



Tiny Focal Loss: A Lightweight and Effective Solution for Tiny Object Detection Based on Scale-Adaptive Optimization

Hongtu You¹, Guanzhe Wang², Yuchen Liu³, Bo Liu¹, Guangqian Ren¹, Yuhan Tang¹, Feiyang Gao¹, Shuang Liu¹, Linyan Xue¹, Yuefeng Li⁴, Hailiang Dong^{5*}, Guojie Yang^{6*}

¹ College of Quality and Technical Supervision, Hebei University, Baoding 071002, China

² School of Automation Science and Electrical Engineering, Beihang University, Beijing 100000, China

³ International College, Hebei University, Baoding 071002, China

⁴ College of Economics, Hebei University, Baoding 071002, China

⁵ Maritime College, Beibu Gulf University, Qinzhou 535011, China

⁶ Affiliated Hospital of Hebei University, Baoding Key Laboratory of Intelligent Diagnosis of Cardiovascular and Cerebrovascular Diseases, Baoding 071002, China

Corresponding Author Email: donghailiang@bbgu.edu.cn; fly.god@163.com

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ts.430124>

ABSTRACT

Received: 5 August 2025

Revised: 30 November 2025

Accepted: 30 December 2025

Available online: 28 February 2026

Keywords:

tiny objects detection, deep learning, loss function, medical image processing, machine vision

Tiny object detection remains a fundamental challenge in computer vision, primarily due to severe scale imbalance and the dominance of gradients from larger instances during optimization. To address this issue, this paper proposes Tiny Focal Loss (TFL), a scale-aware optimization framework. TFL introduces a continuous penalty mechanism that dynamically allocates weights based on the absolute spatial footprint of instances, thereby prioritizing small-scale targets during backpropagation. To enhance generalization across diverse distributions, TFL incorporates adjustable focusing coefficients and piecewise regularization. The approach is validated on the YOLOv5 architecture using five datasets from various domains, including medical, aerial, industrial, and natural scenes. Experimental results show consistent improvements in precision, recall, and mean average precision (mAP), with gains of up to 5.0% in mAP. These improvements are achieved without additional parameters or computational overhead during inference, making TFL an efficient solution for complex visual recognition tasks.

1. INTRODUCTION

Object detection is an important and fundamental task in the field of computer vision. Tasks such as image segmentation, object tracking, and keypoint detection all rely on the performance of object detection [1, 2]. In object detection tasks, objects with an absolute pixel area smaller than a certain threshold are referred to as small objects. For instance, in the MS COCO dataset [3], objects with dimensions smaller than 32×32 pixels are defined as small objects. These small objects are difficult to detect, yet they have a significant impact on the overall performance of the model.

In practical applications, small target detection holds significant practical importance. For instance, in the field of intelligent driving, distant pedestrians and vehicles are considered small targets [4]. Similarly, in the medical field, minor fractures and cracks that are difficult to diagnose are also small targets [5]. In summary, the development of small target detection holds significant value. Enhancing the accuracy and speed of small target detection will contribute to the advancement of target detection [6].

However, small object detection tasks face several major challenges [7]: (1) limited effective features and shallow depth; (2) high requirements for localization accuracy; (3) issues with

the scale distribution of large and small objects; (4) anchor box assignment problems; (5) object aggregation issues; and (6) network structural constraints." Small object detection tasks require further research from scholars.

To address the scale distribution problem, various data augmentation methods have been proposed. While traditional approaches like copy-pasting provided foundational improvements, more advanced strategies have recently emerged. Li et al. [8] proposed a realistic instance-level data augmentation method based on scene understanding to prevent semantic artifacts and improve small object detection. Furthermore, to overcome degradation during training, Yoon et al. [9] introduced an optimal data augmentation strategy using Fast AutoAugment specifically tailored for small objects, achieving significant performance gains. Addressing the issue of tiny objects having few effective features and shallow depth, recent advancements have moved beyond earlier GAN-based super-resolution methods to more sophisticated architectures. Recent works include ESOD, which efficiently handles high-resolution images to promote small object detection without the prohibitive computational costs of simple image enlargement [10]. Additionally, advanced models like SRM-YOLO have been deployed to enhance feature representation, utilizing Reuse Fusion Structures (RFS) and SPD-Conv to

effectively recover high-frequency details for minute targets [11].

To mitigate the scarcity of effective features and shallow depth, improved strategies for multi-scale learning have been developed. While early methods attempted to fuse deep and shallow features directly, modern architectures have significantly enhanced semantic alignment. For example, Cheng et al. [12] proposed the Contrast-Enhanced Feature Pyramid Network (CE-FPN), which introduces a multi-branch fusion module to emphasize texture boundaries and improve semantic consistency across feature scales. Similarly, Liu et al. [13] introduced HyperFusion-DEIM, a cascaded detection paradigm that utilizes a Multi-Path Attention Network to augment shallow semantic cues and edge-texture sensitivity for small object recognition. Targeting the anchor box allocation problem mentioned earlier, the detection paradigm has shifted heavily toward modern anchor-free and attention-centric mechanisms. While early anchor-free models faced efficiency gaps, recent innovations like Roboflow's RF-DETR have completely eliminated traditional anchor boxes and Non-Maximum Suppression (NMS) overhead, achieving true end-to-end detection [14]. Concurrently, the YOLO lineage has evolved significantly; for instance, YOLOv12 introduces an attention-centric architecture that breaks the dominance of traditional CNN backbones [15], while YOLO26 implements native NMS-free inference and advanced optimizers for highly efficient real-time deployment on edge devices [16].

Focusing on the loss function and targeting the problem of the quantity distribution of tiny and big objects, scholars have effectively improved the detection performance of tiny objects by optimizing the loss function. Beyond early attempts to reweight random errors, the evolution of loss functions has moved toward continuous and scale-aware formulations. Li et al. [17] introduced Generalized Focal Loss (GFL), which generalizes Focal Loss from its discrete form to a continuous version, merging localization quality estimation directly into the class prediction vector. More recently, addressing the extreme scale variations in complex imagery, Li et al. [18] proposed a Scale-Adaptive Loss (SAL) that reshapes vanilla IoU-based losses using logarithmic adjustment factors to dynamically assign lower weights to larger objects, explicitly focusing the training process on tiny objects. However, a critical research gap remains: while existing losses like Focal Loss address class imbalance, they often overlook scale-specific penalties for tiny objects. Application scenarios of tiny object detection, such as industrial flaw detection, intelligent driving assistance, and aerial photography, often require real-time performance and a lightweight nature.

To address this research gap, we propose Tiny Focal Loss (TFL), a scale-aware loss function that dynamically allocates weights based on the absolute pixel area of target instances. Our main contributions are threefold:

(i) **Novel Loss Formulation:** We introduce TFL to mitigate the gradient dominance of large objects, significantly enhancing tiny object detection accuracy without compromising larger targets.

(ii) **Generalizable Strategies:** We design adjustable focusing coefficients and piecewise functions within the TFL framework, allowing flexible adaptation to extreme scale variations across diverse datasets.

(iii) **Cross-Domain Validation:** Extensive evaluations on YOLOv5 across five distinct domains (medical, aerial, industrial, natural) demonstrate consistent mAP improvements with strictly zero additional parameter volume or inference

latency.

2. METHODOLOGY

2.1 Focal loss and area loss

Lin et al. [19] conducted relevant research to enable single-stage detectors to achieve the accuracy of two-stage detectors without compromising original speed. They considered that the imbalance in the number of hard and easy samples is one of the reasons for the low accuracy of single-stage detectors. To address this problem, they proposed Focal loss, and its functional expression is shown as follows:

$$FL(P_t) = -(1 - P_t)^\gamma \log(P_t) \quad (1)$$

where, γ represents the focusing parameter, where $\gamma \geq 0$. Experiments found that $\gamma = 2$ yields better results. The part in Eq. (2) is called the modulating factor.

$$(1 - P_t)^\gamma \quad (2)$$

The modulating factor is inversely proportional to the sample's difficulty. Therefore, if a sample is misclassified, P_t is very small, and $(1 - P_t)$ approaches 1. However, for a sample that is correctly and easily classified, P_t approaches 1, and $(1 - P_t)$ is very small, which results in a smaller weight for this sample when calculating the loss. This reduces the model's attention to easily classified samples. Tiny objects also face the problems of sample imbalance and difficulty imbalance. Therefore, we can learn from the improvement idea of Focal loss and improve the effect of tiny object detection by assigning different weights to different objects in the loss function.

While the original Focal Loss successfully addressed class imbalance, the evolution of dense object detectors has increasingly demanded continuous representations. For instance, recent architectures utilize Khalili and Smyth [20] proposed Powerful-IoU (PIoU) to extend the standard CIoU loss from its conventional form to a quality-aware version, successfully merging localization quality estimation directly into the bounding box regression via a non-monotonic attention mechanism on anchor quality q (0 to 1), while adding a corner-difference penalty to focus on moderate-quality anchors. However, when directly applied to tiny object detection, standard Focal Loss and its modern continuous variants still encounter fundamental bottlenecks. Tiny objects carry extremely limited feature information and are notoriously sensitive to minor localization deviations. Furthermore, standard IoU-based loss functions disproportionately penalize smaller objects with significantly greater regression penalties compared to larger ones during the training phase [18].

To explicitly address this scale imbalance, Wang et al. [21] proposed a weight related to the object area size and applied it to the loss function to improve the detection performance of tiny objects. However, their designed loss function, `Area_loss`, simply establishes a negative correlation between the absolute area of the object and the weight of the loss function. It does not consider the impact brought by many variables such as image size, dataset distribution, and the definition of tiny objects. Thus, there is still room for further improvement in the design of loss functions for tiny objects.

2.2 The proposed Tiny Focal Loss

This paper learns from the "focusing" idea of Focal loss and designs a loss function focusing on tiny objects—TFL—based on the different practical problems to be solved. This function can dynamically allocate the loss function weight for each object according to its size, thereby achieving the effect of improving the model's tiny object detection performance as well as the overall detection performance. The overall formula for the TFL loss function is shown as follows:

$$\omega_i = e^{-k \cdot l} + 1 \quad (3)$$

where, ω_i is the weight allocated by TFL to the i -th object. K is the focusing coefficient, which can adjust the focusing degree of the loss function on tiny objects. Subsequently, the parameter l that describes the size of the object is described as:

$$l = \frac{w \cdot h}{S} \quad (4)$$

where, w is the width of the current object's ground truth, and h is the height of the current object's ground truth; their product represents the absolute pixel area of the current object. S is the dividing line for object size in the current dataset, which is a constant (e.g., in the COCO dataset, S is 32×32). Its functional schematic diagram is shown in Figure 1, where k is 1.6 and S is 32.

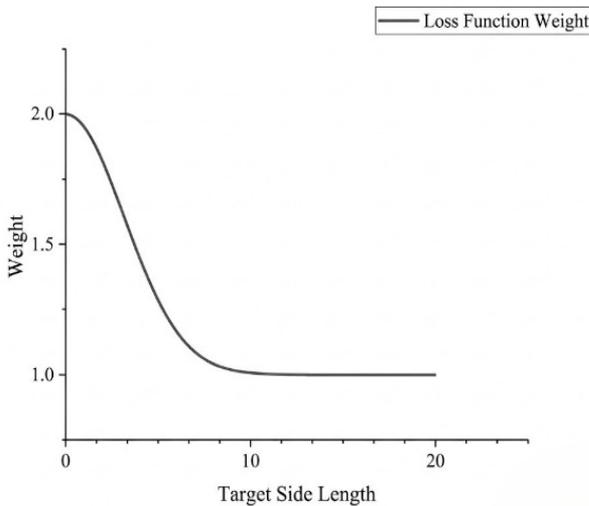


Figure 1. Function image of Tiny Focal Loss

When using this loss function focusing on tiny objects, it will be multiplied directly as a coefficient with the loss function inherently carried by the network model. For example, if the common cross-entropy loss function is $Loss = -\sum_{i=1}^n y_i \log y_i$, the model loss function after applying this loss function becomes $Loss' = -\sum_{i=1}^n \omega_i y_i \log y_i$.

2.3 Adjustment function of Tiny Focal Loss

To improve the generalization of TFL so that it can exert excellent performance on different datasets, this paper also designs two adjustment methods for TFL. The first is the parameter K mentioned in the expression of Eq. (3). K is the focusing coefficient, which can adjust the degree of TFL's focus on tiny objects. K can be any value in the range of $[0, +\infty)$, and the corresponding function curves with different

K values are illustrated in Figure 2.

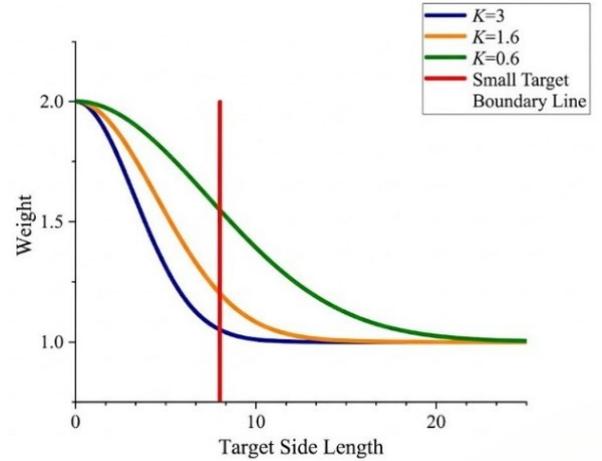


Figure 2. Function images of Tiny Focal Loss with different K values

The second method is that TFL can be set as a piecewise function, meaning TFL is only used for tiny objects, while the weight is set to 1 for non-tiny objects, which is visually depicted in Figure 3.

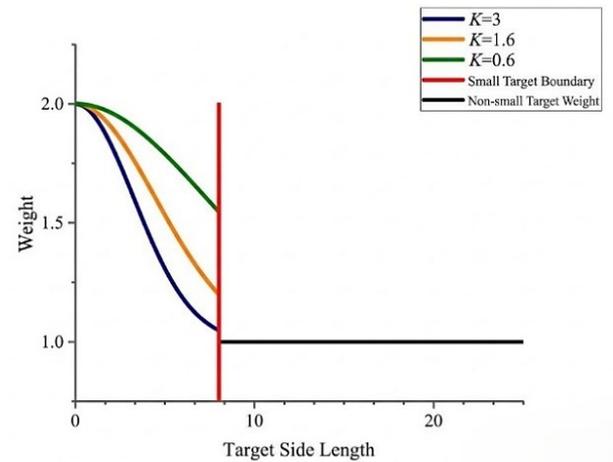


Figure 3. Function image of segmented Tiny Focal Loss with different K values

3. DATASETS

To verify the effectiveness and generalization of the loss function focusing on tiny objects, this paper selects datasets containing multiple image categories and application scenarios.

(i) MSCOCO128 dataset: The MS COCO128 dataset [3] is an ultra-small dataset streamlined and extracted from the MS COCO dataset.

(ii) DOTA dataset: The DOTA dataset [22] contains highly massive aerial images. Aerial images are often applied in multiple fields such as rescue, exploration, and the military, holding significant practical importance.

(iii) MURA dataset: The MURA dataset is an X-ray dataset. [23] This paper only selects the part containing phalangeal and metacarpal fractures. Selecting this dataset can effectively simulate the improvement of the detection performance of tiny

objects in medical images by this loss function.

(iv) PCB defect dataset: The PCB defect dataset [24] can simulate the performance of this loss function in the field of industrial flaw detection.

(v) NWPU VHR-10 dataset: NWPU VHR-10 is a satellite image dataset [25] released by Northwestern Polytechnical University. Satellite images are also frequently used in industrial production and military fields. Improving the detection performance of tiny objects in such images carries important practical significance.

The selection of the above datasets includes both natural and non-natural images, covering multiple fields such as industry, medicine, and the military. Therefore, through comprehensive evaluation on these datasets, the effectiveness and generalization of TFL can be effectively reflected.

4. EXPERIMENTS

4.1 Experimental setup

YOLOv5 offers advantages such as fast detection speed, low memory usage, and quick training. This study uses YOLOv5 as the baseline model to evaluate the proposed method's impact on object detection performance. We note that, to ensure a fair evaluation of TFL's generalization ability across varied domains, this study refrains from using dataset-specific scale thresholds. Instead, the globally unified standard derived from MS COCO is applied universally. Consequently, the threshold parameter S in Eq. (4) is constantly set to 32×32 pixels for all datasets, including COCO128, DOTA, Metacarpal Fracture (MCF), PCB defect, and NWPU VHR-10, throughout the foundational experiments.

To ensure reproducibility, the detailed experimental settings are as follows: the backbone is YOLOv5; the optimizer is SGD with an initial learning rate of 0.01; the batch size is set to 16; and the models are trained for 300 epochs. All experiments were conducted using PyTorch 1.10 on an NVIDIA RTX 3090 GPU. To ensure statistical reliability, all reported metrics are the average of three independent experimental runs.

4.2 Results on COCO128 dataset

The object scale range in the COCO128 dataset is [2, 639] pixels, with a span of 637 pixels, representing a wide object distribution. We compare the baseline YOLOv5 with YOLOv5+TFL to evaluate the detection performance improvements. Table 1 shows the overall detection performance. The evaluation metrics include Precision (P), Recall (R), $mAP@0.5$, and $mAP@0.5:0.95$.

Table 1. Overall detection performance on the COCO128 dataset

Class	P		R		$mAP@.5$		$mAP@.5:.95$	
	w/o	w/	w/o	w/	w/o	w/	w/o	w/
All	0.943	0.953	0.917	0.925	0.966	0.972	0.803	0.822
P-value	<0.05*		>0.05		>0.05		<0.01**	

In the table, "w/" indicates the YOLOv5 network adopting the improved method of the loss function focusing on tiny objects. "w/o" indicates the detection performance of the original YOLOv5 without using TFL. According to the result analysis, introducing TFL into YOLOv5 can effectively improve the overall detection performance. The precision

improved by 1%, the recall rate increased by 0.8%, and $mAP@.5$ and $mAP@.5:.95$ increased by 0.6% and 1.9%, respectively. To more accurately reflect the detection performance improvement of TFL for tiny objects in the COCO128 dataset and its impact on the detection performance of big objects, this paper compiled the respective precision and recall rates for all tiny objects and all big objects, as shown in Table 2.

Table 2. Performance on tiny and big objects in the COCO128 dataset

TFL usage	P		R	
	Tiny objects	Big objects	Tiny objects	Big objects
w/o	0.8498	0.9400	0.8664	0.9823
w/	0.8889	0.9400	0.8762	0.9823

According to the data in the table, it can be found that the detection performance for tiny objects in the COCO128 dataset has improved, while the detection performance for big objects remains unchanged. To intuitively demonstrate the detection performance of TFL for tiny objects, this paper plotted Figure 4 for display. Looking at the comparison images, we first examine the ground truth image. The main difficulties in the image are the three tiny car objects in the distance and two sports balls. In the "Without" image, the two sports balls were not detected, and there is a missed detection of a distant car; furthermore, the confidence of the detected car is not high. Also, an object resembling a streetlamp was mistakenly detected as a tennis racket. However, in the "With" image, all objects that should be detected were identified with relatively high confidence. Nevertheless, the model in the "With" image mistakenly identified a white spot of light in the sky as a sports ball, but it only holds a confidence of 0.27, which is not considered a severe false detection.

Overall, the detection effect in the "With" image is better. The improvement effect of TFL on tiny object detection in the COCO128 dataset is highly intuitive and clear.



Figure 4. Detection results on the COCO128 dataset

5. VALIDATION ON OTHER DATASETS

5.1 Validation on DOTA dataset

The DOTA dataset is highly massive, containing a total of 1,793,658 objects of varying sizes, orientations, and shapes. Its object scale range is also [2,639], making it another dataset with a wide distribution. Table 3 presents the detection performance table for the DOTA dataset.

From the results in the table, the model incorporating TFL showed improvements across all evaluation metrics, with P increasing by 1.9%, R by 1%, $mAP@.5$ by 0.7%, and $mAP@.5:.95$ by 0.4%. It can be seen that in the DOTA dataset, TFL's improvements for P and R are relatively obvious, but the improvement for mAP is somewhat limited. Overall,

however, TFL effectively enhanced the detection performance of YOLOv5 on the DOTA dataset. To accurately observe the impact of TFL on big objects and the improvement in detection performance for tiny objects in the DOTA dataset, this paper also plotted a comparison table of detection effects for tiny and big objects in the DOTA dataset, as shown in Table 4.

Table 3. Overall detection performance on the DOTA dataset

Class	P		R		mAP@.5		mAP@.5:.95	
	w/o	w/	w/o	w/	w/o	w/	w/o	w/
All	0.756	0.77	0.679	0.689	0.691	0.698	0.449	0.453
P-value	<0.05*		>0.05		>0.05		<0.05*	

Table 4. Performance on tiny and big objects in the DOTA dataset

TFL usage	P		R	
	Tiny objects	Big objects	Tiny objects	Big objects
w/o	0.6535	0.8228	0.7020	0.9030
w/	0.6896	0.8518	0.7279	0.8965

As indicated in Table 4, the implementation of TFL yields a substantial impact on the DOTA dataset. In terms of precision, both small and large objects exhibit notable improvements; specifically, the precision for tiny objects increased by 3.61%, and for big objects by 2.90%. Furthermore, the recall rate for tiny objects experienced a significant boost of 2.59%. However, this was accompanied by a marginal decrease of approximately 0.65% in the recall rate for big objects.

This slight decline in large object recall can be attributed to the presence of a sparse number of ultra-large objects distributed within the [320, 639] scale interval in the DOTA dataset. Because TFL dynamically scales down the loss weights for instances with massive pixel areas, the model tends to somewhat deprioritize these ultra-large targets during training, leading to this minor trade-off. Nevertheless, taken comprehensively, TFL effectively elevates both the overall and tiny object detection performance on the DOTA dataset. To intuitively demonstrate these enhancements, visual comparisons of the detection results are presented in Figure 5.

Figure 5 displays the comparison images from the DOTA dataset. In these images, "w/o" represents the original YOLOv5 network, and "w/" represents the YOLOv5 network utilizing TFL. The ground truth represents the actual annotated results. This image depicts an aerial view of a parking lot,

where the vehicles are small and dense, making them difficult to detect accurately. Furthermore, there is a large number of container-like confounding objects on the left side of the image.

Therefore, this image poses considerable detection difficulty. To make the image more intuitive, this paper plotted a difference image. The difference image highlights the objects detected by the TFL network but missed by the original network. Thus, the difference image reflects the enhancement effect of TFL on the YOLOv5 network regarding the DOTA dataset. Based on the difference image, it can be observed that the YOLOv5 network using TFL detected approximately 40 more objects, indicating a very obvious improvement.

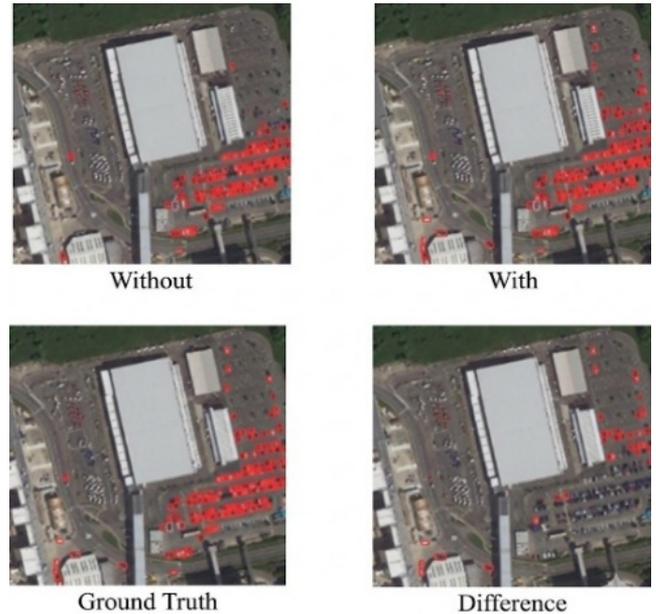


Figure 5. Detection results on the DOTA dataset

5.2 Validation on MCF dataset

The MURA dataset is an upper extremity musculoskeletal X-ray dataset. This paper will solely select the images of phalangeal and metacarpal bones to form a dataset (MCF) for testing TFL's improvement on tiny object detection performance in the medical field. The MCF dataset only contains one object category: fractures. Its scale range is [8,169]. Since the MCF dataset does not specify a predefined test set and validation set, this paper employs 5-fold cross-validation. Table 5 presents the cross-validation results for the MCF dataset.

Table 5. Overall detection performance with 5-fold cross-validation on the MCF dataset

Class	P		R		mAP@.5		mAP@.5:.95	
	w/o	w/	w/o	w/	w/o	w/	w/o	w/
AVG	0.727±0.047	0.759±0.033	0.546±0.021	0.567±0.016	0.604±0.011	0.608±0.017	0.273±0.013	0.28±0.014
P-value	>0.05		<0.01**		>0.05		<0.01**	

Table 6. Performance on tiny and big objects in the MCF dataset

TFL Usage	P		R	
	Tiny Objects	Big Objects	Tiny Objects	Big Objects
w/o	0.6693	0.7627	0.5921	0.6522
w/	0.7059	0.7649	0.6059	0.6517

"AVG" indicates the average results of the 5 folds, and "P-value" indicates the significant difference across the 5 folds. Bold font indicates superior performance after using TFL. In the AVG row of the table, it can be seen that all four evaluation metrics improved after using TFL, and the results for R and mAP@.5:.95 exhibit significant differences. The precision improved by 3.2%, and the recall rate improved by 2.1%, showing an obvious enhancement. Table 6 provides the comparison table for tiny/big object detection effects on the MCF dataset.

In the table, bold font indicates the superior result. Based on the position of the bold fonts, it is apparent that, similar to the situation with the DOTA dataset, the P for both big and tiny objects improved after using TFL; the R for tiny objects also improved, but the R for big objects decreased. However, in the MCF dataset, the R for big objects only decreased by 0.05%. Taken comprehensively, TFL has an effective improvement effect on the MCF dataset, which features a smaller object scale range and a single category of objects.

5.3 Validation on PCB defect dataset

The PCB defect dataset is a synthetic dataset published by Peking University. Its primary detection targets are 6 common defects on PCB circuit boards. Its object scale range is [7,58], which is a very small interval compared to other datasets. Table 7 presents the overall detection performance table for the PCB defect dataset.

Table 7. Overall detection performance on the PCB dataset

Class	P		R		mAP@.5		mAP@.5:.95	
	w/o	w/	w/o	w/	w/o	w/	w/o	w/
All	0.980	0.980	0.971	0.969	0.982	0.981	0.579	0.599
P-value	>0.05		< 0.01**		>0.05		< 0.01**	

The structure of this table aligns with that of the DOTA dataset overall detection performance table. "All" represents the overall detection performance of all categories. "P-value" indicates whether the evaluation metric shows a significant difference. Bold font indicates data where "With" outperforms "Without". According to the table, except for mAP@.5:.95, other evaluation metrics showed no improvement, with even a slight drop of 0.1% to 0.2%. Judging from the overall detection performance table, TFL's improvement on the PCB defect dataset is not outstanding. However, for a comprehensive analysis, it should be combined with the tiny/big object

comparison table for the PCB defect dataset. This comparison table is shown in Table 8.

Table 8. Performance on tiny and big objects in the PCB dataset

TFL usage	P		R	
	Tiny objects	Big objects	Tiny objects	Big objects
w/o	0.9583	0.9964	0.9633	0.9897
w/	0.9600	1.0000	0.9654	0.9883

Looking at the two tables combined, the impact of TFL on the detection performance of the PCB defect dataset, whether positive or negative, is minimal. This paper analyzes that this is because all objects in the PCB defect dataset itself are relatively small, and the scale range is narrow. Thus, the weight allocation does not excessively affect the distribution of training resources among the objects. In summary, TFL yields a certain, albeit non-obvious, improvement effect on datasets like the PCB defect dataset, which are dominated by tiny objects, have small object scale ranges, and feature concentrated object distributions.

5.4 Validation on NWPU VHR-10 dataset

The NWPU VHR-10 dataset (hereinafter referred to as the NWPU dataset) is a satellite image dataset, and its images and objects are similar to the DOTA aerial image dataset. Satellite images have wide applications in industrial production, daily life, and military fields [21]. Studying target detection tasks on this dataset has significant real-world meaning. Unlike other datasets, this paper will compare the performance on the NWPU dataset with other methods mentioned in literature. The NWPU dataset does not predefine test and training sets, requiring researchers to manually divide them and conduct multi-fold cross-validation to ensure the reliability of experimental results. There are two common partition methods for the NWPU dataset: 60% training set, 20% test set, 20% validation set; and 20% training set, 20% test set, 60% validation set. Both methods are widely used, but scholars must clarify the chosen method in their articles to ensure data comparability. This paper will reference the NWPU dataset results table compiled in the literature by Wang et al. [21], adopting the "60% training set, 20% test set, 20% validation set" partition method for training, consistent with the referenced literature. Table 9 presents the overall performance comparison table for the NWPU dataset.

Table 9. Comparison results based on NWPU dataset

Method	Plane	SH	ST	BD	TC	BC	GTF	Harbor	Bridge	Vehicle	mAP↑
Transferred CNN	66.1	56.9	84.3	81.6	35	45.9	80	62	42.9	42.9	59.7
RICNN	5	77.34	85.27	88.12	40.83	58.45	86.73	68.6	61.51	71.1	72.63
R-P-Faster R-CNN	90.4	75	44.4	89.9	79	77.6	87.7	79.1	68.2	73.2	76.5
SSD512	90.4	60.9	79.8	89.9	82.6	80.6	98.3	73.4	76.7	52.1	78.4
DSSD321	86.5	65.4	90.3	89.6	85.1	80.4	78.2	70.5	68.2	74.2	78.8
DSOD300	82.7	62.8	89.2	90.1	87.8	80.9	79.8	82.1	81.2	61.3	79.8
R-FCN	81.7	80.6	66.2	90.3	80.2	69.7	89.8	78.6	47.8	78.3	76.3
Deformable R-FCN	87.3	81.4	63.6	90.4	81.6	74.1	90.3	75.3	71.4	75.5	79.1
Faster R-CNN	94.6	82.3	65.32	95.5	81.9	89.7	92.4	72.4	57.5	77.8	80.9
Deformable Faster R-CNN	90.7	87.1	70.5	89.5	89.3	87.3	87.2	73.5	69.9	88.8	84.4
RDAS512	99.6	85.5	89	95	89.6	94.8	95.3	82.6	77.2	86.5	89.5
Multi-Scale CNN	99.3	92	83.2	97.2	90.8	92.6	98.1	85.1	71.9	85.9	89.6
FMSSD	99.7	89.9	90.3	98.2	86	96.8	99.6	75.6	80.1	88.2	90.4
YOLOV5-ORI	99.6	94.4	93.6	97.5	96.2	99.9	99.6	93.2	95	93.8	93.3
YOLOV5-TFL	99.8	97.4	98.7	97.4	95	1	97.1	90	95.1	92.3	95.4

This paper adopts the same table layout as literature [3], where the "Method" column indicates the used method or network model, columns from "Plane" to "Vehicle" display the detection performance for each object category, and the "mAP" column indicates the overall mAP of the network. In the table, YOLOV5-ORI represents the original YOLOv5 network, and YOLOV5-TFL represents the YOLOv5 network enhanced with TFL. Bold font in the table highlights the optimal value in that column. According to the data and analysis in the table, owing to YOLOv5's inherent excellent performance, the mAP of its original network already outperforms other networks. The YOLOv5-TFL, improved by TFL, further increased by 2.1% relative to the original network. This demonstrates that TFL's improvement on the model's overall detection performance is both effective and significant.

5.5 Exploration of Tiny Focal Loss adjustment function

As mentioned earlier, TFL defines two adjustment functions, namely the focusing coefficient K and the piecewise function. To verify the effect of the two adjustment functions, this paper conducted a total of 30 sets of experiments for six preset TFL forms across the five datasets mentioned above, exploring the impact of different TFL settings on various datasets. This seeks to establish a correlation between the TFL parameter settings and the dataset's object scale alongside its weight distribution histogram, thereby discovering empirical rules for optimal TFL settings and providing methodological guidance for utilizing TFL. Previous experiments showed that TFL's effect on datasets with small object scale ranges is less obvious. For instance, the MCF dataset has an object scale range of [8,169], and the PCB dataset has a range of [7,58]. Under default settings, TFL did not demonstrate adequately outstanding performance on these two datasets. This paper utilized TFL's adjustment function to improve TFL's performance on such datasets. Table 10 and Table 11 present the comparison tables after applying the adjustment functions to the MCF and PCB datasets, respectively.

Table 10. Comparison results of adjustment function based on MCF dataset

TFL Status	P		R	
	Tiny objects	Big objects	Tiny objects	Big objects
w/o	0.669	0.763	0.592	0.652
K=1.6	0.706	0.765	0.609	0.652
K=3	0.734	0.736	0.669	0.698

Table 11. Comparison results of adjustment function based on PCB dataset

TFL Status	P		R	
	Tiny objects	Big objects	Tiny objects	Big objects
w/o	0.958	0.996	0.963	0.988
K=1.6	0.960	1.000	0.965	0.988
K=0.6	0.972	0.999	0.966	0.991

According to the result analysis, the adjustment function yields obvious improvement effects on datasets with smaller object scale ranges, like PCB and MCF, effectively increasing TFL's generalization capability across different datasets.

Based on the outcomes of the 30 experimental sets, this paper can provide the following empirical rules for using TFL adjustment functions: For the vast majority of datasets, adopting the default setting a non-piecewise function with $K = 1.6$ can achieve satisfactory performance improvements. For datasets with small object scale ranges, specifically those whose scale range length is less than one-fourth of the image size. For example, if YOLOv5's default image size is 640×640 pixels, a dataset with a scale range length of less than 160 pixels falls into this category. The piecewise function form of TFL should be selected. Concurrently, the smaller the scale range length, the more radical the TFL should be, implying that the K value should be smaller. This paper analyzes that the cause of this pattern is that in datasets with small scale range lengths, the discrepancy between big and tiny objects is not conspicuous. Therefore, a piecewise function form is necessary to widen the gap in weights assigned to big and tiny objects. Simultaneously, a smaller scale range length demands further widening of this gap, necessitating a smaller K value.

5.6 Impact of TFL on model volume and inference speed

Starting from the practical application needs of tiny object detection, this paper attempts to improve the tiny object detection performance and overall detection performance of the model without increasing model complexity or slowing down model inference speed. To verify this perspective, this paper conducted a comparative analysis of the volume and speed of the models trained on the five previously mentioned datasets. The results are shown in Table 12.

Table 12. Comparison results of model volume and inference speed

Dataset	Tiny Focal Loss (TFL)	Weight file size (MB)	Preprocessing time (ms)	Inference time (ms)
COCO	w/o	14.6	0.4	14.5
	w/	14.6	0.5	14.4
DOTA	w/o	13.9	0.5	12.7
	w/	13.9	0.5	12.4
MCF	w/o	14.0	0.4	14.5
	w/	14.0	0.4	14.6
PCB	w/o	14.2	0.4	13.7
	w/	14.2	0.5	13.5
NWPU	w/o	15.0	0.4	15.6
	w/	15.0	0.6	15.5

According to the results in the table, it can be observed that the size of the model weight files improved by the method proposed in this paper underwent no changes. This indicates that the proposed method does not affect the model's complexity, making it conducive to model lightweighting. Moreover, after utilizing this method, the model's preprocessing time and inference time fluctuated within ± 0.2 milliseconds, illustrating that this method has virtually no impact on the model's preprocessing and inference speeds. Therefore, the TFL proposed in this paper is an excellent method for enhancing target detection performance with almost no negative effects, making it highly applicable for scenarios demanding lightweight execution and real-time capability in tiny object detection.

6. CONCLUSIONS

Starting from the target detection task and focusing on the challenge of tiny object detection, this paper considered practical production and living needs. A loss function focusing on tiny objects was designed, successfully enhancing tiny object detection performance without impairing big object detection performance, thereby elevating the model's overall detection performance. Additionally, through further research, this paper adjusted the parameter settings and functional segmentation of TFL, subsequently raising TFL's performance on datasets with small-scale ranges, which proved the high generalization of TFL. Regarding future improvements on loss functions, closer ties should be maintained with the actual conditions of datasets to maximize the loss function's adaptability to diverse datasets. While tiny object detection undoubtedly remains a challenging aspect of the target detection task, utilizing methods such as super-resolution or even hardware performance upgrades as an "indirect approach"—rendering the tiny objects no longer "tiny"—might present a more viable improvement methodology to resolve tiny object detection challenges and advance target detection tasks.

ACKNOWLEDGMENT

This study was financially supported by the Postgraduate Innovation Funding Project of Hebei Province (Grant No.: CXZZSS2026004); the College Students' Innovation and Entrepreneurship Training Program of Hebei University (Grant No.: XJLX252514); the Guangxi Higher Education Undergraduate Teaching Reform Project in 2024 (Grant No.: 2024JGB275); the Guangxi Natural Science Foundation in 2026 (Grant No.: 2025JJH160108); the Qinzhou Scientific Research and Technology Development Plan Project in 2023 (Grant No.: 20233141); and the Qinzhou Scientific Research and Technology Development Plan Project in 2025 (Grant No.: 20251706).

DATA AVAILABILITY STATEMENT

The publicly available datasets used in this study (MS COCO128, DOTA, MURA, PCB defect dataset, and NWPU VHR-10) can be found at their respective official open-source repositories.

REFERENCES

- [1] Nikouei, M., Baroutian, B., Nabavi, S., Taraghi, F., Aghaei, A., Sajedi, A., Moghaddam, M.E. (2025). Small object detection: A comprehensive survey on challenges, techniques and real-world applications. *Intelligent Systems with Applications*, 27: 200561. <https://doi.org/10.1016/j.iswa.2025.200561>
- [2] Aldubaikhi, A., Patel, S. (2025). Advancements in small-object detection (2023–2025): Approaches, datasets, benchmarks, applications, and practical guidance. *Applied Sciences*, 15(22): 11882. <https://doi.org/10.3390/app152211882>
- [3] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740-755. https://doi.org/10.1007/978-3-319-10602-1_48
- [4] Tian, S., Zhao, K., Song, L. (2025). Research on small target detection algorithm for autonomous vehicle scenarios. *Journal of Advanced Transportation*, 2025(1): 8452511. <https://doi.org/10.1155/atr/8452511>
- [5] Yeung, M., Sala, E., Schönlieb, C.B., Rundo, L. (2022). Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, 95: 102026. <https://doi.org/10.1016/j.compmedimag.2021.102026>
- [6] Hua, W., Chen, Q. (2025). A survey of small object detection based on deep learning in aerial images. *Artificial Intelligence Review*, 58(6): 162. <https://doi.org/10.1007/s10462-025-11150-9>
- [7] Shi, Y., Li, J., Jia, Y., Hong, Q. (2026). LDA-DETR: A lightweight dynamic attention-enhanced DETR for small object detection. *PloS one*, 21(1): e0340977. <https://doi.org/10.1371/journal.pone.0340977>
- [8] Li, C., Zhang, Z., Zhong, P., He, J. (2026). A realistic instance-level data augmentation method for small-object detection based on scene understanding. *Remote Sensing*, 18(4): 647. <https://doi.org/10.3390/rs18040647>
- [9] Yoon, D., Kim, S., Yoo, S., Lee, J. (2025). Data augmentation for small object using fast autoaugment. *arXiv preprint arXiv:2506.08956*. <https://doi.org/10.48550/arXiv.2506.08956>
- [10] Liu, K., Fu, Z., Jin, S., Chen, Z., Zhou, F., Jiang, R., Ye, J. (2024). ESOD: Efficient small object detection on high-resolution images. *IEEE Transactions on Image Processing*, 34: 183-195. <https://doi.org/10.1109/TIP.2024.3501853>
- [11] Yao, B., Zhang, C., Meng, Q., Sun, X., Hu, X., Wang, L., Li, X. (2025). SRM-YOLO for small object detection in remote sensing images. *Remote Sensing*, 17(12): 2099. <https://doi.org/10.3390/rs17122099>
- [12] Cheng, Q., Cai, Z., Lin, Y., Li, J., Lan, T. (2025). CE-FPN-YOLO: A contrast-enhanced feature pyramid for detecting concealed small objects in X-ray baggage images. *Mathematics*, 13(24): 4012. <https://doi.org/10.3390/math13244012>
- [13] Liu, J., Tao, J., Liu, X., Ma, J., Guo, C., Dong, C., Shi, P. (2025). Multi path attention and scale aware fusion for accurate object detection in remote sensing imagery. *Scientific Reports*, 15(1): 41810. <https://doi.org/10.1038/s41598-025-25900-w>
- [14] Robinson, I., Robicheaux, P., Popov, M., Ramanan, D., Peri, N. (2025). RF-DETR: Neural architecture search for real-time detection transformers. *arXiv preprint arXiv:2511.09554*. <https://doi.org/10.48550/arXiv.2511.09554>
- [15] Tian, Y., Ye, Q., Doermann, D. (2025). Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*. <https://doi.org/10.48550/arXiv.2502.12524>
- [16] Sapkota, R., Cheppally, R.H., Sharda, A., Karkee, M. (2025). YOLO26: Key architectural enhancements and performance benchmarking for real-time object detection. *arXiv preprint arXiv:2509.25164*. <https://doi.org/10.48550/arXiv.2509.25164>
- [17] Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., Yang, J. (2020). Generalized focal loss: Learning

- qualified and distributed bounding boxes for dense object detection. *Advances in Neural Information Processing Systems*, 33: 21002-21012.
- [18] Li, J., Huang, Y., Song, H., Wang, T., Xia, J., Lin, Y., Yang, J. (2025). Scale-aware relay and scale-adaptive loss for tiny object detection in aerial images. *arXiv preprint* [arXiv:2511.09891](https://doi.org/10.48550/arXiv.2511.09891). <https://doi.org/10.48550/arXiv.2511.09891>
- [19] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980-2988.
- [20] Khalili, B., Smyth, A.W. (2024). SOD-YOLOv8—Enhancing YOLOv8 for small object detection in aerial imagery and traffic scenes. *Sensors*, 24(19): 6209. <https://doi.org/10.3390/s24196209>
- [21] Wang, P., Sun, X., Diao, W., Fu, K. (2019). FMSSD: Feature-merged single-shot detection for multiscale objects in large-scale remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5): 3377-3390. <https://doi.org/10.1109/TGRS.2019.2954328>
- [22] Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L. (2018). DOTA: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3974-3983.
- [23] Rajpurkar, P., Irvin, J., Bagul, A., Ding, D., Duan, T., Mehta, H., Yang, B., Zhu, K., Laird, D., Ball, R.L., Langlotz, C., Shpanskaya, K., Lungren, M. P., Ng, A. Y. (2017). MURA: Large dataset for abnormality detection in musculoskeletal radiographs. *arXiv preprint* [arXiv:1712.06957](https://doi.org/10.48550/arXiv.1712.06957). <https://doi.org/10.48550/arXiv.1712.06957>
- [24] Huang, W., Wei, P. (2019). A PCB dataset for defects detection and classification. *arXiv preprint* [arXiv:1901.08204](https://doi.org/10.48550/arXiv.1901.08204). <https://doi.org/10.48550/arXiv.1901.08204>
- [25] Cheng, G., Han, J., Zhou, P., Guo, L. (2014). Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS Journal of Photogrammetry and Remote Sensing*, 98: 119-132. <https://doi.org/10.1016/j.isprsjprs.2014.10.002>