Pattern Recognition 137 (2023) 109256

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Control Distance IoU and Control Distance IoU Loss for Better Bounding Box Regression

Chen Dong[a], Miao Duoqian[a,1,*]

Tongji University, No.4800, Cao'an Highway, Jiading District, Shanghai China

ARTICLE INFO

Article history:
Received 16 May 2022
Revised 16 November 2022
Accepted 13 December 2022
Available online 21 December 2022

Keywords:
Computer vision
Object detection
IoU
Loss function

ABSTRACT

Numerous improvements in feedback mechanisms have contributed to the great progress in object detection. In this paper, we first present an evaluation-feedback module, which consists of an evaluation system and feedback mechanism. Then we analyze and summarize traditional evaluation-feedback modules. We focus on both the evaluation system and the feedback mechanism, and propose **C**ontrol **D**istance IoU and **C**ontrol **D**istance IoU loss function (CDIoU and CDIoU loss) without increasing parameters in models, which make significant enhancements on several classical and emerging models. Finally, we propose **A**utomatic **G**round **T**ruth **C**lustering (AGTC) and **F**loating **L**earning **R**ate **D**ecay (FLRD) for faster regression in object detection. Experiments show that a coordinated evaluation-feedback module can effectively improve model performance. Both CNN and transformer-based detectors with CDIoU + CDIoU loss, AGTC, and FLRD achieve excellent performances. There are a maximum AP improvement of 2.9%, an average AP of 1.1% improvement on MS COCO, a maximum AP improvement of 8.2%, and an average AP improvement of 3.7% on Visdrone dataset.

1. Introduction

Tremendous works have been made for more accurate and more efficient object detection in recent years. Data augmentation [1], deeper layers of neural networks [2], more complex structured FPN modules, and even more number of iterations make the models for object detection the state-of-the-art. Undoubtedly these CNN detectors [3] have achieved remarkable success, however, at the same time these models have huge parameters and unsatisfactory FLOPs, such as Detectron2 Mask R-CNN X101-FPN (parameters: 107M, FLOPs: 277B) [4], ResNet-50 + NAS-FPN (1280@384) (parameters: 104M, FLOPs: 1043B) [5], AmoebaNet+ NAS-FPN +AA(1280) (parameters: 185M, FLOPs: 1317B) [6] and AmoebaNet+ NAS-FPN + AA(1536) (parameters: 209M, FLOPs: 3045B). Obviously, with the increase of model size, the performance of the model continues to improve. However, this performance improvement is limited. Meanwhile, detectors [7,8] under transformer framework suffer from the same problem.

This paper focuses efficient regression of object detection on automatic ground truth clustering, float learning rate decay, and the evaluation system & feedback mechanism (namely IoU modules and loss functions, combined called evaluation-feedback module) of region proposals without increasing the number of parameters or FLOPs [9,10].

The evaluation-feedback modules have 3 main roles: (1) Evaluating region proposals, using ground truth as a criterion. (2) Ranking a set of region proposals (with the same ground truth criterion). (3) Feeding the gap between region proposals (RP) and ground truths (GT) to the neural network, which is used to correct the next evaluation module. Considering evaluation-feedback module is fundamental, this module should be efficient and contain few parameters. A good evaluation-feedback module should meet the following 3 conditions:

- A measure overlapping area.
- The good ability to differentiate and a measure of the degree of difference (sometimes understood as centroid distance and aspect ratio).
- The IoUs calculation can be correlated with loss functions calculation.

Numerous previous studies have tended to focus on the study of feedback mechanism at the expense of evaluation system. In this paper, the Control Distance IoU(CDIoU) and the Control Distance IoU loss function (CDIoU loss) are proposed and given the same importance. CDIoU has good continuity and derivability, and simplifies the calculation by measuring the distance between RP and GT in a unified way, optimizing the calculation of DIoU and

* Corresponding author.
E-mail addresses: 1910691@tongji.edu.cn (C. Dong), dqmiao@tongji.edu.cn (M. Duoqian).
[1] Deputy director of Key Laboratory of embedded system and service computing, Ministry of Education

https://doi.org/10.1016/j.patcog.2022.109256
0031-3203/

CIoU [11] for centroid distance and aspect ratio, and completing the evaluation quickly. The CDIoU loss function can be correlated with CDIoU calculation, which enables the feedback mechanism to characterize more accurately and feed back the difference between RP and GT, thus making the objective function of the deep learning network converge faster and improving the overall efficiency. The CDIoU and CDIoU loss functions are highly adaptive and show significant improvements on several different models, compared to traditional IoU modules and loss functions.

To prevent the learning rate from being too large and swinging back and forth when converging to the global optimum, it is important to let the learning rate keep decreasing with the number of training rounds and converge to a gradient-decreasing learning step. The traditional object detectors are executed with a fixed learning rate or a learning rate decay strategy. However, the existing strategies neglect to provide feedback on the results of the loss function in real-time during the regression process. In other words, the learning rate should not be decayed continuously during the training process, but should be increased and decreased during the training process according to the result of the loss function. We propose float learning rate decay (FLRD) to build a feedback mechanism for the learning rate and loss function.

By generating anchors after analysis of ground truth in original dataset, the results of object detectors can be improved. Faster RCNN [12] and SSD [13] propose the foundational steps of anchor generation, and then YOLOv3 generates anchor boxes by K-means clustering. But this method requires repeated experiments to determine the centers and number of clusters. In order to save training time and faster regression, we propose automatic ground truth clustering (AGTC), which can solve the information and number of cluster centers (namely the scale and number of anchor boxes) in a specific dataset at once.

We aim to improve the performance of 2D object detection without significantly increasing the running cost. Saving running cost (model size and running time) is not the priority of this article. At the same time, under the premise of slightly improving the results of general object detection, the performance of small object detection is significantly improved.

The main contributions of this work can be summarized as:

- CDIoU is proposed as a new evaluation system and CDIoU loss as a new feedback mechanism for more accurate regression.
- AGTC and FLRD are proposed for faster regression.
- Improving the results, while the number of parameters and running time are not increased.
- With wide applicability, make significant improvements on several models.

2. Related work

The first culmination of deep learning for object detection was the proposal of R-CNN [14], Fast R-CNN and Faster R-CNN [15] models, which laid down the basic framework and data processing for deep learning applied to object detection. YOLO [16] provides a more straightforward way by directly regressing the location of the bounding box and the class to which the bounding exploitation belongs in the output layer, thus transforming the object detection problem into a regression problem. After this, YOLOv2 [17], YOLOv3, YOLOv4 were proposed, which made the deep learning network not only improve in accuracy but also in computing speed. R-CNN series and YOLO series are the classical representatives [18] of two-stage model [19,20] and one-stage model [21] in object detection.

Neural network backbone and convolution kernel

The backbone networks of deep learning are also evolving. LeNet (1998), AlexNet (2012), VGGNet (2014), GoogLeNet (2014), ResNet (2015), and MobileNet (2017) are preserved in the path of deep learning development [22]. EfficientNet (2019) [23] proposes a more generalized idea on the optimization of current classification networks, arguing that the three common ways of enhancing network metrics, namely widening the network, deepening the network and increasing the resolution, should not be independent of each other.

Along with the backbone [24], the convolution kernel [25] is also evolving and changing. Deformable conv [26] adds an offset variable to the position of each sampled point in the convolution kernel, enabling random sampling around the current position without being restricted to the previous regular grid points. Dilated conv [27] can effectively focus on the semantic information of the local pixel blocks, instead of letting each pixel rub together with the surrounding blocks, which affects the detail of segmentation.

Evaluation-feedback module

Based on IoU, GIoU [28] focuses not only on overlapping regions but also on other non-overlapping regions, which can better reflect the overlap of RP and GT. DIoU [11] takes the distance between the object and anchor, overlap rate, and scale into consideration, which makes the object box regression more stable and does not have problems such as scattering during training like IoU and GIoU.

GIoU loss [29] still has the problems of slow convergence and inaccurate regression. It is found that GIoU first tries to overlap the object box by increasing the size of the detection box, and then uses the IoU loss term to maximize the overlap area with the object box [30].

DIoU loss and CIoU loss [11] greatly enriched the connotation of IoU calculation results, adding the measurement of difference, including “centroid distance” and “aspect ratio” separately. DIoU loss cannot distinguish which RPs is more similar to GT when the center points of RPs are at the same position. We can know that the calculation process of CIoU loss is more time-consuming, which will eventually drag down the overall training and test time [31].

3. Analysis of traditional IoUs and loss functions

In this section, we systematically explain the disadvantages of traditional IoUs and loss functions in the formal article. Fig. 1, 2, 3, and 4 gradually illustrate the disadvantages of the traditional evaluation systems.

In object detection, the function of IoUs is to evaluate the similarity between RP and GT. The evaluation between RP and GT is given through the IoU method, which plays a fundamental role in the selection of positive and negative samples. In the evaluation-feedback module, the most representative methods are IoU, GIoU, DIoU loss and CIoU loss, which play a fundamental role in the great progress of object detection, but there still is much room for optimization.

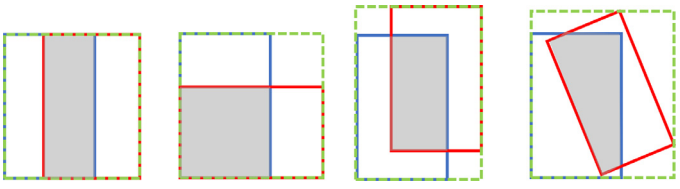


Fig. 1. Different relative position relations, the same IoU results. The red box is RP and the blue one is GT, while the green dashed box is the minimum bounding rectangle (MBR). MBR will be used in later chapters. The red box in the fourth image is rotated, which is to highlight the disadvantages of the traditional evaluation method. While all the detection frameworks used in the paper (e.g. Faster R-CNN, Cascade R-CNN, and YOLO) can only predict horizontal boxes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

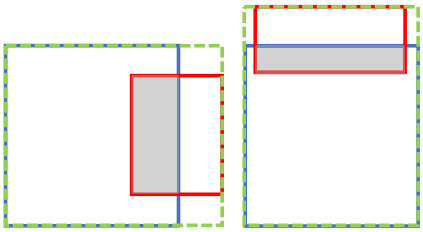


Fig. 2. Comparison of IoU GIoU CDIoU and Loss of GIoU & CDIoU. PR is outside GT. This figure emphasizes the "insensitivity" of IoU, GIoU to tiny differences. While CDIoU showed satisfactory sensitivity to such tiny differences.

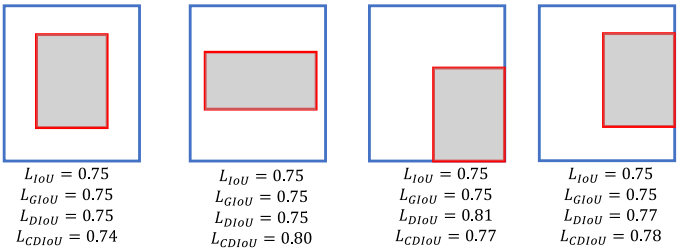


Fig. 3. Comparison of multiple IoU losses. PR is inside GT. This figure emphasizes the "insensitivity" of IoU loss, GIoU loss, CIoU loss and DIoU loss to tiny differences. While CDIoU loss showed satisfactory sensitivity to such tiny differences.

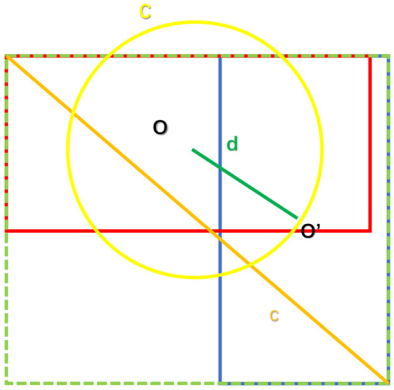


Fig. 4. DIoU loss: unuseful relative position relationship between RP and GT. in fact, as long as the center point of the region proposal is on arc C of circle O, the penalty terms of DIoU loss are consistent.

3.1. Analysis of traditional IoUs

IoU is a basic evaluation method, and as shown in Figure 1, the relative position relationship between RP and GT is obviously different. The human brain can clearly distinguish the differences, but the evaluation results of IoU are the same ($IoU = 0.33$).

Based on original IoU, many evaluation systems are derived, which enrich the evaluation dimensions of previous IoU from those different aspects. IoU only considers the calculation of overlapping area. Meanwhile, GIoU pays attention to overlapping area and non overlapping area, and strengthens the discussion of evaluation system. However, GIoU obviously ignored the "measurement of difference" between RP and GT.

The "measurement of difference" between RP and GT include the distance between the center points (centroid) and the ratio of length-width(aspect ratio). DIoU takes centroid into account in the calculation of the evaluation system, but omits aspect ratio. As shown in Figure 2, 3, DIoU can't recognize the difference between the high-thin region proposal and the short-fat region proposal, and gives them the same value. But in fact, the human brain can easily differentiate which one is better.

3.2. IoU:Smooth L1 loss and IoU loss

The method of smooth loss is proposed from Fast RCNN [12], which initially solves the problem of characterizing the boundary box loss. Assuming that x is the numerical difference between RP and GT, L_1 and L_2 loss are commonly defined as:

$$\mathcal{L}_1 = |x| \frac{d\mathcal{L}_2(x)}{x} = 2x, \tag{1}$$

$$\mathcal{L}_2 = x^2. \tag{2}$$

The corresponding derivative:

$$\frac{d\mathcal{L}_1(x)}{x} = \begin{cases} 1 & , \text{if } x \geq 0 \\ -1 & , \text{otherswise,} \end{cases} \tag{3}$$

$$\frac{d\mathcal{L}_2(x)}{x} = 2x. \tag{4}$$

From the derivative of loss function to x, we can know that the derivative of loss function $\mathcal{L}_1$ to x is constant. In the late training period, when x is very small, if the learning rate is constant, the loss function will fluctuate around the stable value, and it is difficult to converge to higher accuracy. When the derivative of loss function $\mathcal{L}_2$ to x is large, its derivative is also very large and unstable at the beginning of training. $smooth_{\mathcal{L}_1}(x)$ perfectly avoids the shortcomings of $\mathcal{L}_1$ and $\mathcal{L}_2$ loss.

$$\text{smooth}_{\mathcal{L}_1}(x) = \begin{cases} 0.5x^2 & , \text{if} |x| < 1 \\ |x| - 0.5 & , \text{otherswise,} \end{cases} \tag{5}$$

$$\frac{dsmooth_{\mathcal{L}_1(x)}}{x} = \begin{cases} x & , \text{if } |x| < 1 \\ \pm 1 & , \text{otherswise,} \end{cases} \tag{6}$$

However, in the actual object detection, the loss in box regression task is

$$\mathcal{L}_{loc}(t^u, v) = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{\mathcal{L}_1}\left(t_i^u - v_i\right), \tag{7}$$

Where $v = \left(v_x, v_y, v_w, v_h\right)$ represents the box coordinates of GT, and $t^u = \left(t_x^u, t_y^u, t_w^u, t_h^u\right)$ represents the predicted box coordinates, that is to calculate the loss of four points respectively, and then add them as the bounding box regression loss.

Remark 1. When the above losses are used to calculate the bounding box loss of object detection, the loss of four points is calculated independently, and then the final bounding box loss is obtained by adding. The assumption of this method is that the four points are independent of each other, and there is a certain correlation in fact. The calculation of smooth can not be unified with IoU, which leads to errors of the feedback mechanism and evaluation system. The actual indicator of evaluation is to use IoU, which is not equivalent. IoU loss cannot avoid this scenario,"different RPs, same feedback results" in Figure 2,3.

$$\mathcal{L}_{IoU} = -\ln(IoU), \tag{8}$$

$$\mathcal{L}_{IoU} = 1 - IoU. \tag{9}$$

3.3. GIoU and GIoU loss

On the premise of not increasing the calculation time, GIoU [28] initially optimized the calculation of IoU for overlapping area, and reduced the calculation error, but GIoU still did not take the measurement of the difference into account in the calculation results.

$$GIoU = IoU - \frac{|C \backslash (A \cup B)|}{|C|}, \tag{10}$$

$$\mathcal{L}_{GIoU} = 1 - GIoU. \tag{11}$$

GIoU loss still has the problems of slow convergence and inaccurate regression. It is found that GIoU first tries to overlap the object box (GT) by increasing size of the detection box (RP), and then uses IoU loss term to maximize the overlap area with the object box. At the same time, when the two boxes contain each other, GIoU loss will degenerate into IoU loss. At this time, the alignment of the bounding box becomes more difficult and the convergence is slow.

3.4. DIoU loss and CIoU loss

DIoU loss and CIoU loss [11] greatly enriched the connotation of IoU calculation results, adding the measurement of difference, including "centroid distance" and "aspect ratio" separately.

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2\left(\mathbf{b}, \mathbf{b}^{gt}\right)}{c^2} \tag{12}$$

First of all, DIoU loss cannot distinguish which region proposals are more similar to ground truth when the center points of region proposals are at the same position. Then when two boxes are completely coincident, $\mathcal{L}_{IoU} = \mathcal{L}_{GIoU} = \mathcal{L}_{DIoU} = 0$; when two boxes do not intersect, GIoU loss can't distinguish region proposals exactly and $\mathcal{L}_{GIoU} = \mathcal{L}_{DIoU} \to 2$.

As shown in the Figure 4, in fact, as long as the center point of the region proposal is on arc C of circle O, the penalty terms of DIoU loss are consistent. This is DIoU, which loses the accuracy of the evaluation system. The distance of RP and GT centers are the same, but the relative relationship between RP and GT can be different. Please take a closer look at the first and second subgraphs of Fig. 3 (O⟶ O',overlap), RP in the first is better then that in the 2nd, but DIoU does not know that, while CDIoU does.

The penalty term of CIoU loss is composed of a factor αv and DIoU loss penalty term, which takes that into account the aspect ratio of RP and GT.

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2\left(\mathbf{b}, \mathbf{b}^{gt}\right)}{c^2} + \alpha v \tag{13}$$

The penalty is

$$R_{CIoU} = \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha v, \tag{14}$$

$$\alpha = \frac{v}{(1 - IoU) + v}, \tag{15}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2. \tag{16}$$

Because the calculation of CIoU loss involves the inverse trigonometric function, and through comparative experiments, we can know that the calculation process of CIoU loss is more time-consuming, which will eventually drag down the overall training time. For detailed comparison tests, see "**Ablation studies**".

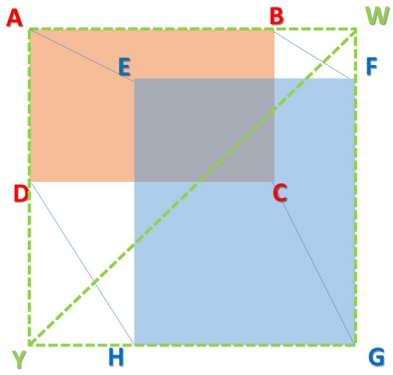


Fig. 5. Calculation of CDIoU. Minimum bounding rectangle (MBR) is the smallest rectangle that can contain RP and GT. The definition of minus operation between RP and GT is vector subtraction. "AE, BF,...", represent the length of the line segment.

4. CDIoU and CDIoU loss functions

Based on traditional IoUs and loss functions [32], CDIoU and CDIoU loss functions are proposed in this section. The improved loss functions [24,33] have to increase the computational intensity in order to improve the accuracy of the regression. Without increasing the operation time, the running efficiency and AP are significantly improved. The CDIoU loss function converges faster and reduces the complexity of the operation significantly.

Control Distance Intersection over Union (CDIoU) is a new evaluation method that directly examines the similarity of RP and GT, and it does not directly measure the distance between their centroids and the similarity of their shapes. For detailed information, see Figure 5.

$$\begin{aligned} \text{diou} &= \frac{\|RP - GT\|_2}{4\text{MBR}'\text{sdiagonal}} \\ &= \frac{AE + BF + CG + DH}{4WY} \end{aligned} \tag{17}$$

where RP and GT represent the vector (x1, y1, x2, y2) in the Eq. 17. The definition of minus operation between RP and GT is vector subtraction. "AE, BF,...", represent the length of the line segment. The diou(L2 loss) is different from DIoU, extracting the center distance between RP and GT and the shape of RP relative to GT at the same time.

$$CDIoU = IoU + \lambda(1 - \text{diou}) \tag{18}$$

Although the formula for CDIoU does not mention "centroid distance" and "aspect ratio", the final calculation results reflect a measure of the degree of difference between RP and GT. The diou(L2 loss) is different from DIoU, extracting the center distance between RP and GT and the shape of RP relative to GT at the same time. The higher the value of CDIoU, the lower the degree of difference; the higher the value of CDIoU, the higher the similarity.

$$\mathcal{L}_{CDIoU} = \mathcal{L}_{IoU_s} + \text{diou} \tag{19}$$

In order to cooperate with the calculation of CDIoU, this paper also proposes the CDIoU loss function. By observing this formula, we can intuitively feel that after backpropagation, the deep learning model tends to pull the four vertices of the region proposal toward the four vertices of the ground truth until they overlap. For detailed information, see Algorithm 1 and Fig. 6. In the subsequent experimental chapters of the article, we use IoU and IoU loss in the CDIoU and CDIoU loss.

5. Modules for faster regression

In this section, we propose two improved methods for model design. These methods can be applied to a variety of object detection models without increasing the running time of the model.

Algorithm 1 CDIoU and CDIoU loss function .

Input: RP for region proposal;GT for ground truth;
Output: CDIoU and CDIoU loss;
For RP and GT, find MBR;
compute $IoU = \frac{|RP \cap GT|}{|RP \cup GT|}$,diou $= \frac{\|RP-GT\|_2}{4\mathrm{MBR}'\mathrm{sdiagonal}}$;
compute $CDIoU = IoU + \lambda(1 - \mathrm{diou})$;
compute $\mathcal{L}_{CDIoU} = \mathcal{L}_{IoU_s} + \mathrm{diou}$, $\mathcal{L}_{IoU_s}$ could be $\mathcal{L}_{IoU} = -\ln(IoU)$, $\mathcal{L}_{IoU} = 1 - IoU$, $\mathcal{L}_{IoU} = 1 - IoU$ or $\mathcal{L}_{DIoU}$, $\mathcal{L}_{CIoU}$;

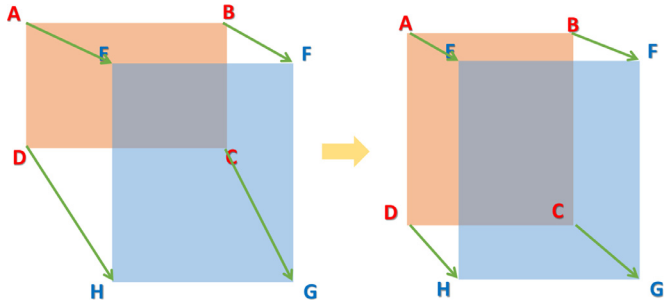


Fig. 6. Region proposal turning and variations by CDIoU loss.

These tips are particularly useful for some basic models in this paper.

5.1. Floating learning rate decay

The basic idea of learning rate decay is that the learning rate decays gradually as training proceeds. Common learning rate decay methods include piecewise decay, linear decay, exponential decay, natural exponential decay, polynomial decay and cosine decay. Among them, linear decay and exponential decay are the most commonly used decay methods.

The fixed learning rate decay method initially solves the problem of excessive constant learning rate in the late stage of deep learning training. However, during the gradual decay of the learning rate, the traditional methods cannot provide timely feedback on the impact of the decaying learning rate on the deep learning training process. Traditional methods simply mechanically perform tasks with progressively lower learning rates. In the training process of deep learning, too large learning rate will lead to the training process loss function divergence, while too small learning rate will lead to the training process convergence speed is too slow.

For the above reasons, we propose float learning rate decay (or FLRD for short) to check the loss every $\mathcal{K}$ iterations and increase the learning rate slightly, if the loss function does not decrease continuously. In this way, the learning rate will decrease and float appropriately at regular intervals to promote the decrease of the loss function.

$$lr_n = \begin{cases} 1.1lr_{n-1}, & Ls < 0 \\ \xi lr_{n-1}, & Ls > 0 \end{cases} \qquad (20)$$
$$Ls = \mathrm{loss}_i - \mathrm{loss}_{i-k}$$

where lr_{n-1} represents the learning rate of the previous decay cycle and lr_{n-1} represents the learning rate of the next decay cycle, Ls represents the difference between the loss of the first item and the loss of the last item in one decay cycle, and ξ represents decay rate. We take exponential decay as an example to introduce floating learning rate decay.

5.2. Automatic GT clustering analysis

It is well known that AP can be effectively improved by performing cluster analysis on GT in the original dataset. The conventional object detectors adjust anchor sizes and aspect ratios parameters based on the results of this cluster analysis. However, we do not know the number of clusters through the current approach. The main solution is to keep trying the number of clusters $\mathcal{N}$, and then judge by the final result AP. Obviously, this exhaustive method takes a lot of time.

In this section, automatic GT clustering analysis based on original dataset is proposed, using *K-means/PAM, Hierarchical Clustering, Spectral Clustering, DBSCAN* and *Mean-shift* methods respectively, where *DBSCAN* and *Mean-shift* methods are able to obtain the number of clusters autonomously.

In the MS COCO dataset, we can guide the generation of anchor by clustering the shapes (length and width) of GT. Since the set of GT shapes can be approximated as a dense dataset, we can take the DBSCAN method to cluster the GT set. The width and the length of GT are used as two dimensions to cluster the dataset of GT.

The above methods were evaluated using *SSE* (sum of the squared errors), *Silhouette Coefficient* and *Calinski-Harabaz*, and then two recommended schemes are obtained. These recommended schemes include the number of clusters and the central GT of each cluster. We obtained anchor information from the central GT before executing the complex deep learning network, so that the experiments in this paper are much more efficient.

The new aspect ratios are generated by automatic GT clustering method (or AGTC for short). And for consistency, anchor sizes of AGTC follow those of Faster RCNN and SSD (see in Table 1).

Table 1
Aspect ratios and anchor sizes in different detectors. AGTC means that automatic GT clustering generates the aspect ratio. * means that in at least 6 convolutional layers, multiple aspect ratio combinations are set up in SSD method. ** means that anchor sizes of AGTC follow those of Faster RCNN and SSD in ablation studies. Aspect ratios = width : length.

Method	Aspect ratios	Anchor sizes
Faster RCNN	1:1, 1:2, 2:1	3
SSD	1:1, 2:1, 3:1, 1:2, 1:3	*
AGTC	1:1, 1.5:1, 2:1, 1:2, 1:2.5, 1:3, 1:3.5, 1:4	**

6. Experiments

To ensure the rigidity and richness of the experiments, we did a lot of training and testing on representative models, such as Faster R-CNN, Cascade R-CNN [34], YOLOs and ATSS [35]. Also, we try not to use tricks and allow individual models to compare differences purely due to changes in IoUs or loss functions. Following ATSS works, the multi-scale training strategy is adopted for these experiments, i.e., randomly selecting a scale between 640 to 800 to resize the shorter side of images during training.

6.1. Working environment and preparation

The following experiments were conducted on MS coco 2017 dataset using two GeForce RTX 2080 Ti GPUs or two Tesla V100 PCIe 32GB GPUs. All models under Pytorch framework are standard models without using any tricks.And we double the total number of iterations to 180K and the learning rate reduction points to 120K and 160K.

Dataset. We perform experiments on COCO 2017 and Visdrone.

COCO 2017 contains 118k training images, 5k validation images and 20K test-dev images. The ablation study is performed using the validation set, and a system-level comparison is reported on test-dev. Each image is annotated with bounding boxes and panoptic segmentation. There are 7 instances per image on average, up to 63 instances in a single image in training set, ranging from small to large on the same images.

Table 2
Analysis(%) of different values of hyperparameter λ on the **MS COCO** `val` set.

λ	1.0	0.1	0.01	0.001	0.0001
Faster R-CNN	35.6	36.5	37.0	38.5	38.0
ATSS(R_50_FPN)	38.1	39.0	39.3	39.5	38.9

The Visdrone dataset consists of 400 video clips formed by 265,228 frames and 10,209 static images, captured by various drone-mounted cameras, covering a wide range of aspects including location, environment, objects (10 classes). These frames are manually annotated with more than 2.6 million bounding boxes or points of targets of frequent interests, such as pedestrians, cars, bicycles, and tricycles.

Training. Our method is trained with Adamw and SGD optimizers, changing Adamw to SGD until very final stage. We adopt our models with the learning rate (2^{-5}) for backbone. The backbone is the ImageNet-pretrained model with batchnorm layers fixed, and the Transformer parameters are initialized using the Xavier initialization scheme. The weight decay is set to be 10^{-4}.

6.2. CDIoU and CDIoU loss for object detection

In order to verify the effectiveness of CDIoU and CDIoU loss in object detection, experiments are designed and applied to numerous models in this paper. These models encompass existing classical models and emerging models, reflecting certain robustness and wide adaptability. We conduct several experiments to study the robustness of the hyperparameter λ in Table 2. Overall, the only hyperparameter $\lambda = 0.001$ is quite robust and the proposed CDIoU can be nearly regarded as hyperparameter-free[2].

CDIoU and CDIoU loss are universally adaptable, exhibiting differential performance gains on different models. As shown in Table 3, we can see that the more complex the backbone structure is, the less the CDIoU and CDIoU loss improvement is, while for the basic model, the CDIoU and CDIoU loss improvement is more obvious. By using CDIoU and CDIoU loss, the models in Table 3 improved AP by an average of 0.8%.

CDIoU & CDIoU loss are able to improve both of basic and new detectors' performances. Lots of experiments on representative SATA methods are token, including DETR [37], Swin Transformer and D2Det [38]. It's clear that the best AP (%) of DETR and D2Det are still not better than those of ATSS + CDIoU & loss, but the AP has been significantly improved in Table 4.

6.3. Ablation studies

Faster R-CNN is a classical model, while ATSS is a new model recently. In Table 5, the evaluation system of all models is IoU, the most basic original one. But in the loss function, we selects different calculation functions as feedback mechanism. IoU, GIoU, DIoU and CIoU are representative classic loss functions, which are implemented on many CNN-based and Transformer-based methods. *FPS* represents the number of images that can be processed per second. CDIoU loss does not increase the amount of computation and barely improves *FPS*.

As shown in Table 5, we can accurately see that CDIoU loss function can significantly improve the AP results by 0.2∼1.9 % compared to other loss functions, and this effect is more obvious in traditional and basic models.

From Fig. 7 and 8, it is clear that the loss function values of both Faster R-CNN and ATSS-R-50-FPN-1x models drop and reach convergence faster with CDIoU and CDIoU loss function, compared

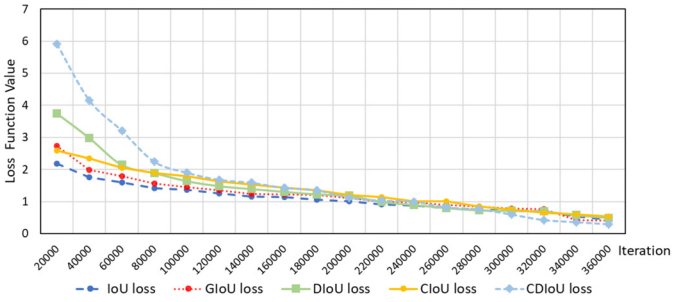


Fig. 7. Loss Function value of Faster R-CNN.

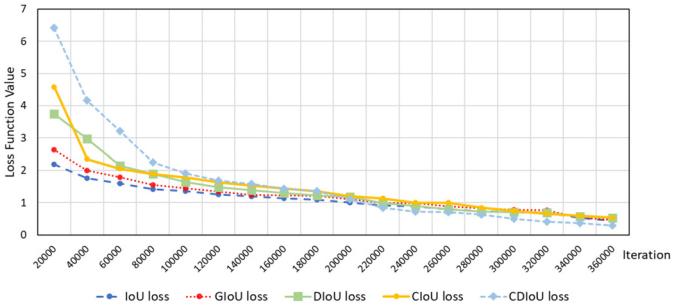


Fig. 8. Loss Function value of ATSS-R-50-FPN-1x.

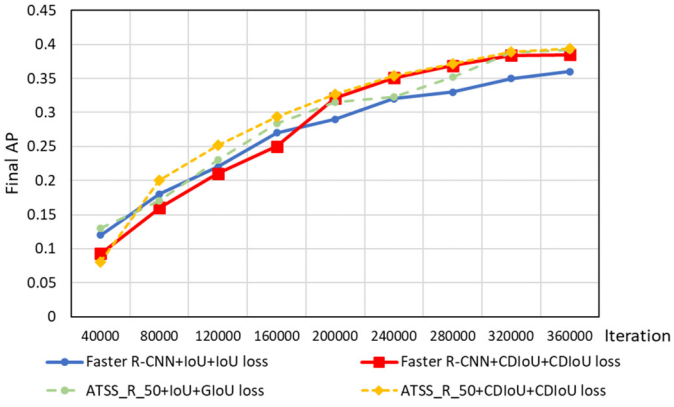


Fig. 9. The final AP of Faster R-CNN and ATSS-R-50-FPN-1x.

to the other loss functions (IoU loss, GIoU loss, DIoU loss, and CIoU loss). As the number of iterations increases, it is obvious that the model using CDIoU and CDIoU loss reaches a stable state in a less iteration even with larger original loss. When scholars run large detection algorithms, CDIoU and CDIoU loss can save a lot of running time to get the same performance. Fig. 7 and 8 show the advantages of CDIoU and CDIoU loss in terms of speed.

Fig. 9 shows that the CDIoU and CDIoU loss functions can help the models achieve higher AP values with fewer iterations. These results show that the CDIoU and CDIoU loss functions have strong convergence and highlight their more accurate evaluation of region proposals. The methods with CDIoU and CDIoU loss get better performance in less iteration, compared to original methods. Fig. 9 shows the advantages of CDIoU and CDIoU loss in terms of performance and speed.

In order to rigorously verify the effectiveness of CDIoU and CDIoU loss function proposed in this paper, a large number of comparative experiments are designed to suggest supporting evidence. We can learn from Table 6 that CDIoU loss can indeed achieve better results than IoU loss, GIoU loss, DIoU loss, etc., while using the same IoU section.

By comparing the above experiments, we can observe that using different IoU modules with different IoU loss functions yields

[2] Code is available in https://www.github.com/Alan-D-Chen/CDIoU-CDIoUloss

Table 3
Detection models results with and without IoUs and IoU loss function on the **MS COCO** `test-dev` set. MS means multi-scale testing. + CDIoU & loss means that this model uses CDIoU and CDIoU loss as evaluation-feedback module. + IoU & loss means that this model uses IoU and IoU loss as evaluation-feedback module. The original ATSS models use IoU and GIoU loss as evaluation-feedback module, and the original Faster R-CNN,YOLOv4,RetinaNet-R101,ResNet-50 + NAS-FPN,Detectron2 Mask R-CNN,Cascade R-CNN models use IoU and IoU loss or *L1-smooth* as evaluation-feedback module. Bold fonts indicate the best performance. 1*x* and 2*x* mean the model is trained for 90K and 180K iterations, respectively.

Models	test-dev(%)			
	AP	AP_s	AP_m	AP_l
Faster R-CNN [12]	36.0	-	-	-
YOLOv4 [12]	41.2	20.4	44.4	56.0
RetinaNet-R101 (1024) [36]	40.8	24.1	44.2	51.2
ResNet-50 + NAS-FPN (1280@384) [5]	45.4	-	-	-
Detectron2 Mask R-CNN R101-FPN [4]	44.3	-	-	-
Cascade R-CNN [34]	42.8	23.7	45.5	55.2
FCOS [21]	43.2	26.5	46.2	53.3
ATSS_R_50_FPN_1x [35]	39.2	-	-	-
ATSS_dcnv2_R_50_FPN_1x [35]	43.0	-	-	-
ATSS_R_101_FPN_2x [35]	43.6	-	-	-
ATSS_dcnv2_R_101_FPN_2x [35]	46.3	-	-	-
ATSS_X_101_32x8d_FPN_2x [35]	45.1	-	-	-
ATSS_dcnv2_X_101_32x8d_FPN_2x [35]	47.7	-	-	-
ATSS_dcnv2_X_101_64x4d_FPN_2x [35]	47.7	-	-	-
Comparison test 1				
ATSS_R_50_FPN_1x + IoU & loss	38.6	20.7	37.4	45.7
ATSS_dcnv2_R_50_FPN_1x + IoU & loss	41.9	24.0	45.8	53.8
ATSS_dcnv2_R_101_FPN_2x + IoU & loss	45.8	25.9	48.6	56.9
ATSS_X_101_32x8d_FPN_2x + IoU & loss	44.5	26.8	47.2	53.2
ATSS_dcnv2_X_101_32x8d_FPN_2x + IoU & loss	46.8	27.9	49.7	58.7
ATSS_dcnv2_X_101_32x8d_FPN_2x(MS) + IoU & loss	49.6	31.2	50.5	60.3
Comparison test 2				
Faster R-CNN + CDIoU & loss	**38.3**	17.3	38.0	54.4
YOLOv4 + CDIoU & loss	**41.4**	20.4	46.1	55.8
RetinaNet-R101 (1024) + CDIoU & loss	**41.2**	22.5	43.1	50.9
ResNet-50 + NAS-FPN (1280@384) + CDIoU & loss	**45.8**	22.1	48.0	65.1
Detectron2 Mask R-CNN R101-FPN + CDIoU & loss	**45.0**	21.3	46.0	64.9
Cascade R-CNN + CDIoU & loss	**43.0**	25.0	45.7	66.1
ATSS_R_50_FPN_1x + CDIoU & loss	**39.4**	22.5	42.2	49.8
ATSS_dcnv2_R_50_FPN_1x + CDIoU & loss	**43.1**	24.4	46.0	55.8
ATSS_dcnv2_R_101_FPN_2x + CDIoU & loss	**46.4**	27.8	49.7	58.6
ATSS_X_101_32x8d_FPN_2x + CDIoU & loss	**45.2**	27.8	48.3	55.2
ATSS_dcnv2_X_101_32x8d_FPN_2x + CDIoU & loss	**47.9**	29.6	50.8	60.5
ATSS_dcnv2_X_101_32x8d_FPN_2x(MS) + CDIoU & loss	**50.7**	33.2	52.6	62.7

Table 4
Detector results with CDIoU and CDIoU loss. CDIoU means that this model uses CDIoU and CDIoU loss. *diff.* means that the difference between ATSS, DETR, Swin Transformer and D2Det with or without CDIoU & loss.

Methods	ATSS	ATSS(MS)	DETR [37]	D2Det [38]	Swin Transformer [1]
originals	46.8	49.6	44.9	47.4	58.7
CDIoU	47.9	50.7	47.1	48.6	60.0
diff.	1.2	1.1	2.2	1.2	1.3

Table 5
Comparison of effects and running results(%) of various IoU losses on the **MS COCO** `val` set. The default evaluation module IoUs is IoU in this table. The FPS of the same model with different loss terms vary so much in Table 5, because different loss terms depend on different IoUs. Even if GIoU, CIoU, DIoU and CDIoU are not used in evaluation system, they still need to be calculated in loss terms. Backbone used in Faster R-CNN with FPN is VGG16. Backbone used in ATSS is R_50_FPN.

IoUs loss	Model	AP	FPS
L1-smooth	Faster R-CNN [15]	36.0	7.5
IoU loss		36.8	7.7
GIoU loss		36.9	8.5
DIoU loss		38.0	7.9
CIoU loss		38.2	6.3
CDIoU loss		38.5	7.7
L1-smooth	ATSS [35]	37.5	11.1
IoU loss		38.0	10.8
GIoU loss		39.2	11.0
DIoU loss		39.0	11.3
CIoU loss		39.2	8.8
CDIoU loss		39.4	11.2

different and thought-provoking results. Combining Table 5, we can analyze that under the same IoU loss function condition, using CDIoU module alone can improve the final result by 0.3 $\sim$ 1.8%; under the same IoU module, using CDIoU loss function alone can improve the final result by 0.2 $\sim$ 1.7%.

At the same time, if IoU module and IoU loss function could be unified in the computational form (eg.GIoU + GIoU loss function and CDIoU + CDIoU loss function), the final result would seem to be better than the sum of the results of using the two optimization schemes independently.

In order to verify the independence and validity of CDIoU and CDIoU loss function, we designs the following comparison test in Table 9, using original IoU + IoU loss(or GIoU loss) function and CDIoU + CDIoU loss function in training and validation stages of the program respectively, and finally comparing their AP results. We use CDIoU and CDIoU loss in the training and validation stages respectively, which can improve the final AP. If we used CDIoU and

Table 6
Comparison of effects and running results(%) of various IoUs and IoU loss functions on the **MS COCO** `val` set. For more information about results of evaluation module IoU, please refer to Table 5. IoU and IoU loss are still the most common evaluation-feedback modules, and we still regard them as the primary comparison standards just like GIoU, DIoU, and CIoU's authors do. The ablation study shows that the evaluation system and feedback mechanism coordinate with each other to improve AP more significantly. However, we find the effect is small if you only change one of two. Faster R-CNN: 36.9%⟶38.5%; ATSS R 50 FPN 1x:38.0%⟶39.5% in Table 6. CDIoU & CDIoU loss are proved to improve AP by 2.5% to 4.0% in non-mainstream datasets by Github and local Lab researchers.

Model	IoUs		IoU loss functions					AP
	GIoU	CDIoU	IoU loss	GIoU loss	DIoU loss	CIoU loss	CDIoU loss	
Faster R-CNN [15]	✓		✓					36.9
	✓			✓				37.3
	✓				✓			38.0
	✓					✓		38.2
	✓						✓	38.3
Faster R-CNN [15]		✓	✓					37.1
		✓		✓				37.3
		✓			✓			38.1
		✓				✓		38.3
		✓					✓	**38.5**
ATSS [35]	✓		✓					38.0
	✓			✓				39.3
	✓				✓			39.0
	✓					✓		39.2
	✓						✓	39.4
ATSS [35]		✓	✓					38.1
		✓		✓				39.3
		✓			✓			39.2
		✓				✓		39.2
		✓					✓	**39.5**

Table 7
Object detectors with different learning rate setting on the **MS COCO** `val` set. Epoches represent the minimum epoches to get the same standard result. The standard result of Faster RCNN is 36.0%, and the standard result of ATSS is 37.5%. Intact ATSS is trained for 90K or 180K iterations, respectively. Backbone used in Faster R-CNN with FPN is VGG16; backbone used in ATSS is R_50_FPN. Faster RCNN and ATSS run with original IoU and IoU loss function in Table 8.

Method	Learning Rate Setting	Epoches
Faster R-CNN	Fixed learning rate (original)	7
	Linear decay	5
	Exponential decay	5
	Floating learning rate decay	4
ATSS	Linear decay (original)	8
	Exponential decay	8
	Floating learning rate decay	6

Table 8
Object detectors with different anchor generation on the **MS COCO** `val` set. FPN is orignal part in Faster RCNN and ATSS. FPN + AGTC represents that FPN generate anchors with automatic GT clustering. Faster RCNN and ATSS run with original IoU and IoU loss function in Table 8. **MS** means multi-scale. *dcnv2* denotes deformable convolutional networks v2.

Method	Backbone	Anchor generation	AP
Faster RCNN	VGG16	FPN	36.0
		FPN + AGTC	37.4
	R_50_FPN	FPN	38.6
		FPN + AGTC	39.8
	dcnv2_R_50_FPN	FPN	41.9
		FPN + AGTC	42.7
ATSS	dcnv2_R_101_FPN	FPN	45.8
		FPN + AGTC	46.4
	X_101_32x8d_FPN	FPN	44.5
		FPN + AGTC	44.9
	dcnv2_X_101_32x8d	FPN	46.8
		FPN + AGTC	47.7
	dcnv2_X_101_32x8d(MS)	FPN	49.6
		FPN + AGTC	50.5

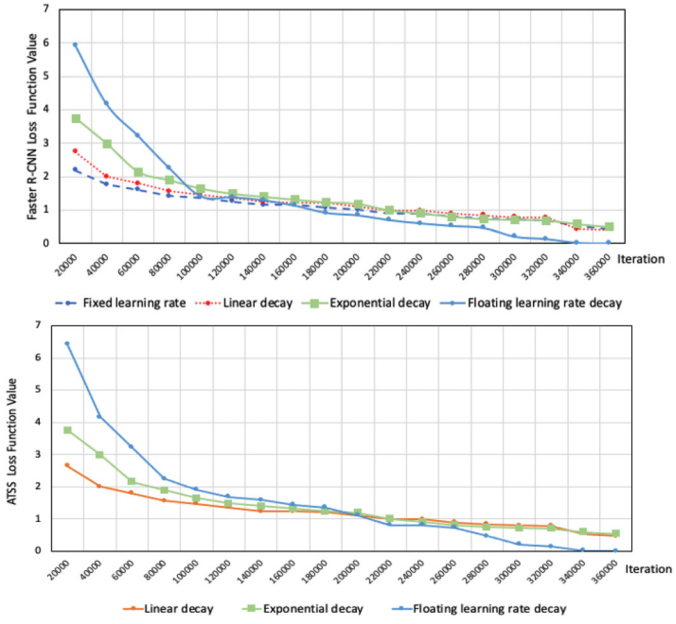


Fig. 10. The comparison of detectors with different learning rate setting on the **MS COCO** `val` set. Epoches represent the minimum epoches to get the same standard result.

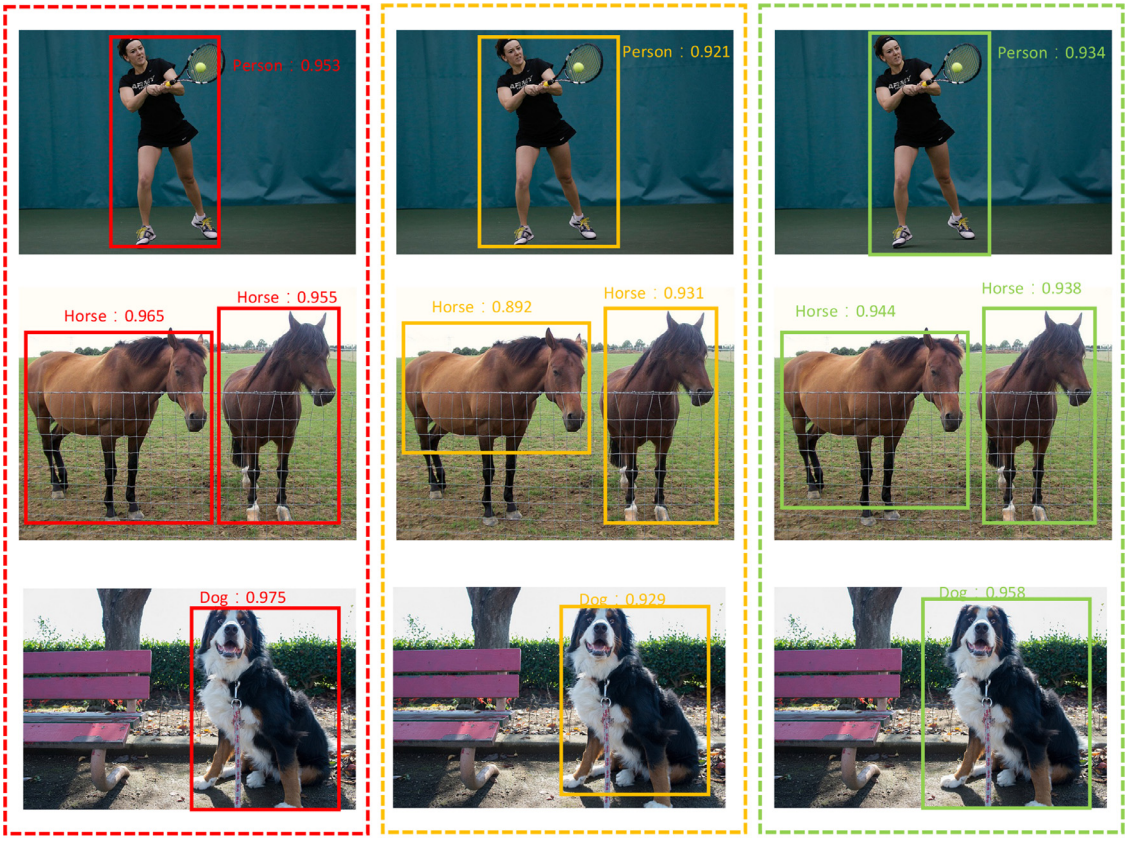
CDIoU loss in both training and validation stages, we could get more significant improvement results.

At the same time, we set up experiments on Faster RCNN and ATSS method to verify float learning rate decay and AGTC. Experiments in Table 7 and Figure 10 show that floating learning rate decay can effectively improve the efficiency of model operation.

According to experiments with different backbones on Faster RCNN and ATSS, we can infer that float learning rate decay is unable to improve results of detectors, but float learning rate decay is able to reduce running time evidently, comparing to fixed learning

Table 9
Comparison of effects and running results(%) in training and validation on the **MS COCO** `train` and `val` set. *Originals* means that original Faster R-CNN uses IoU+IoU loss as evaluation-feedback module and original ATSS (backbone: _R_50_FPN_1x) uses IoU+GIoU loss as evaluation-feedback module.

Model	Training			Validation		AP
	IoU loss	GIoU loss	CDIoU loss	Originals	CDIoU loss	
Faster R-CNN	✓			✓		36.8
	✓				✓	37.2
			✓	✓		37.7
			✓		✓	**38.5**
ATSS		✓		✓		39.2
		✓			✓	39.3
			✓	✓		39.3
			✓		✓	**40.2**



(a) CDIoU and CDIoU loss (b) IoU and IoU loss (c) GIoU and GIoU loss

Fig. 11. Comparison of CDIoU-CDIoU loss, IoU-IoU loss and GIoU-GIoU loss under ATSS (backbone: _R_50_FPN_1x) on COCO. CDIoU-CDIoU loss have better performance than IoU-IoU loss and GIoU-GIoU loss.

rate, linear decay, and exponential decay. Float learning rate decay can reduce the running time by 10 ∼ 15% on average.

AGTC module significantly improved the results of object detectors, particularly in detectors with smaller backbone net. Table 8 shows an obvious trend. With the deepening of backbone network, the results (AP) produced by FPN and FPN + AGTC gradually approach and are roughly the same in the ATSS with dcnv2_X_101_32x8d_FPN and multi-scale.

6.4. Analysis of experiments

The improvement effect of CDIoU + CDIoU loss tends to decrease as the model is updated. First, as the backbone of the model deepens, the model itself enhances the strength of feature extraction. Second, the continuous improvement of FPN modules also optimizes the function of traditional evaluation systems. The above two points offset the advantages of CDIoU and CDIoU loss compared with the traditional evaluation-feedback modules.

ATSS bridges the gap between anchor-based and anchor-free detection via adaptive training sample selection. Comparison tests on ATSS exclude the essential interference between anchor-based and anchor-free detection. In these tests, the interference of positive and negative sample generation is eliminated, which give tests based on ATSS more representativeness.

From Table 9, we can clearly observe that replacing IoU module and loss function separately can improve the results of the original model, and replacing IoU module and loss function at the same time also has a certain additive effect, achieving the synergy of "one plus one is greater than two". It lies on the calculation form consistency between evaluation system and feedback mechanism.

Table 10
The results (%) of detectors combined with different components on the **MS COCO** `val` set. ✓ means that detectors run with this component. Original Faster RCNN setting: VGG16 + IoU & IoU loss + fixed learning rate + FPN. Original ATSS setting: R_50_FPN + GIoU & GIoU loss + linear learning rate decay + FPN.

Modules	CDIoU loss	FLRD	AGTC + FPN	AP
Faster RCNN				36.0
	✓			37.7
	✓	✓		37.7
	✓	✓	✓	39.1
ATSS				39.2
	✓			39.4
	✓	✓		39.4
	✓	✓	✓	40.8

The numerical fluctuations of the feedback mechanism reflect the differences of the evaluation system, which makes the evaluation-feedback module more targeted (see Figure 11).

From Figure 11, it is obvious that the performancee of CDIoU is much better than that of IoU and GIoU. First of all, from the comparison of three groups in Fig. 11, it can be clearly seen that the classification effect of CDIoU is better than that of IoU and GIoU. Secondly, the regression (positioning) effect of CDIoU is also better than that of IoU and GIoU. The region proposal of IoU and GIoU will split the whole object (for example, the horse's leg and the dog's tail in yellow and green box).

Table 10 represents CAF (CDIoU & CDIoU loss, AGTC, and FLRD) have obvious phased improvement. There are experiments with or without CAF on object detectors in Table 11 and 12. Detectors with CAF have been greatly improved compared with the original object detection models. There is a maximum AP improvement of 2.9% and an average AP of 1.1% improvement on MS COCO and a maximum AP improvement of 8.2% and an average AP of 3.7% improvement on Visdrone dataset.

Table 12
Detection models results with and without CAF module on the **VisDrone-DET2021. + CAF** means that this model uses CDIoU and CDIoU loss, AGTC, and FLRD. **+ IoU & loss** means that this model uses IoU and IoU loss as evaluation-feedback module. The original ATSS models use IoU and GIoU loss as evaluation-feedback module, and the original Faster R-CNN,YOLOv4,RetinaNet-R101,ResNet-50 + NAS-FPN,Detectron2 Mask R-CNN,Cascade R-CNN models use IoU and IoU loss or *L1-smooth* as evaluation-feedback module. *dcnv2* denotes deformable convolutional networks v2. 1*x* and 2*x* mean the model is trained for 90K and 180K iterations, respectively.

Models	VisDrone-DET2021(%)		
	AP	AP_{50}	AP_{75}
Comparison test 3			
Faster R-CNN [12]	32.5	-	-
YOLOv4 [12]	35.8	50.4	43.7
RetinaNet-R101 (1024) [36]	40.8	54.1	44.2
ResNet-50 + NAS-FPN (1280@384) [5]	42.4	-	-
Detectron2 Mask R-CNN R101-FPN [4]	44.3	-	-
Cascade R-CNN [34]	39.8	-	-
FCOS [21]	40.1	-	-
ATSS_R_50_FPN_1x + IoU & loss	34.9	50.5	37.9
ATSS_dcnv2_R_50_FPN_1x + IoU & loss	35.0	54.0	39.8
ATSS_dcnv2_R_101_FPN_2x + IoU & loss	35.9	56.9	38.9
ATSS_X_101_32x8d_FPN_2x + IoU & loss	35.5	56.0	40.8
ATSS_dcnv2_X_101_32x8d_FPN_2x + IoU & loss	37.9	57.5	39.9
ATSS_dcnv2_X_101_32x8d_FPN_2x(MS) + IoU & loss	39.6	61.5	45.0
Comparison test 4			
Faster R-CNN + CAF	**34.8**	57.5	39.4
YOLOv4 + CAF	**37.8**	60.5	40.9
RetinaNet-R101 (1024) + CAF	**42.0**	62.8	44.4
ResNet-50 + NAS-FPN (1280@384) + CAF	**42.9**	64.5	48.8
Detectron2 Mask R-CNN R101-FPN + CAF	**43.8**	61.3	45.1
Cascade R-CNN + CAF	**41.9**	66.7	44.2
ATSS_R_50_FPN_1x + CAF	**37.4**	63.2	41.0
ATSS_dcnv2_R_50_FPN_1x + CAF	**38.4**	64.9	42.0
ATSS_dcnv2_R_101_FPN_2x + CAF	**39.8**	67.8	42.4
ATSS_X_101_32x8d_FPN_2x + CAF	**40.7**	68.7	45.3
ATSS_dcnv2_X_101_32x8d_FPN_2x + CAF	**45.2**	69.6	49.0
ATSS_dcnv2_X_101_32x8d_FPN_2x(MS) + CAF	**47.8**	73.2	51.4

Table 11
Detection models results with and without CAF module on the **MS COCO** `test-dev` set. **+ CAF** means that this model uses CDIoU and CDIoU loss, AGTC, and FLRD. **+ IoU & loss** means that this model uses IoU and IoU loss as evaluation-feedback module. The original ATSS models use IoU and GIoU loss as evaluation-feedback module, and the original Faster R-CNN,YOLOv4,RetinaNet-R101,ResNet-50 + NAS-FPN,Detectron2 Mask R-CNN,Cascade R-CNN models use IoU and IoU loss or *L1-smooth* as evaluation-feedback module. *dcnv2* denotes deformable convolutional networks v2. 1*x* and 2*x* mean the model is trained for 90K and 180K iterations, respectively.

Models	test-dev(%)			
	AP	AP_s	AP_m	AP_l
Comparison test 1				
Faster R-CNN [12]	36.0	-	-	-
YOLOv4 [12]	41.2	20.4	44.4	56.0
RetinaNet-R101 (1024) [36]	40.8	24.1	44.2	51.2
ResNet-50 + NAS-FPN (1280@384) [5]	45.4	-	-	-
Detectron2 Mask R-CNN R101-FPN [4]	44.3	-	-	-
Cascade R-CNN [34]	42.8	23.7	45.5	55.2
FCOS [21]	43.2	26.5	46.2	53.3
ATSS_R_50_FPN_1x + IoU & loss	38.6	20.7	37.4	45.7
ATSS_dcnv2_R_50_FPN_1x + IoU & loss	41.9	24.0	45.8	53.8
ATSS_dcnv2_R_101_FPN_2x + IoU & loss	45.8	25.9	48.6	56.9
ATSS_X_101_32x8d_FPN_2x + IoU & loss	44.5	26.8	47.2	53.2
ATSS_dcnv2_X_101_32x8d_FPN_2x + IoU & loss	46.8	27.9	49.7	58.7
ATSS_dcnv2_X_101_32x8d_FPN_2x(MS) + IoU & loss	49.6	31.2	50.5	60.3
Comparison test 2				
Faster R-CNN + CAF	**38.9**	17.3	39.2	56.4
YOLOv4 + CAF	**41.8**	20.8	47.1	56.7
RetinaNet-R101 (1024) + CAF	**42.5**	22.5	43.3	54.8
ResNet-50 + NAS-FPN (1280@384) + CAF	**46.9**	23.5	449.5	67.0
Detectron2 Mask R-CNN R101-FPN + CAF	**45.9**	21.3	48.0	66.0
Cascade R-CNN + CAF	**44.0**	26.0	46.8	68.0
ATSS_R_50_FPN_1x + CAF	**40.1**	23.5	44.8	42.6
ATSS_dcnv2_R_50_FPN_1x + CAF	**43.4**	24.4	48.0	58.0
ATSS_dcnv2_R_101_FPN_2x + CAF	**47.3**	27.8	51.0	59.9
ATSS_X_101_32x8d_FPN_2x + CAF	**45.9**	27.8	50.3	57.0
ATSS_dcnv2_X_101_32x8d_FPN_2x + CAF	**48.2**	29.6	52.0	62.4
ATSS_dcnv2_X_101_32x8d_FPN_2x(MS) + CAF	**50.8**	33.2	53.7	64.0

7. Conclusion

In previous works, scholars have focused on the designing of neural networks and the tuning of convolution structures, but ignored evaluation-feedback module, which improve detectors performance in CAF manner. In this paper, we propose that an efficient regression detector should focus on the evaluation-feedback mechanism, automatic ground truth clustering, and float learning rate decay. CDIoU & CDIoU loss, AGTC, and FLRD are proposed to improve the performance of object detection models without increasing running time and parameters. Through a large number of experiments, it is evidenced that the models improve their AP significantly by using CAF modules on MS COCO and Visdrone dataset, comparing to traditional object detection models.

At the same time, this method also has some technical weaknesses. This paper aims to improve the performance of 2D object detection. But CAF modules can not deal with detection task in 3D and video. And AGTC module is highly dependent on datasets and lacks generalization ability. Secondly, CAF module does not significantly reduce the size and running time of model. Thirdly, CAF module ignores the role of medium objects and large objects in general object detection, and only pays attention to the impact of small objects on general object detection performance.

In the future work, we will mainly solve the application of CAF modules in 3D object detection and video object detection. At the same time, we should pay attention to size-different objects customarily. Size-different objects should use different detection strategies. Customized solutions should be adopted for various objects in computer vision in the future. Object detection is a basic task, while semantic segmentation and instance segmentation need to face more complex data. We do hope that our work will play a role of cornerstone to encourage the evaluation-feedback mechanism in computer vision subtasks (such as object detection, semantic segmentation, and instance segmentation) with less time and lighter model size.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The data that has been used is confidential.

Acknowledgements

This research was supported in part by the National Natural Science Foundation of China 61,976,158 and Grant Nos. 62006172.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2022.109256.

References

[1] Y. Chen, P. Zhang, T. Kong, Y. Li, X. Zhang, L. Qi, J. Sun, J. Jia, Scale-Aware Automatic Augmentations for Object Detection with Dynamic Training, IEEE, 2022.
[2] M. Tan, R. Pang, Q.V. Le, Efficientdet: Scalable and efficient object detection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[3] C. Wang, C. Zhong, Adaptive feature pyramid networks for object detection, IEEE Access 9 (2021) 107024–107032.
[4] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, R. Girshick, Detectron2, 2019, (https://www.github.com/facebookresearch/detectron2).
[5] S. Wang, An augmentation small object detection method based on nas-fpn, in: 2020 7th International Conference on Information Science and Control Engineering (ICISCE), IEEE, 2020, pp. 213–218.
[6] B. Zoph, E.D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, Q.V. Le, Learning data augmentation strategies for object detection, in: European Conference on Computer Vision, Springer, 2020, pp. 566–583.
[7] P. Gao, M. Zheng, X. Wang, J. Dai, H. Li, Fast convergence of detr with spatially modulated co-attention, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3621–3630.
[8] Z. Gao, L. Wang, B. Han, S. Guo, Adamixer: A fast-converging query-based object detector, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5364–5373.
[9] Q. Zhang, Y.-B. Yang, Rest: an efficient transformer for visual recognition, Adv. Neural. Inf. Process. Syst. 34 (2021) 15475–15485.
[10] B. Wu, J. Gu, Z. Li, D. Cai, X. He, W. Liu, Towards efficient adversarial training on vision transformers, in: European Conference on Computer Vision, Springer, 2022, pp. 307–325.
[11] Z. Zheng, P. Wang, W. Liu, J. Li, D. Ren, Distance-iou loss: Faster and better learning for bounding box regression, in: AAAI Conference on Artificial Intelligence, 2020.
[12] M. Maity, S. Banerjee, S.S. Chaudhuri, Faster r-cnn and yolo based vehicle detection: A survey, in: 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), IEEE, 2021, pp. 1442–1447.
[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, in: European conference on computer vision, Springer, 2016, pp. 21–37.
[14] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation (2013).
[15] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149.
[16] X. Han, J. Chang, K. Wang, You only look once: unified, real-time object detection, Procedia Comput. Sci. 183 (2021) 61–72.
[17] Z. Liu, J. Li, Y. Shu, D. Zhang, Detection and recognition of security detection object based on yolo9000, in: 2018 5th International Conference on Systems and Informatics (ICSAI), IEEE, 2018, pp. 278–282.
[18] K. Duan, L. Xie, H. Qi, S. Bai, Q. Huang, Q. Tian, Corner proposal network for anchor-free, two-stage object detection, in: European Conference on Computer Vision, Springer, 2020, pp. 399–416.
[19] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916.
[20] Z. He, L. Zhang, Multi-adversarial faster-rcnn for unrestricted object detection, in: Proceedings of the IEEE International Conference on Computer Vision(ICCV), 2019, pp. 6668–6677.
[21] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2020.
[22] D. Yang, Y. Zhou, A. Zhang, X. Sun, D. Wu, W. Wang, Q. Ye, Multi-view correlation distillation for incremental object detection, Pattern Recognit. 131 (2022) 108863.
[23] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, 2019, pp. 6105–6114.
[24] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O.R. Zaiane, M. Jagersand, U2-Net: going deeper with nested u-structure for salient object detection, Pattern Recognit. 106 (2020) 107404.
[25] G. Wen, P. Cao, H. Wang, H. Chen, X. Liu, J. Xu, O. Zaiane, Ms-ssd: multi-scale single shot detector for ship detection in remote sensing images, Appl. Intell. (2022) 1–19.
[26] X. Zhu, H. Hu, S. Lin, J. Dai, Deformable convnets v2: More deformable, better results, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[27] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, arXiv preprint arXiv:1706.05587 (2017).
[28] Z. Hou, X. Liu, L. Chen, Object detection algorithm for improving non-maximum suppression using giou, in: IOP Conference Series: Materials Science and Engineering, volume 790, IOP Publishing, 2020, p. 012062.
[29] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions (2016).
[30] M. Yang, L. Liao, K. Ke, G. Gao, Multi-feature sparse similar representation for person identification, Pattern Recognit. (2022) 108916.
[31] H. Wang, Q. Wang, P. Li, W. Zuo, Multi-scale structural kernel representation for object detection, Pattern Recognit. 110 (2021) 107593.
[32] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377.
[33] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, N. Sebe, Binary neural networks: a survey, Pattern Recognit. 105 (2020) 107281.
[34] Z. Cai, N. Vasconcelos, Cascade r-cnn: Delving into high quality object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6154–6162.
[35] S. Zhang, C. Chi, Y. Yao, Z. Lei, S.Z. Li, Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection, CVPR, 2020.
[36] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Doller, Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2999–3007.
[37] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End–to-end object detection with transformers, in: European Conference on Computer Vision, Springer, 2020, pp. 213–229.

[38] J. Cao, H. Cholakkal, R.M. Anwer, F.S. Khan, Y. Pang, L. Shao, D2det: Towards high quality object detection and instance segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11485–11494.

Chen Dong graduated from Jinan University with a bachelor's degree and now studies for a doctorate in Tongji University. The research fields are computer vision, artificial intelligence, deep learning and object detection.

Duoqian Miao, Professor of College of Electronics and Information Engineering of Tongji University, Fellow of International Rough Set Society (IRSS), Fellow of Chinese Association for Artificial Intelligence (CAAI),of. Miao works in Department of Computer Science and Technology of Tongji University, and Key Laboratory of Embedded System and Service Computing Ministry of Education as vice director. Prof. Miao's research interests include Artificial Intelligence, Machine Learning, Big Data Analysis, Granular Computing and Rough Sets, etc. Prof. Miao has published about 180 scientific articles in "IEEE Transactions on Cybernetics", "IEEE Transactions on Information Forensics and Security", "IEEE Transactions on Fuzzy Systems", Pattern Recognition", "Information Sciences" and so on. Over 100 articles have been cited by SCI or EI,of which areI papers. In addition, he has published 10 academic books and owned 12 patents. Prof. Miao leads 6 projects supported by National Natural Science Foundation of China, 2 project supported by Research Fund for the Doctoral Program of Higher Education. He also participates in other projects as a key member, such as 1 project supported by 973 Program, 1 project supported by 863 Program, 1 project supported by National Major Research Program 1 project supported by National Key R&D Program, 2 courses sponsored by Shanghai Education Commission. Representative awards include the Second Prize of Wuwenjun AI Science and Technology(2018),the First Prize of Natural Science of Chongqing(2010), the First Prize of Technical Invention of Shanghai(2009), the First Prize of Ministry of Education Science and Technology Progress Award(2007). Prof. Miao is president of International Rough Sets Society (IRSS), vice president of Shanghai Computer Federation, vice president of Shanghai Association for Artificial Intelligence, appraisal expert of Department of Information Sciences of National Natural Science Foundation, chair of CAAI Granular Computing and Knowledge Discovery Technical Committee and standing member of CCAI Machine Learning Technical Committee, distinguished member of China Computer Federation(CCF), vice chair of teaching guidance committee for computer science and technology in Shanghai. Prof. Miao is editors of International Journal of Approximate Reasoning, CAAI Transactions on Intelligence Technology, Journal of Computer Research and Development, Journal of Electronics and Information Technology, Journal of Frontiers of Computer Science & Technology, Journal of Tongji University (Natural Science), etc.