Pedestrian Detection using YOLO with D-IoU and C-IoU

Hao Li
Department of Computer Science
University of Canterbury
Christchurch, New Zealand
hli128@uclive.ac.nz

Richard Green
Department of Computer Science
and Software Engineering
University of Canterbury
Christchurch, New Zealand
richard.green@canterbury.ac.nz

***Abstract*—This paper proposes a method to improve the performance of pedestrian detection. The method is based on the You Only Look Once (YOLO) algorithm and the improved Intersection over Union (IoU) loss function. In pedestrian detection, the ultimate regression from many bounding boxes to the target box of pedestrians is a critical step. The earliest proposed IoU loss function has strong limitations in the case of overlapping bounding boxes. The later generalised-IoU (GIoU) loss function also has insufficient regression performance and convergence speed. Therefore, this paper proposes Distance-IoU (DIoU) loss and Complete IoU (CIoU) loss. DIoU can minimise the normalised distance between the centre points of two bounding boxes to improve the convergence speed. The CIoU comprehensively considers and combines the characteristics of previous loss functions including DIoU and considers the aspect ratio of bounding box to increase the accuracy of regression. The experiments show that the CIoU loss has a relatively increased average precision of 6.5%, which contributes to better pedestrian detection in the future.**

Keywords—Pedestrian detection, YOLO, DIoU, CIoU, loss function

I. INTRODUCTION

Pedestrian detection has always been a popular research topic in the field of computer vision. It can be applied to many scenes in life, such as Autonomous Driving, Intelligent Monitoring and Intelligent Robot Systems [1]. Dalal's Histograms of Oriented Gradients (HOG) has been the classic detection method in the past [2]. It has excellent performance on the MIT pedestrian database, but HOG is easily affected by occlusions and background clutter. With the rise of Machine Learning in Artificial Intelligence, some excellent pedestrian detection algorithms have emerged, such as the real-time algorithm You Only Look Once (YOLO) used in this article. Accurately determining the pedestrian bounding box is the ultimate goal of these algorithms. However, these algorithms inevitably have some positioning errors, and the frequency of occurrence even exceeds the classification errors [3]. A better loss function can reduce the impact of background class imbalance, improve positioning accuracy and convergence speed [4], so a better design of loss function has become a timeless research, the purpose of this article is to better design the loss function to reduce these negative effects. The following are the specific work:

- Stating the results and limitations of prior research.
- Introducing the improved loss function and related experiments
- Comparing and analysing the results of this experiment with prior research
- Summarising the limitations of this study and future research directions

II. BACKGROUND

A. *Pedestrian Detection using YOLO*

The YOLO object detection algorithm is a clever Convolutional Neural Network (CNN). Proposed in 2016 [5], it has updated three versions. The latest version is YOLO v3. It has been able to recognise more than 9,000 objects since the arrival of YOLO v2. YOLO introduced a new detection concept: it treats object detection as a regression process to spatially associated class probabilities and separated bounding boxes [5]. While Fast and Faster R-CNN are using neural networks and

sharing computation to propose regions rather than predicting possible bounding boxes [6], the cumbersome steps directly lead to the detection speed is not as good as YOLO. The specific time comparison as Figure 1. Fast YOLO model is a tiny version of the network that could process images 91 frames per second, the normal YOLO processes 45 frames per second [7].

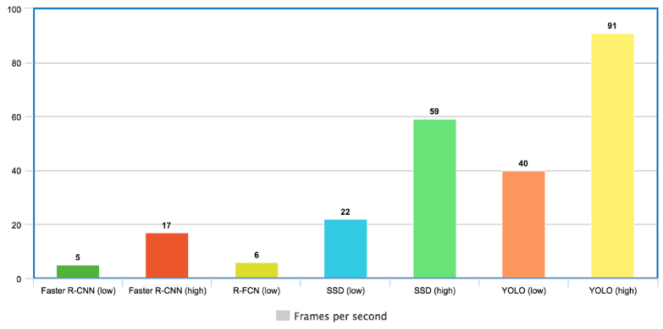

Figure. 1. Object Detection Algorithm Comparison [7]

The latest YOLO v3 uses the Darknet-53 network structure (53 convolutional layers) [8]. The increase of the number of layers means that the more parameters available for adjustment, the greater degree of freedom for adjustment resulting in the better the fitting effect. But it also has negative effects, such as the disappearance of gradients and explosion problems because the network structure is too deep. And as the number of layers increases, there might will also be a degradation problem [12]. The essence of this problem is overfitting due to the loss of information. To this end, Lan added three Passthrough layers to the original YOLO v2 network [11]. Refer to Figure 2. For the feature map of 26 x 26 x 512, the passthrough layer becomes 13 x 13 x 2048. The channel dimension is increased, and the size of feature map is reduced. This network can better learn shallow pedestrian feature information, reduce the false positive (alarm) and recall, improve the detection accuracy of pedestrians, but the detection speed is not ideal, it is only 25 FPS, which is lower than the normal YOLO v2 model.

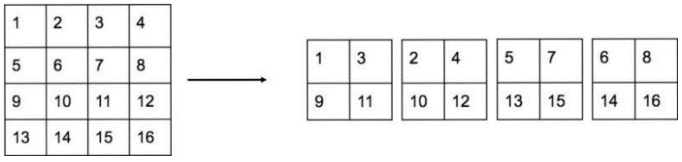

Figure. 2. Passthrough Layer

B. *Loss Function Applied in Models*

The loss function is an essential aspect of designing and training models. It reduces the advantages and disadvantages of a complex system to a scalar value [10], which is "loss". It allows candidate solutions to be compared and evaluated, for example, a set of weights from neurons. The improvements in that "loss" are a sign of a better model.

Lin proposed a Focal Loss function commonly used for one stage detectors like YOLO [19]. This function is designed to solve extreme imbalance between background and foreground classes during training. This function focuses on the instances where it predicted the wrong class. Focal Loss reduces the weight on correctly predicted objects or objects that easy to be correct predictions, putting more weight (impacts) on the objects or outliers that were hard to classify. YOLO v3, however, has conditional class predictions and separate objectness predictions, so YOLO v3 is already robust to the problem that Focal Loss is trying to solve. According to Joseph 's experiments [8], if YOLO v3 uses focal loss, mAP will decrease by 2%, so focal loss does not help to improve the performance of YOLO v3 in pedestrian detection.

Real-world datasets are constantly growing. During the training sample, not all samples are normally distributed. Sometimes a few categories account for most of the data (the red area in Figure 3), and most categories are underrepresented (orange area in Figure 3), which the Long Tail effect is formed. As the number of samples increases, the additional benefits of newly added data points will decrease. Cui, however, proposed Class-Balanced Loss [13], which associates each sample with a smaller neighboring area instead of a single point to measure data overlap and uses the effective number of samples in each class to rebalance the loss. loss can be applied to a wide range of deep networks and loss functions, and can achieve good results. However, it has high requirements on the number of trainings. It takes more than 60 trainings to see its performance advantages [13]. This has high requirements for hardware such as Graphics Processing Unit (GPU), because the training period will otherwise be too long.

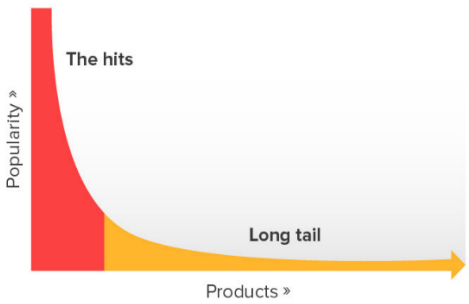

Figure. 3. Long Tail Effect

C. *Limitations*

- The changes to network structure of the object detection algorithm can often improve performance in specific fields, such as accuracy. This comes at a cost, however, because the performance of some remaining aspects will decrease. The most representative is speed. Pedestrian detection has high requirements for real-time performance. Once the delay is enhanced, there will be serious consequences, such as collision accidents in Automatic Driving.
- Some loss functions are not suitable for YOLO series algorithms. This is because YOLO itself is robust, and additional loss function changes will actually reduce its performance.
- Pedestrian detection is a specific detection for one class. Some loss functions need to be built on multiple class classifications for comprehensive comparison and analysis. It is not applicable to detect pedestrians only.
- IoU loss only works when the bounding box has an overlap, and does not provide any moving gradient for non-overlapping cases.
- GIoU can solve the problem of gradient disappearance in non-overlapping situations, but because it relies heavily on IoU, it requires more iterations to converge, especially for horizontal and vertical bounding boxes. GIoU loss in YOLO usually does not converge well, resulting in the detection is not accurate [14].

III. METHOD

A. *Basic IoU loss*

The basic IoU and GIoU loss functions are:

$$IoU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|}$$

$$\mathcal{L} = 1 - IoU + \mathcal{R}(B, B^{gt})$$

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

Where B^{gt} is the ground-truth (target box), it includes w^{gt} h^{gt} x^{gt} y^{gt}. B is the predicted box, it includes w, h, x and y. $R(B, B^{gt})$ is the penalty term for target box and predicted box. C is the smallest enclosing box covering B^{gt} and B. Next, DIoU loss and CIoU loss are generated by redesigning the penalty term.

B. *Distance-IoU loss*

The idea of DIoU Loss is to minimise the normalised distance between central points of bounding boxes [14]. Where b^{gt} and b represents the central points of B and B^{gt}. ρ () calculates the Euclidean distance, c is the diagonal length of the smallest enclosing box covering the two bounding boxes. Finally, the penalty term and DIoU loss function are designed as:

$$\mathcal{R}_{DIoU} = \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2}$$

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

The enclosing box in DIoU loss can be observed through Figure 4. DIoU loss minimises the distance between the two center points (d), while GIoU loss focuses on reducing the area of C - B ∪ B^{gt} [14].

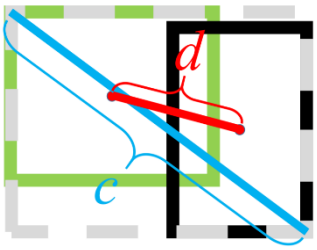

Figure 4. DIoU loss Model

C. *Complete-IoU loss*

CIoU loss considers the effects of multiple loss functions. IoU loss focuses on the overlapping area; GIoU loss focuses on the non-overlapping area; DIoU loss focuses on the overlapping area of the bounding box and the distance between the center points; and CIoU considers the consistency of the bounding box aspect ratio, based on the DIoU loss.

Where α is a positive parameter for trade-off, and v measures the aspect ratio consistency, the factor of overlap area obtains the higher priority for regression; it is more obvious in non-overlapping situations. The additional term is:

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

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

The updated penalty term is:

$$\mathcal{R}_{CIoU} = \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v$$

Then the CIoU loss function is:

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

According to the following function, w2 + h2 as a dominator is usually a small number when h and w are between 0 and 1, this may cause a gradient explosion [14]. In order to stabilise convergence, this paper removes dominator which is w2 + h2, and replaces it with 1, the gradient direction can still be consistent.

$$\frac{\partial v}{\partial w} = \frac{8}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}) \times \frac{h}{w^2 + h^2},$$
$$\frac{\partial v}{\partial h} = -\frac{8}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}) \times \frac{w}{w^2 + h^2}.$$

The core idea and pattern of DIoU and CIoU loss to deal with the regression problem have not changed. They covered the features of the previous loss function and can provide the mobile strategy and direction for the bounding box when it does not overlap with the target box. When the two bounding boxes match exactly, LIoU = LGIoU = LDIoU = 0. If the two bounding boxes do not match, LGIoU = LDIoU = 2. Referring to Figure 5, however, the number of convergence times of DIoU loss is lower, which also makes the convergence speed faster. When containing two bounding boxes in the horizontal or vertical direction, the convergence speed of DIoU loss is still very fast, while CIoU loss has almost the same performance as IoU loss, namely $|C - A \cup B| \rightarrow 0$ [14].

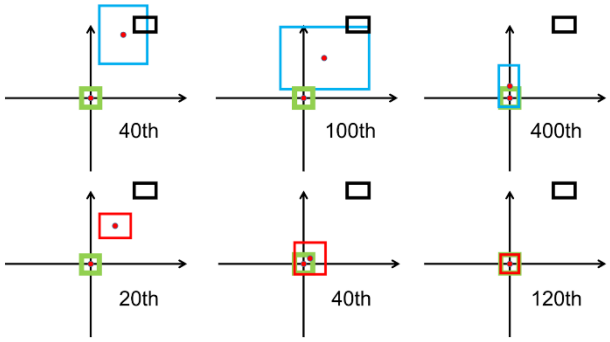

Figure 5. Regression times in GIoU loss (First row) and DIoU (second row). Green represents target box, black represents anchor box, red and blue represent predicted boxes for DIoU and GIoU loss [16].

Non-Maximum Suppression (NMS) algorithm is a method commonly used in pedestrian detection in the post-processing stage. IoU metric is usually used to suppress redundant bounding boxes. However, due to the simple constraints, the NMS algorithm sometimes cannot effectively eliminate missed and erroneous pedestrian detection results. Especially in highly overlapping dense object scenes [15]. Therefore, applying DIoU to NMS is a good choice, as only overlapping regions are considered in the suppression criterion. When the distance between centre points of two bounding boxes is not considered, many unnecessary iterations will be performed. The method of applying DIoU is also flexible, because only a few lines of code need to be added to the pedestrian detection pipeline.

IV. RESULTS

In this paper, the source of the training and testing dataset is PASCAL Visual Object Classes (VOC) 2007, which contains 9963 images. There are 50 iterations in the training, done on the GPU. This training mainly uses LIoU as the baseline for comparison, and tests GIoU loss, DIoU loss, CIoU loss, and CIoU loss combined with DIoU-NMS. Their performance indexes are compared and analysed by detection speed (FPS) and Average Precision (AP). The developing environments in the experiment as following Table 1:

Operating System	Linux Mint 19
Central Processing Unit (CPU)	Intel Core i7-9700KF
Graphics Processing Unit (GPU)	GeForce RTX 1080
IDE	Wing101
Programming Language	Python 3.6
RAM	16 GB
Frameworks	keras
Backend	TensorFlow

Table 1. Developing Environments

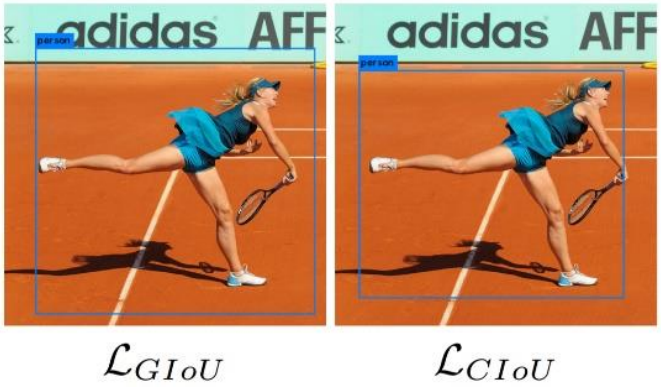

Figure 6. Comparison between GIoU Loss and CIoU

The Distance-IoU (DIoU) loss and Complete IoU (CIoU) loss proposed in this paper were tested in the YOLO v3 algorithm. The results show that it is helpful to improve the performance of pedestrian detection. As shown in Table 2, GIoU loss which is a common version of IoU can improve the detection of certain aspects of performance, but it is not significant. DIoU loss only considers the distance of the centre points of bounding boxes, while AP can increase by 2.54. Furthermore, after considering the aspect ratio consistency of CIoU loss, the AP is improved by 3.27, which is 6.1% higher than the baseline (IoU loss), a significant performance

improvement. The idea of applying DIoU to NMS mentioned earlier in this article has also been applied and tested with CIoU loss to confirm that this is an effective idea. The AP increased by 2.97 (6.5%), which is higher than the experimental results of Zheng 0.6%.

	Average Precision (AP)	Relative Changes
LIoU	45.24	0
LGIoU	46.30	+1.06
LDIoU	47.78	+2.54
LCIoU	48.02	+2.78
LCIoU + [DIoU-NMS]	48.21	+2.97

Table 2. YOLO v3 Performance of AP in different loss functions

As the loss function becomes more complicated, the speed will fluctuate. This experiment, therefore, records the detection speed. Table 3 shows that from the start of GIoU loss until the application of CIoU and loss of DIoU-NMS, the detection speed is in a state of decline. However, the decrease of DIoU is smaller than that of GIoU, which proves that the convergence speed of the bounding box is faster. In the end, however, the detection speed decreased by 1.46 FPS (4.9%) compared to the baseline, which affected the detection performance of YOLO v3. Relative to the improvement of accuracy, the decrease in speed is also the limitation of DIoU and CIoU. YOLO v3 has a similar situation than YOLO v2 [8] [9]. In addition, due to the private nature of the training and testing environment, the computing power of the hardware is not prominent, which leads to the setting of the number of iterations when training the data set is not too high, and the medium size of data set is also required. The YOLO v3 model trained in such an environment may have immature issues [17].

	FPS	Relative Changes
LIoU	29.47	0
LGIoU	28.92	-0.48
LDIoU	29.41	-0.06
LCIoU	28.90	-0.57
LCIoU + DIoU-NMS	28.01	-1.46

Table 3. YOLO v3Performace of detection speed in different loss functions

V. CONCLUSION

This paper proposes two loss functions. DIoU loss is used to minimise the normalised distance between central points of bounding boxes, while CIoU loss is based on DIoU loss. The paper also considers the consistency of the aspect ratio of the bounding box, as well as the practice of DIoU NMS. After testing on the YOLO v3 algorithm, the convergence AP of the bounding box is improved by 2.97, and the convergence speed is 0.42 FPS faster than the previous GIoU loss. This shows that the method proposed here can improve the performance when detecting pedestrians. Comparing similar studies conducted by Zheng before, average precision is 0.6% higher than his experimental results [14]. The detection speed in the experiment is slower than Joseph's results [18].

Based on the practice of method in this paper, our method has achieved good preliminary results. There is still, however, potential for improvement in the future, to overcome the existing limitations and achieve better results:

- Adding multiple dataset sources to train the model can improve robustness. So that it can maintain performance in detecting pedestrians in different environments, such as ignoring the change of camera angle and the complexity of the background.
- Testing the method on multiple algorithms to find more suitable cooperation, such as Faster R-CNN and SSD. Because the design concept of each algorithm is different, the replacement of the loss function may cause the original performance to increase or decrease.
- Further improve of the network structure in deep learning algorithms, using a more powerful training environment (hardware computing) to train better detection models.

REFERENCES

[1] Zhao and B. Chen, "Real-Time Pedestrian Detection Based on Improved YOLO Model," *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC),* Hangzhou, China, 2019, pp. 25-28, doi: 10.1109/IHMSC.2019.10101.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.

[3] J. Tao, H. Wang, X. Zhang, X. Li and H. Yang, "An object detection system based on YOLO in traffic scene," *2017 6th International Conference on Computer Science and Network Technology (ICCSNT),* Dalian, 2017, pp. 315-319, doi: 10.1109/ICCSNT.2017.8343709.

[4] Z. Zhao and X. Lei, "Improved Real-time Pedestrian Detection Method," *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT),* Dalian, China, 2019, pp. 298-302, doi: 10.1109/ICCSNT47585.2019.8962471.

[5] Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi, "You only look once: Unified real-time object detection", *Proc. IEEE Conf. Comp. Vis. Patt. Recogn,* pp. 779-788, 2016.

[6] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal net-works. arXiv preprint *arXiv:1506.01497,* 2015

[7] Jonathan Hui. “Object detection: speed and accuracy comparison (Faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOv3).” https://medium.com/@jonathan_hui/object-detection-speed-and-

accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359, Mar 28, 2018

[8] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. CoRR, abs/1804.02767, 2018.

[9] Joseph Redmon and Ali Farhadi, "YOLO9000: better faster stronger", Proc. IEEE Conf. Comp. Vis. Patt. Recogn., pp. 7263-7271, 2017.

[10] Russell Reed and Robert J MarksII. *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. Location: MIT Press, 1999, p155

[11] W. Lan, J. Dang, Y. Wang and S. Wang, "Pedestrian Detection Based on YOLO Network Model," *2018 IEEE International Conference on Mechatronics and Automation (ICMA),* Changchun, 2018, pp. 1547-1551, doi: 10.1109/ICMA.2018.8484698.

[12] He, Kaiming, et al. "Deep residual learning for image recognition." arXiv preprint arXiv:1512.03385, 2015.

[13] Y. Cui, M. Jia, T. Lin, Y. Song and S. Belongie, "Class-Balanced Loss Based on Effective Number of Samples," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* Long Beach, CA, USA, 2019, pp. 9260-9269, doi: 10.1109/CVPR.2019.00949.

[14] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-IoU Loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI),* 2020

[15] D. Wang, X. Chen, H. Yi and F. Zhao, "Improvement of Non-Maximum Suppression in RGB-D Object Detection," in *IEEE Access*, vol. 7, pp. 144134-144143, 2019, doi: 10.1109/ACCESS.2019.2945834.

[16] Alvin Prayuda Juniarta Dwiyantoro. “Distance-IoU Loss: An Improvement of IoU-based Loss for Object Detection Bounding Box Regression.” https://medium.com/nodeflux/distance-iou-loss-an-improvement-of-iou-based-loss-for-object-detection-bounding-box-regression-4cbdd23d8660. Jan 2020.

[17] Y. Guo, S. Wang, H. He, L. Sun and S. Ma, "Research on Boat Identification Based on Improved Loss Function of Deep Convolutional Neural Networks," 2019 WRC Symposium on Advanced Robotics and Automation (WRC SARA), Beijing, China, 2019, pp. 278-283, doi: 10.1109/WRC-SARA.2019.8931939.

[18] Joseph Redmon and Ali Farhadi. “YOLO: Real-Time Object Detection.” Internet: https://pjreddie.com/darknet/yolo/, 2018.

[19] T. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318-327, 1 Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.