal{L}_{GIoU} = \mathcal{L}_{IoU}$
$\mathcal{L}_{DIoU}$	$\mathcal{L}_{DIoU} > \mathcal{L}_{IoU}$	$\mathcal{L}_{DIoU} > \mathcal{L}_{IoU}$	$\mathcal{L}_{DIoU} \geq \mathcal{L}_{IoU}$
$\mathcal{L}_{CIoU}$	$\mathcal{L}_{CIoU} = \mathcal{L}_{DIoU}$	$\mathcal{L}_{CIoU} = \mathcal{L}_{DIoU}$	$\mathcal{L}_{CIoU} = \mathcal{L}_{DIoU}$
$\mathcal{L}_{ICIoU}$	$\mathcal{L}_{ICIoU} > \mathcal{L}_{CIoU}$	$\mathcal{L}_{ICIoU} > \mathcal{L}_{CIoU}$	$\mathcal{L}_{ICIoU} > \mathcal{L}_{CIoU}$

comprehensively reflect the positive effect of considering the change of box size on the loss of $\mathcal{L}_{ICIoU}$. Therefore, ICIoU not only embodies the advantages of CIoU but also improves the degradation problem of CIoU when the aspect ratio of the bounding box of ground truth is equal to the aspect ratio of the bounding box of prediction.

In addition, the optimization of loss $\mathcal{L}_{ICIoU}$ is same with that of loss $\mathcal{L}_{CIoU}$, except that the gradient of v^v w.r.t. w^p and h^p should be specified same as CIoU, as shown in (7) and (8), respectively.

$$\frac{\partial v^v}{\partial w^p} = \frac{16}{\pi^2}\left(arctan\frac{w^{gt}}{w^p} - \frac{\pi}{4}\right) \times \frac{w^{gt}}{(w^p)^2 + (w^{gt})^2} \quad (7)$$

$$\frac{\partial v^v}{\partial h^p} = \frac{16}{\pi^2}(arctan\frac{h^{gt}}{h^p} - \frac{\pi}{4}) \times \frac{h^{gt}}{(h^p)^2 + (h^{gt})^2} \quad (8)$$

The dominator $(w^p)^2 + (w^{gt})^2$ is usually a small value for the cases w^p and w^{gt} ranging in [0,1], which is likely to yield gradient explosion. And thus, in our implementation, the step size $1/((w^p)^2 + (w^{gt})^2)$ is replaced by 1 and the gradient direction is still consistent with (7). For the same reason, the step size $1/((h^p)^2 + (h^{gt})^2)$ is replaced by 1 and the gradient direction is still consistent with (8).

Two algorithms were designed to calculate the loss $\mathcal{L}_{ICIoU}$. Algorithm 1 is expressed as follows:

Algorithm 1 Calculating ICIoU Loss

```
Input: Bounding box of Ground truth $B^{gt} = (w^{gt}, h^{gt}, x^{gt}, y^{gt})$
Input: Bounding box of Prediction $B^p = (w^p, h^p, x^p, y^p)$
Output: $\mathcal{L}_{ICIoU}$
If $(B^{gt} \neq 0) \cup (B^p \neq 0)$ do
    If $\frac{w^{gt}}{h^{gt}} = \frac{w^p}{h^p}$ then
        $v^v = \frac{8}{\pi^2}[(arctan\frac{w^{gt}}{w^p} - \frac{\pi}{4})^2 + (arctan\frac{h^{gt}}{h^p} - \frac{\pi}{4})^2]$
        $\alpha = \frac{v^v}{(1-IoU)+v^v}$
        $\mathcal{L}_{ICIoU} = 1 - IoU + \frac{\rho^2(b^p,b^{gt})}{c^2} + \alpha v^v$
    else
        $\mathcal{L}_{ICIoU} = L_{CIoU} = 1 - IoU + \frac{\rho^2(b^p,b^{gt})}{c^2} + \alpha v$
else
    $\mathcal{L}_{ICIoU} = 0$
```

In Algorithm 1, if B^{gt} of the labeled object exists and B^p is not zero, when $w^{gt}/h^{gt} = w^p/h^p$, $\mathcal{L}_{ICIoU}$ are used to calculate the loss, and if $w^{gt}/h^{gt} \neq w^p/h^p$, they are used to calculate the loss. Algorithm 2 is expressed as follows:

Algorithm 2 Calculating ICIoU Loss

```
Input: Bounding Box of Ground Truth $B^{gt} = (w^{gt}, h^{gt}, x^{gt}, y^{gt})$
Input: Bounding Box of Prediction $B^p = (w^p, h^p, x^p, y^p)$
Output: $\mathcal{L}_{ICIoU}$
If $(B^{gt} \neq 0) \cup (B^p \neq 0)$ do
    $v^v = \frac{8}{\pi^2}[(arctan\frac{w^{gt}}{w^p} - \frac{\pi}{4})^2 + (arctan\frac{h^{gt}}{h^p} - \frac{\pi}{4})^2]$
    $\alpha = \frac{v^v}{(1-IoU)+v^v}$
    $\mathcal{L}_{ICIoU} = 1 - IoU + \frac{\rho^2(b^p,b^{gt})}{c^2} + \alpha v^v$
Else
    $\mathcal{L}_{ICIoU} = 0$
```

In Algorithm 2, if neither B^{gt} nor B^p is zero, then $\mathcal{L}_{ICIoU}$ is used to calculate all losses.

IV. EXPERIMENT AND RESULTS

All detection baselines were trained and all results on three standard object detection benchmarks reported, i.e., not only on the Udacity dataset but also on two popular benchmarks, namely the PASCAL VOC and MS COCO datasets [30], [31], [32]. The proposed ICIoU algorithm was evaluated by incorporating them into the state-of-the-art object detection algorithms, including the one-stage detection algorithm YOLO v4. The details of the algorithms' training protocol and evaluation are provided below. *All the source codes and our trained models will be made publicly available.*

A. EXPERIMENTAL ENVIRONMENT

The experimental environment in this paper was configured as follows: Intel (R) Core (TM) i9-10850 K CPU (3.60 GHz, 4 cores), 64 GB memory, Windows 10 Pro 64-bit operating system. NVIDIA GeForce GTX 3080 graphics card, 10 GB of video memory, Cuda V11.1, OpenCV version: 4.5.1.

B. DATASETS

1) UDACITY

The Udacity self-driving dataset contains two subsets of 9,423 and 15,000 images from a continuous video [30]. The videos were shot by a point gray research camera running at a full resolution of 1920 × 1200 pixels at 2 Hz when driving in Mountain View, California, and neighboring cities during daylight. The dataset as used includes 9,423 images labeled with 6,500 two dimensional labels for Car, Truck, and Pedestrian. In fact, there are 9,218 images that indicate

TABLE 2. Quantitative comparison of YOLOv4 trained using $\mathcal{L}_{IoU}$ (baseline), $\mathcal{L}_{GIoU}$, $\mathcal{L}_{DIoU}$, $\mathcal{L}_{CIoU}$ and $\mathcal{L}_{ICIoU}$. The results are reported on the test set of Udacity.

Loss/Evaluation	AP50	AP55	AP60	AP65	AP70	**AP75**	AP80	AP85	AP90	AP95	**AP**
$\mathcal{L}_{IoU}$	83.4	80.09	76.19	69.86	61.72	49.92	39.27	24.33	8.48	0.42	49.37
$\mathcal{L}_{GIoU}$	84.11	81.26	77.36	71.62	63.7	51.37	39.13	25.81	10.13	0.55	**50.50**
Relative improv. %	0.85%	1.46%	1.54%	2.52%	3.21%	2.90%	-0.36%	6.08%	19.46%	30.95%	**2.30%**
$\mathcal{L}_{DIoU}$	83.96	80.79	76.93	71.65	63.22	51.11	39.48	24.42	9.05	0.46	50.12
Relative improv. %	0.67%	0.87%	0.97%	2.56%	2.43%	2.38%	0.53%	0.37%	6.72%	9.52%	1.50%
$\mathcal{L}_{CIoU}$	83.85	80.96	76.78	70.88	61.99	51.48	38.97	24.93	9.93	0.55	50.03
Relative improv. %	0.54%	1.09%	0.77%	1.46%	0.44%	3.12%	-0.76%	2.47%	17.10%	30.95%	1.35%
Algorithm 1 ($\mathcal{L}_{CIoU}$ & $\mathcal{L}_{ICIoU}$)	83.49	81.36	77.19	71.34	62.94	50.49	39.38	25.51	9.99	0.48	50.22
Relative improv. %	0.11%	1.59%	1.31%	2.12%	1.98%	1.14%	0.28%	4.85%	17.81%	14.29%	1.72%
Algorithm 2 ($\mathcal{L}_{ICIoU}$)	83.41	80.92	77.03	71.96	62.16	**51.54**	39.55	25.5	10.37	0.73	**50.32**
Relative improv. %	0.01%	1.04%	1.10%	3.01%	0.71%	**3.25%**	0.71%	4.81%	22.29%	73.81%	**1.92%**

the object information of the three classes of Car, Truck, and Pedestrian, among which 205 images do not include the objects of the selected classes. The 9,218 images were divided into 6,452 images for the training set, 1,383 for the validation set, and 1,383 for the testing set.

2) PASCAL VOC

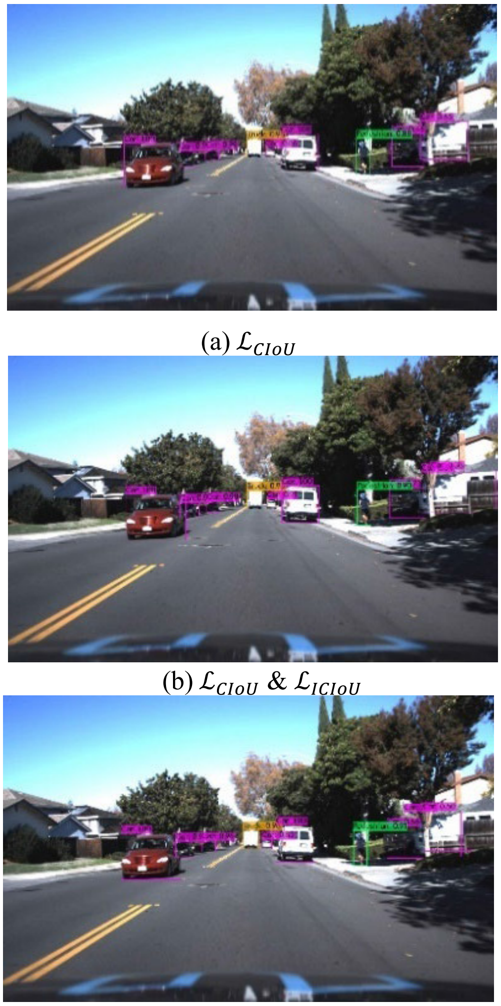
The PASCAL VOC 2007 benchmark dataset is one of the most widely used datasets for classification, object detection, and semantic segmentation. It consists of 9,963 images split for training and testing, in which objects from 20 pre-defined classes have been annotated with bounding boxes. The 9,963 images were divided into 5,977 images for the training set, 1,993 for the validation set, and 1,993 for the testing set.

3) MS COCO

Another popular benchmark for image captioning, recognition, detection, and segmentation is the more recent Microsoft Common Objects in Context. The MS COCO 2017 dataset as used consists of 122,218 images across training, validation, and testing sets with annotated object instances from 80 classes. The 122,218 images were divided into 73,330 for the training set, 24,444 for the validation set, and 24,444 for the testing set.

C. EVALUATION PROTOCOL

The same performance measures were adopted to report all results. This performance measures include the calculation of mAP over different class labels for a specific value of IoU threshold. The main performance measure used in this benchmark is shown by AP, AP = (AP50 + AP55 + $\cdots$ + AP95)/10, which is the average of mAP values across different values of 10 IoU thresholds, i.e., IoU = {0.5, 0.55, $\cdots$, 0.95}. Additionally, other values for AP are reported and were relatively improved by modifying the evaluation script to use, separately, GIoU, DIoU, CIoU, ICIoU, and CIoU+ICIoU instead of IoU. In addition, values for AP75 were also reported and relatively improved when IoU thresholds were equal to 0.75.


(a) $\mathcal{L}_{CIoU}$

(b) $\mathcal{L}_{CIoU}$ & $\mathcal{L}_{ICIoU}$

(c) $\mathcal{L}_{ICIoU}$

FIGURE 3. Detection sample image using YOLO v4 trained on Udacity.

TABLE 3. Quantitative comparison of YOLOv4 trained using $\mathcal{L}_{IoU}$ (baseline), $\mathcal{L}_{GIoU}$, $\mathcal{L}_{DIoU}$, $\mathcal{L}_{CIoU}$ and $\mathcal{L}_{ICIoU}$. The results are reported on the test set of PASCAL VOC 2007.

Loss/Evaluation	AP50	AP55	AP60	AP65	AP70	**AP75**	AP80	AP85	AP90	AP95	**AP**
$\mathcal{L}_{IoU}$	81.22	79	76.1	72.19	65.6	55.18	41.75	23.2	6.81	0.39	50.14
$\mathcal{L}_{GIoU}$	81.46	79.1	76.35	71.81	65.24	55.79	42.26	25.14	6.67	0.33	50.42
Relative improv. %	0.30%	0.13%	0.33%	-0.53%	-0.55%	1.11%	1.22%	8.36%	-2.06%	-15.38%	0.56%
$\mathcal{L}_{DIoU}$	81.55	79.31	76.36	71.79	64.49	54.84	40.43	22.74	6.59	0.19	49.83
Relative improv. %	0.41%	0.39%	0.34%	-0.55%	-1.69%	-0.62%	-3.16%	-1.98%	-3.23%	-51.28%	-0.63%
$\mathcal{L}_{CIoU}$	81.85	79.96	77.14	72.25	65.46	56.36	42.6	24.77	7.78	0.56	50.87
Relative improv. %	0.78%	1.22%	1.37%	0.08%	-0.21%	2.14%	2.04%	6.77%	14.24%	43.59%	1.45%
Algorithm 1 ($\mathcal{L}_{CIoU}$ & $\mathcal{L}_{ICIoU}$)	81.54	79.58	76.95	72.22	64.98	56.53	43.62	25.44	8.21	0.51	50.96
Relative improv. %	0.39%	0.73%	1.12%	0.04%	-0.95%	2.45%	4.48%	9.66%	20.56%	30.77%	1.62%
Algorithm 2 ($\mathcal{L}_{ICIoU}$)	81.73	79.82	76.87	72.33	66.1	**57.08**	43.18	25.22	7.4	0.32	**51.01**
Relative improv. %	0.63%	1.04%	1.01%	0.19%	0.76%	**3.44%**	3.43%	8.71%	8.66%	-17.95%	**1.72%**

D. YOLOv4

YOLOv4 is one of the most popular neural network models at present. While ensuring speed, YOLOv4 greatly improves detection accuracy of a model compared with YOLOv3. YOLOv4 is mainly composed of three parts: BackBone, Neck, and Head. Compared with Darknet53 of Yolov3, the BackBone of YOLOv4 uses CSPDarknet53. In place of YOLOv3's feature pyramid networks (FPN) [33], YOLOv4's Neck used SPP and PANet [2], [34]. The Head of YOLOv4 contains the Head of YOLOv3.

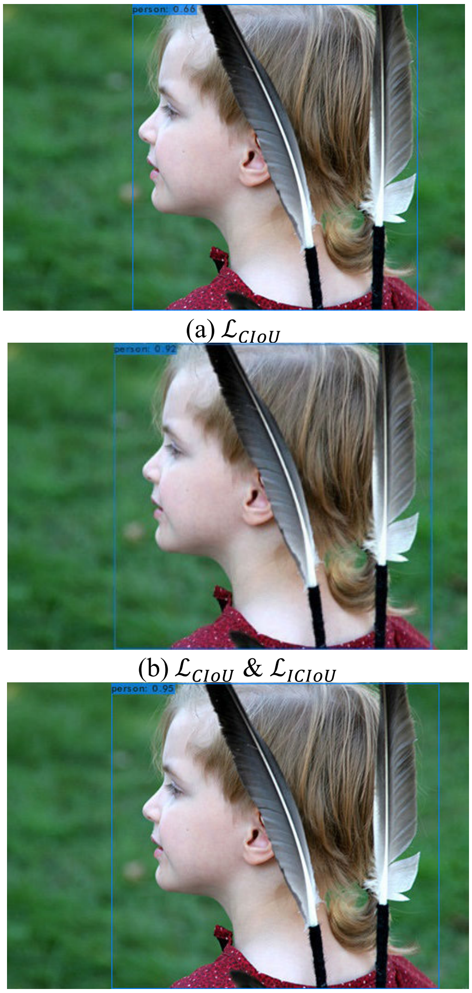
The Darknet[1] training protocol was followed exactly; the backbone network was Darknet416, and the maximum iteration was set to 30K.

E. RESULTS

1) YOLO V4 ON UDACITY

Udacity is one of the most popular self-driving datasets for object detection. YOLOv4 was separately trained on Udacity using Algorithms 1 and 2 with $\mathcal{L}_{ICIoU}$ and with $\mathcal{L}_{IoU}$, $\mathcal{L}_{GIoU}$, $\mathcal{L}_{DIoU}$, and $\mathcal{L}_{CIoU}$. For comparison, the performance for each loss is reported in Tab. 2.

From Tab. 2, $\mathcal{L}_{IoU}$ gains of 49.37 AP and 49.92 AP75 can be seen. Taking $\mathcal{L}_{IoU}$ as the evaluation benchmark and $\mathcal{L}_{GIoU}$ as the general version of $\mathcal{L}_{IoU}$, AP is increased to its highest level by 2.30% and 2.90%, respectively. However, $\mathcal{L}_{CIoU}$ does not perform well and improves the performance the least. Compared with $\mathcal{L}_{IoU}$, $\mathcal{L}_{CIoU}$ only improves the performance by 1.35% AP and 3.12% AP75, which is lower than $\mathcal{L}_{DIoU}$'s values of 1.50% AP and 2.38% AP75. The proposed Algorithm 1 combines $\mathcal{L}_{CIoU}$ and $\mathcal{L}_{ICIoU}$ to improve performance by 1.72% AP and 1.14% AP75, while Algorithm 2, $\mathcal{L}_{ICIoU}$, improves the performance by 1.92% AP and, the highest, 3.25% AP75. In particular, $\mathcal{L}_{ICIoU}$ obtains the highest value, 51.54% AP75, which is higher than the $\mathcal{L}_{CIoU}$ value of 51.48 AP75, indicating that the removal of the influence of the factor of equal aspect ratio between the bounding boxes of ground truth and prediction has the ability to improve the detection accuracy.



(a) $\mathcal{L}_{CIoU}$

(b) $\mathcal{L}_{CIoU}$ & $\mathcal{L}_{ICIoU}$

(c) $\mathcal{L}_{ICIoU}$

FIGURE 4. Detection sample image using YOLO v4 trained on PASCAL VOC.

[1] https://github.com/AlexeyAB/darknet

TABLE 4. Quantitative comparison of YOLOv4 trained using $\mathcal{L}_{IoU}$ (baseline), $\mathcal{L}_{GIoU}$, $\mathcal{L}_{DIoU}$, $\mathcal{L}_{CIoU}$ and $\mathcal{L}_{ICIoU}$. The results are reported on the test set of MS COCO 2017.

Loss/Evaluation	AP50	AP55	AP60	AP65	AP70	**AP75**	AP80	AP85	AP90	AP95	**AP**
$\mathcal{L}_{IoU}$	59.39	56.6	52.68	47.59	40.66	31.39	20.61	9.07	1.64	0.04	31.97
$\mathcal{L}_{GIoU}$	59.54	56.56	53.02	47.48	40.62	31.56	20.68	9.27	1.92	0.04	32.07
Relative improv. %	0.25%	-0.07%	0.65%	-0.23%	-0.10%	0.54%	0.34%	2.21%	17.07%	0.00%	0.32%
$\mathcal{L}_{DIoU}$	59.55	56.69	52.84	47.55	40.69	31.53	20.65	9.14	1.72	0.04	32.04
Relative improv. %	0.27%	0.16%	0.30%	-0.08%	0.07%	0.45%	0.19%	0.77%	4.88%	0.00%	0.23%
$\mathcal{L}_{CIoU}$	59.67	56.88	53.01	48.01	41.04	32.27	21.26	9.52	1.76	0.04	32.35
Relative improv. %	0.47%	0.49%	0.63%	0.88%	0.93%	2.80%	3.15%	4.96%	7.32%	0.00%	1.19%
Algorithm 1 ($\mathcal{L}_{CIoU}$ & $\mathcal{L}_{ICIoU}$)	59.6	56.81	52.91	47.75	40.66	31.93	20.72	9.53	1.77	0.04	32.17
Relative improv. %	0.35%	0.37%	0.44%	0.34%	0.00%	1.72%	0.53%	5.07%	7.93%	0.00%	0.64%
Algorithm 2 ($\mathcal{L}_{ICIoU}$)	59.64	56.79	52.93	47.96	41.11	**32.48**	21.29	9.57	1.88	0.04	**32.37**
Relative improv. %	0.42%	0.34%	0.47%	0.78%	1.11%	**3.47%**	3.30%	5.51%	14.63%	0.00%	**1.26%**

As shown in Fig. 3, a sample image of Udacity is taken as an example with which to compare the detection results of $\mathcal{L}_{CIoU}$ with the proposed Algorithms 1 and 2.

2) YOLO V4 ON PASCAL VOC

PASCAL VOC is one of the most popular datasets for object detection. YOLO v4 was trained on PASCAL VOC using Algorithms 1 and 2 including $\mathcal{L}_{ICIoU}$ for comparison with $\mathcal{L}_{IoU}$, $\mathcal{L}_{GIoU}$, $\mathcal{L}_{DIoU}$, and $\mathcal{L}_{CIoU}$. The performance for each loss is reported in Tab. 3.

Tab. 3 shows $\mathcal{L}_{IoU}$ gains of 50.14 AP and 55.18 AP75. Based on $\mathcal{L}_{IoU}$, $\mathcal{L}_{GIoU}$ increases by 0.56% AP and 1.11% AP75. However, $\mathcal{L}_{DIoU}$ does not perform well, but reduces the performance of $\mathcal{L}_{IoU}$ by 0.63% AP and 0.62% AP75. Based on $\mathcal{L}_{DIoU}$, $\mathcal{L}_{CIoU}$ obtains better performance by considering the scale information of the aspect ratio of the boundary boxes, i.e., 1.45% AP and 2.14% AP75. Algorithm 1 combines $\mathcal{L}_{CIoU}$ and $\mathcal{L}_{ICIoU}$ to improve the performance by 1.62% AP and 2.45% AP75. In Algorithm 2, $\mathcal{L}_{ICIoU}$ achieves the highest performance improvement on the basis of $\mathcal{L}_{CIoU}$, i.e., 1.72% AP and 3.44% AP75. In particular, $\mathcal{L}_{ICIoU}$ achieves the highest performance, i.e., 57.08 AP75, which is higher than that, i.e., 56.36 AP75, of $\mathcal{L}_{CIoU}$, indicating once again that the improved algorithm has the ability to improve detection accuracy.

A sample image of the PASCAL VOC dataset is shown in Fig. 4 to compare the detection results of $\mathcal{L}_{CIoU}$ with those of Algorithms 1 and 2.

3) YOLO V4 ON MS COCO

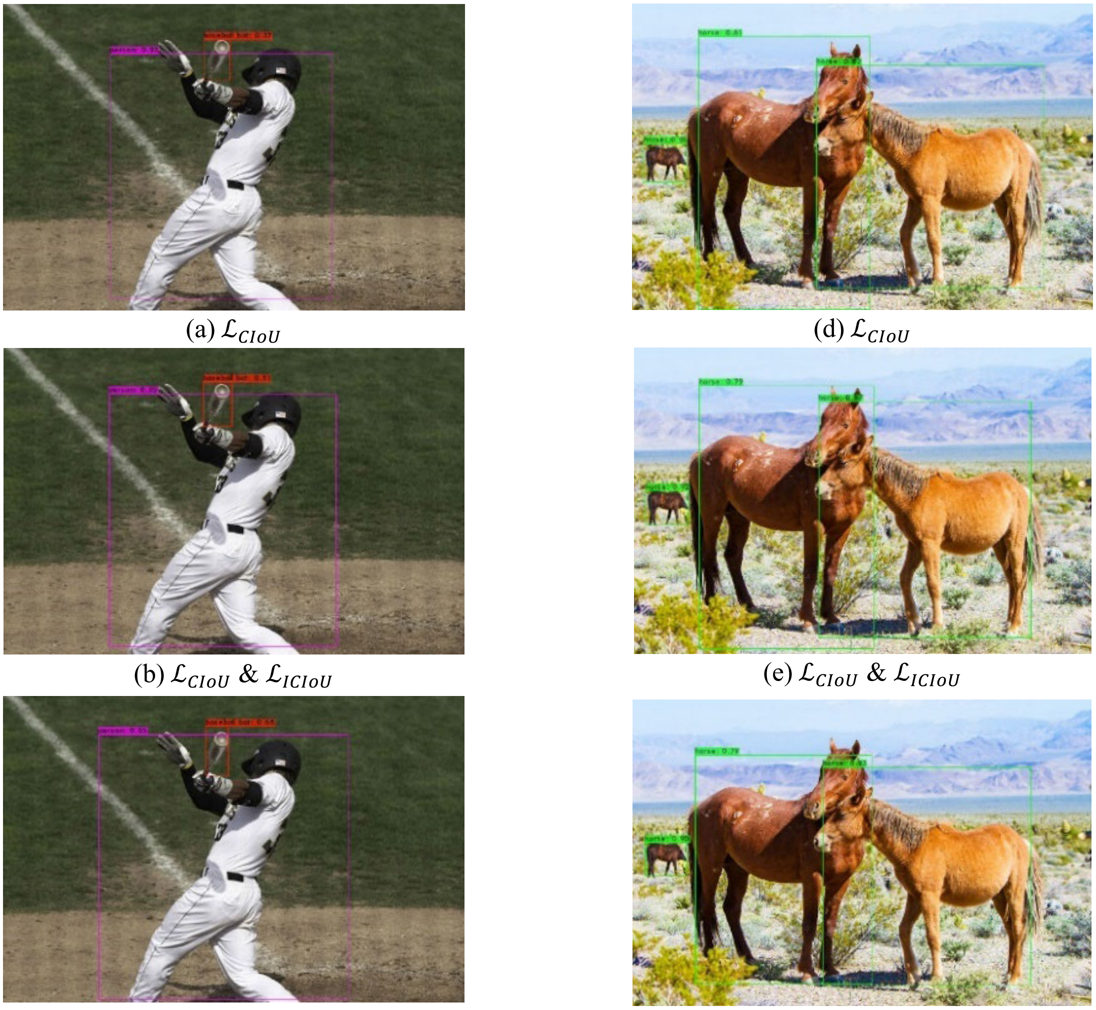
MS COCO 2017 is one of the most difficult and complex datasets for object detection. YOLO v4 was trained on MS COCO 2017 using Algorithms 1 and 2 including $\mathcal{L}_{ICIoU}$ for comparison with $\mathcal{L}_{IoU}$, $\mathcal{L}_{GIoU}$, $\mathcal{L}_{DIoU}$, and $\mathcal{L}_{CIoU}$. The performance for each loss is reported in Tab. 4.

Tab. 4 shows $\mathcal{L}_{IoU}$ gains of 31.97 AP and 31.39 AP75. Using $\mathcal{L}_{IoU}$ as the evaluation benchmark, $\mathcal{L}_{GIoU}$ improves the detection accuracy by 0.32% AP and 0.54% AP75. $\mathcal{L}_{DIoU}$ also improves the performance more than $\mathcal{L}_{IoU}$, with increases of 0.23% AP and 0.45% AP75, which is slightly lower those of than $\mathcal{L}_{GIoU}$. $\mathcal{L}_{CIoU}$ achieves better performance improvement, with 1.19% AP and 2.80% AP75. Algorithm 1 uses $\mathcal{L}_{CIoU}$ and $\mathcal{L}_{ICIoU}$ together to improve the performance by 0.64% AP and 1.72% AP75. Algorithm 2 $\mathcal{L}_{ICIoU}$ achieves the highest performance improvement, with 1.26% AP and 3.47% AP75, slightly higher than $\mathcal{L}_{CIoU}$.

Algorithm 2 exhibits better performance on the MS COCO dataset and has a growth rate of 1.26% AP, while it has a growth rate of 1.92% AP on Udacity in the Tab. 2 and of 1.72% AP on PASCAL VOC in the Tab. 3, which are the lowest growth rates compared to datasets with fewer samples. Because the number of objects in the MS COCO dataset is too large, even if the number of targets with the same aspect ratio between the bounding box of ground truth and that of prediction increases, with the sharp increase of all objects in the dataset as a whole, the proportion of objects meeting the degradation condition in all of the dataset objects still decreases.

Fig. 5 shows two sample images of the MS COCO dataset with which to compare the detection results of $\mathcal{L}_{CIoU}$ with the proposed Algorithms 1 and 2.

By comparing the results in Tab. 2, 3 and 4, we found that the $\mathcal{L}_{ICIoU}$ gets the highest scores on VOC 2007 and COCO Datasets, however, its scores are lower than the $\mathcal{L}_{GIoU}$ on Udacity Dataset. We think the reason for this result is related to the number of images and object classes in the data set. The number of object classes in Udacity dataset is only 15.0% of VOC 2007 dataset when the number of images is similar. The number of images in the Udacity dataset accounts for 7.7% of the COCO dataset, and the number of object classes in the Udacity dataset accounts for



(a) $\mathcal{L}_{CIoU}$

(d) $\mathcal{L}_{CIoU}$

(b) $\mathcal{L}_{CIoU}$ & $\mathcal{L}_{ICIoU}$

(e) $\mathcal{L}_{CIoU}$ & $\mathcal{L}_{ICIoU}$

(c) $\mathcal{L}_{ICIoU}$

(f) $\mathcal{L}_{ICIoU}$

FIGURE 5. Detection sample images using YOLO v4 trained on MS COCO.

only 3.8% of the COCO dataset. Therefore, we believe that there are fewer instances of Inclusion location relationships between the bounding boxes of ground truth and prediction. In addition, according to the analysis in Tab. 1, when there is Inclusion position relations between bounding boxes of ground truth and prediction, GIoU loss declines to IoU loss. Therefore, there is much less chance for GIoU loss to decay into IoU loss in the Udacity dataset than in the other two data sets, resulting in highest scores of $\mathcal{L}_{GIoU}$ in the Udacity dataset among the three datasets.

V. CONCLUSION

In the work described in this paper, the penalty term in $\mathcal{L}_{CIoU}$ of the bounding box regression loss function was improved. The new penalty term more comprehensively considers the aspect ratio relationship between the bounding box of ground truth and prediction, and includes the bounding box of prediction with the same aspect ratio as that of ground truth. The loss function, which is made up of the new penalty term, is called $\mathcal{L}_{ICIoU}$. Experiments on the Udacity, PASCAL VOC, and MS COCO datasets have proved the effectiveness of $\mathcal{L}_{ICIoU}$ in improving localization accuracy of the model by using the one-stage target detector YOLOV4. The method improved the comprehensiveness of the positioning judgment of the bounding box of prediction, strengthened the effect of the penalty function, and improved localization accuracy of the model. According to the theoretical analysis and calculation results of $\mathcal{L}_{ICIoU}$ function, our algorithm is advanced to a certain extent. However, we believe that the main limitation of the research in this paper is that it is only verified in the YOLOv4 network at present. In order to better demonstrate the effectiveness of the proposed method, it needs to be verified in other neural networks in the future. At the same time, the loss function of better performance is also one of the contents of our future research.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[4] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

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

[6] Y. Song, X. Li, and L. Gao, "Improved non-maximum suppression for detecting overlapping objects," in *Proc. 12th Int. Conf. Mach. Vis. (ICMV)*, Jan. 2020, doi: 10.1117/12.2556361.

[7] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*. [Online]. Available: https://arxiv.org/abs/2004.10934

[8] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 21–37.

[10] X. Zhang, F. Wan, C. Liu, X. Ji, and Q. Ye, "Learning to match anchors for visual object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jan. 12, 2021, doi: 10.1109/TPAMI.2021.3050494.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[12] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards realtime object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[14] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[15] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[16] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: An advanced object detection network," in *Proc. ACM Multimedia Conf. (MM)*, Oct. 2016, pp. 516–520.

[17] 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.

[18] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," 2019, *arXiv:1911.08287*. [Online]. Available: https://arxiv.org/abs/1911.08287

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

[20] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 821–830.

[21] Y. Li, Y. Chen, N. Wang, and Z. Zhang, "Scale-aware trident networks for object detection," 2019, *arXiv:1901.01892*. [Online]. Available: https://arxiv.org/abs/1901.01892

[22] B. Zhu, Q. Song, L. Yang, Z. Wang, C. Liu, and M. Hu, "CPM R-CNN: Calibrating point-guided misalignment in object detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3248–3257.

[23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

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

[25] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6569–6578.

[26] L. Tychsen-Smith and L. Petersson, "Improving object localization with fitness NMS and bounded IoU loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6877–6885.

[27] X. Qian, S. Lin, G. Cheng, X. Yao, H. Ren, and W. Wang, "Object detection in remote sensing images based on improved bounding box regression and multi-level features fusion," *Remote Sens.*, vol. 12, no. 1, p. 143, Jan. 2020.

[28] D. Sun, Y. Yang, M. Li, J. Yang, B. Meng, R. Bai, L. Li, and J. Ren, "A scale balanced loss for bounding box regression," *IEEE Access*, vol. 8, pp. 108438–108448, 2020.

[29] S. Wu, X. Li, and X. Wang, "IoU-aware single-stage object detector for accurate localization," *Image Vis. Comput.*, vol. 97, May 2020, Art. no. 103911.

[30] *An Open Source Self-Driving Car*, Udacity, Emeryville, CA, USA, 2017.

[31] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 740–755.

[33] T.-Y. Lin, P. Dollar, 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. 2117–2125.

[34] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8759–8768.

XUFEI WANG received the B.S. degree in mechanical engineering from Shaanxi University of Technology, China, in 1999, and the M.S. degree in mechanical engineering from Xinjiang University, China, in 2007. He is currently pursuing the Ph.D. degree in computer engineering with Pai Chai University, South Korea. Since 2011, he has been an Associate Professor with Shaanxi University of Technology. His research interests include self-driving and machine learning.

JEONGYOUNG SONG (Member, IEEE) received the B.S. degree in computer engineering from Hannam University, South Korea, in 1984, and the M.S. and Ph.D. degrees in electrical information and system from Waseda University, Japan, in 1992 and 1995, respectively. From 1995 to 1997, he was a Researcher in computer science with Cheongun University, South Korea. Since 1997, he has been a Professor with the Computer Engineering Department, Pai Chai University, South Korea. From 2011 to 2012, he was an Invited Scholarship Professor with the Department of Electrical Engineering, Idaho State University, USA. His research interests include pattern processing (image, speech, character) and machine learning.