GCA-YOLO: An Edge-optimized Traffic Sign Detection Model

Peiyan Yuan[1], Yifan Pei[1], Chenyang Wang[2,3,*], Xiaoyan Zhao[1], Zhou Liu[3], Xiaoqiang Zhu[4], and Tarik Taleb[5]
[1]College of Computer and Information Engineering, Henan Normal University, Xinxiang, China
[2]College of Computer Science & Software Engineering, Shenzhen University, Shenzhen, China
[3]Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China
[4]School of Cyberspace Science and Technology, Beijing Jiaotong University, Beijing, China
[5]Faculty of Electrical Engineering and Information Technology, Ruhr University Bochum, Bochum, Germany
Email: {peiyan,zhaoxiaoyan}@htu.cn, pyff0812@163.com, chenyangwang@ieee.org,
liuzhou@gml.ac.cn, xqzhu@bjtu.edu.cn, tarik.taleb@rub.de

***Abstract*—To address the challenges of small target features being less prominent, susceptibility to background interference, and sample imbalance in road traffic sign detection, which leads to insufficient model detection accuracy, as well as the high complexity of current object detection models that struggle to operate efficiently on resource-constrained edge devices, we propose a traffic sign detection model based on GCA-YOLO. By adding small target detection layers and removing large target detection layers, the model enhances its small target detection capabilities and reduces its parameter size. The introduction of the T-BiFPN (Tiny-BiFPN) structure improves multi-scale feature fusion, while the C2f-CP module increases computational efficiency on edge devices. The GCA (Global Coordinate Attention) mechanism enhances feature extraction, and the Focaler-CIoU loss function enables the model to focus more on difficult samples and accelerate the convergence of bounding boxes. Experimental results show the superiorities of the proposed GCA-YOLO that compared to YOLOv8n, GCA-YOLO improves precision, recall, mAP@50, and mAP@50:95 by 8.6%, 6.1%, 8.7%, and 6.2%, respectively, while reducing the model's parameter count and size by 38.57% and 33.21%, respectively.**

***Index Terms*—Traffic Sign Detection, Multi-Scale Feature Fusion, Edge Computing, Attention Mechanism, YOLOv8**

I. Introduction

The rapid development of intelligent transportation systems and advanced driver assistance systems has highlighted the growing importance of addressing traffic management and safety challenges. Traffic signs are essential for road traffic regulation, and their accurate real-time detection enhances the safety and reliability of autonomous driving systems while supporting traffic data analysis and management [1]–[3]. However, traffic signs often occupy fewer pixels in images due to their placement on roadsides or overhead and the distance from devices like surveillance cameras or dashcams. Furthermore, complex road environments with background interferences, such as vehicles, buildings, and trees, exacerbate detection challenges. Issues like small target size, background interference, and sample imbalance demand higher accuracy in traffic sign detection [4].

*Corresponding author: Chenyang Wang.

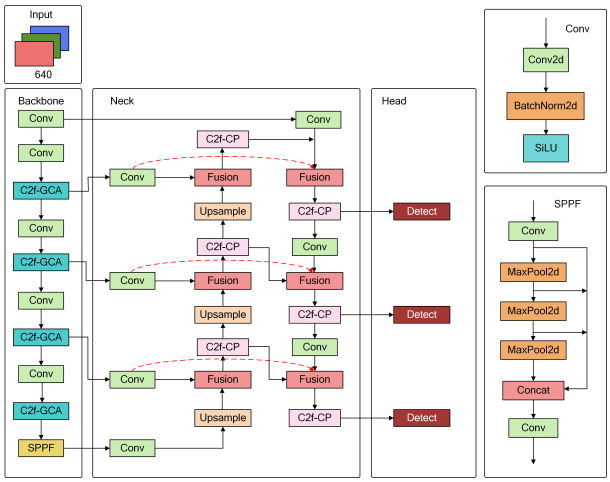

Fig. 1. GCA-YOLO network model structure diagram

Traditional object detection algorithms, which rely on handcrafted features and classifiers, face significant limitations in complex scenarios involving occlusion, varying angles, and lighting changes [5]. In contrast, deep learning-based object detection algorithms have been widely adopted across various fields and are generally categorized into single-stage and two-stage methods. Two-stage algorithms, such as R-CNN [6], Faster R-CNN [7], and D2Det [8], first generate candidate regions and then classify them, achieving high accuracy but suffering from slower detection speeds due to their complex structures, limiting real-time applicability. Single-stage algorithms, including SSD [9], RetinaNet [10], and YOLO [11], directly predict object locations and categories using a single neural network, offering faster speeds and better suitability for practical scenarios. Recently, Transformer-based models [12], leveraging self-attention mechanisms, have shown potential in object detection but require significant computational resources due to their high parameter count and complexity.

In intelligent transportation and driving systems, detection tasks often rely on edge devices for local processing, reduc-

ing dependence on cloud servers while mitigating bandwidth limitations and transmission latency, thereby improving system responsiveness and reliability [13], [14]. However, the limited storage capacity of edge devices poses challenges for deploying complex object detection models, despite their high accuracy [15]–[17]. To address this, this paper adopts the single-stage YOLOv8n model as the benchmark, known for its fast detection speed and simple structure [18]. Nevertheless, directly applying YOLOv8n to traffic sign detection reveals challenges such as insufficient accuracy and relatively large model size, limiting its practical applicability.

To address the constraints of model parameters and size on edge devices while enhancing the accuracy of traffic sign detection, this paper proposes a GCA-YOLO based traffic sign detection model, with its architecture illustrated in Fig. 1. The main contributions of this paper are as follows:

- We propose GCA-YOLO, a novel traffic sign detection model specifically designed to address the challenges of small target detection, background interference, and sample imbalance in traffic scenarios. The model is optimized for deployment on resource-constrained edge devices by balancing detection accuracy and computational efficiency.
- To enhance small target detection, the proposed model incorporates a specialized design by adding small target detection layers while removing large target detection layers. Additionally, the T-BiFPN structure is introduced to improve multi-scale feature fusion, effectively capturing detailed features across scales.
- Besides, the model integrates the C2f-CP module with Partial Convolution (PConv) to reduce computation and memory usage, the GCA mechanism to enhance feature extraction by focusing on key regions, and the Focaler-CIoU loss function to refine bounding box regression by prioritizing hard samples and accelerating convergence.

II. Related Work

The YOLOv8 algorithm is available in variants of different sizes: YOLOv8x, YOLOv8l, YOLOv8m, YOLOv8s, and YOLOv8n.The architecture of this model consists of four main components: Input, Backbone, Neck, and Head. The Backbone is used to extract features from the image. The neck adopts a PANet structure to fuse multi-scale features and enhance information flow, leading to improved detection accuracy. The head utilizes a decoupled design and an anchor-free mechanism for efficient object classification and localization.

Single-stage and two-stage object detection algorithms are widely applied in traffic sign detection. Zhang et al. [19] proposed a cascaded R-CNN structure with a multi-scale attention mechanism using dot product and softmax weighting to enhance traffic sign features and detection accuracy. Han et al. [20] developed YOLO-SG, integrating the SPD-Conv down-sampling structure and GhostNet for improved small traffic sign detection in complex scenarios. Xiong et al. [21] introduced Ghost-YOLOv8, integrating the GAM attention mechanism, GIoU loss function, and a C2fGhost module to improve feature extraction while reducing model parameters. Chen et al. [22] proposed a semi-supervised framework combining CNN and Transformer encoder-decoder structures. Using a hierarchical sampling method (HSM) and a local-global information aggregator (LGIA), their framework enhances feature representation by fusing local and global traffic sign features.

III. Improved GCA-YOLO Algorithm

A. Small Object Detection Layer

In practical scenarios, recognizing small-sized traffic signs from long distances is often critical. However, the YOLOv8n network utilizes detection layers with scales of 20×20, 40×40, and 80×80. For small objects, lower-resolution feature maps struggle to capture fine details and edge information, leading to challenges in accurate detection and localization. In contrast, high-resolution feature maps better preserve the detailed features of small objects. To address this, a 160×160 small object detection layer is added, enhancing the model's capability to capture and recognize small objects, while the 20×20 large object detection layer is removed to reduce model complexity and parameter count.

B. T-BiFPN Multi-scale Feature Fusion Structure

The Neck part of the YOLOv8 uses the PANet structure as shown in Fig. 2(a). However, for small traffic sign targets, although PANet enhances feature fusion at different levels through top-down and bottom-up paths, the small size of the targets can lead to information loss and inadequate feature representation. Therefore, based on the added small object detection layer and BiFPN [23], a T-BiFPN feature pyramid structure is designed to effectively fuse multi-scale features while reducing the model's parameter count.

Bidirectional Feature Pyramid Network (BiFPN) structure enables bidirectional cross-level connections between feature maps at different scales, allowing information to flow both bottom-up and top-down within the network. At the same time, it performs feature fusion in the horizontal direction, thereby achieving more effective multi-scale information integration. Its structure is shown in Fig. 2(b). Meanwhile, BiFPN employs a weighted feature fusion mechanism, *i.e.,*:

$$O = \sum_i \frac{\omega_i}{\varepsilon + \sum_j \omega_j} \times I_i \tag{1}$$

where ω is the weight of the input, $\varepsilon = 0.0001$ is used to avoid numerical instability, and I_i denotes the input features.

Although BiFPN effectively fuses multi-scale features, its lack of shallow feature fusion can result in the loss of high-resolution details, reducing accuracy for small targets. To address this, the T-BiFPN structure is proposed, as shown in Fig. 2(c), based on the modified network with a small target detection layer. To reduce model parameters, unnecessary feature fusion connections are removed, retaining only the P2, P3, and P4 feature maps for output. However, the P5 layer is preserved to extract abstract global semantic features, which

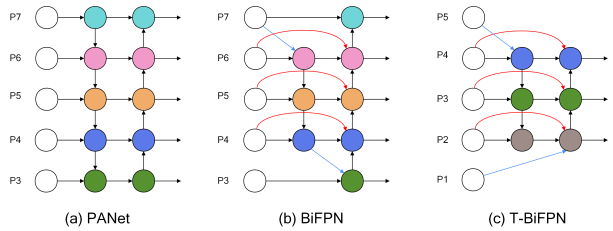

Fig. 2. PANet, BiFPN, and T-BiFPN feature pyramid structures

are fused with the P4 channel to enhance representation. Additionally, the high-resolution feature maps of the P1 layer are fused with the P2 channel, improving detection performance for small traffic signs in complex scenarios.

C. C2f-CP Module

To reduce network parameter count and mitigate high similarity and feature redundancy among channels caused by excessive stacking of 3×3 convolution operations in the Bottleneck structure, the C2f-CP module is proposed. This module replaces one of the 3×3 standard convolutions in the original Bottleneck with a Partial Convolution (PConv) from FasterNet [24], achieving a balance between parameter efficiency and detection performance. The structure of the C2f-CP module is illustrated in Fig. 3.

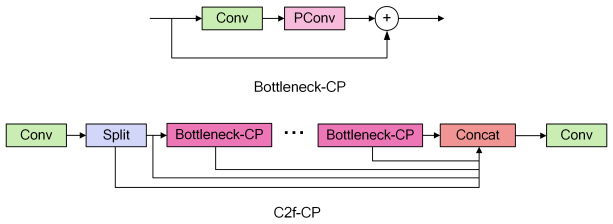

Fig. 3. C2f-CP module structure diagram

The PConv in the C2f-CP module retains the benefits of DWConv while mitigating the drawback of increased memory access frequency. Leveraging the high similarity of feature maps across channels, PConv performs convolution operations on a subset of input feature map channels while leaving the remaining channels unchanged.

PConv not only significantly reduces memory access volume but also effectively decreases redundant computations, thereby enhancing overall computational efficiency. The computational complexity of standard convolution and PConv is:

$$\begin{aligned} \mathrm{FLOPs}_{Conv} &= h \times w \times k^2 \times c^2 \\ \mathrm{FLOPs}_{PConv} &= h \times w \times k^2 \times {c_p}^2 \end{aligned} \tag{2}$$

where h and w are the height and width of the feature map, k is the size of the convolution kernel, c is the number of channels for the regular convolution operation, and c_p is the number of channels for the PConv operation. When $c_p = \frac{c}{4}$, the FLOPs of PConv are $\frac{1}{16}$ of those of a standard convolution.

Although PConv preserves substantial information from the original channels, it may struggle with capturing complex spatial feature variations. To address this, an additional 3×3 standard convolution layer is retained, creating a complementary structural design that significantly enhances the network's capability to capture spatial features.

D. GCA Attention Mechanism

The GCA attention mechanism, inspired by Su et al. [25], is designed to enhance feature extraction for small targets and improve detection performance, as illustrated in Fig. 4. It begins by applying average pooling and max pooling along the height and width of the input feature map X, preserving spatial directional information by combining global average and prominent feature details. The resulting pooled maps are stacked along their respective directions to generate new feature maps X_h and X_w, which are then transformed and fused to produce the enriched contextual feature map Y. To capture directional information more effectively and reduce model complexity, 1×2 and 2×1 convolutional kernels, along with a decay rate r, are employed to reduce dimensionality, resulting in the refined feature map $Y^{'}$.

$$\begin{aligned} \mathrm{X}_h &= c\left(Avg_h\left(X\right), Max_h\left(X\right)\right) \\ \mathrm{X}_w &= c\left(Avg_w\left(X\right), Max_w\left(X\right)\right) \\ Y &= c\left(X_h, X_w\right) \\ Y^{'} &= F_1\left(Y\right) \end{aligned} \tag{3}$$

where Avg_h and Max_h denote average and max pooling operations along the height direction, respectively, while Avg_w and Max_w represent the corresponding operations along the width direction. The operator c indicates concatenation along specified dimensions, and F_1 refers to convolutional kernels of size 1×2 or 2×1.

The segmented feature maps are then processed using a 1×1 convolution to enable cross-dimensional interaction between features from two directions. This design ensures lightweight efficiency, preserves the independence of directional information, and effectively captures long-range dependencies. Finally, the feature weights for both directions are computed using a sigmoid function and multiplied with the original feature layer, producing the final output feature layer Z.

$$\begin{aligned} Y_h^{'}, Y_w^{'} &= f_{1\times1}\left(Y^{'}{}_{split}\right) \\ Y_h^{''}, Y_w^{''} &= F_3\left(\delta\left(F_2\left(Y_h^{'}, Y_w^{'}\right)\right)\right) \\ Z &= X * \sigma\left(Y_h^{''}\right) * \sigma\left(Y_w^{''}\right) \end{aligned} \tag{4}$$

where $split$ denotes the segmentation operation, $f_{1\times1}$ represents the 1×1 convolution operation, and F_2 and F_3 are $1 \times 1 \times C \times C/r$ and $1 \times 1 \times C/r \times C$ convolution kernels, respectively. δ represents the ReLU function, and σ denotes the sigmoid function. The GCA attention mechanism enhances feature representation by capturing complex spatial and channel dependencies through multi-level processing and information fusion, thereby improving the discriminative capability of the features.The lightweight design of the GCA

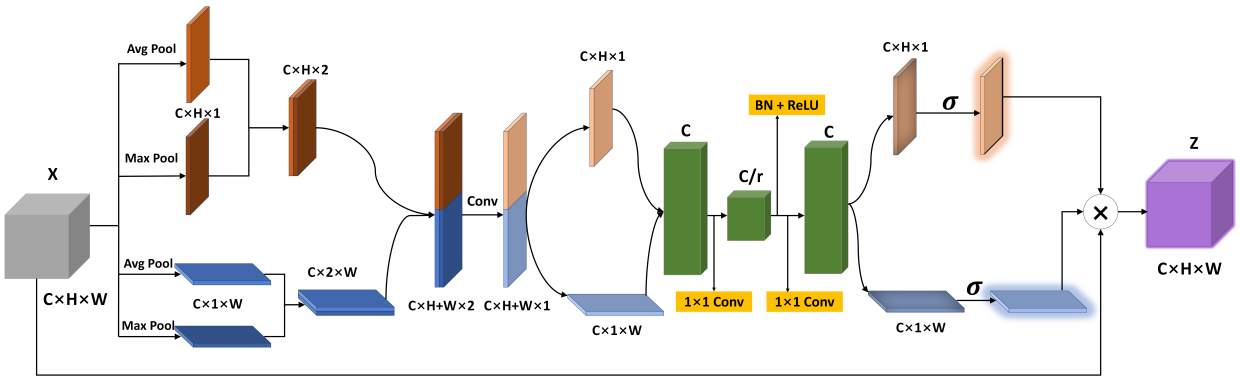

Fig. 4. GCA attention mechanism

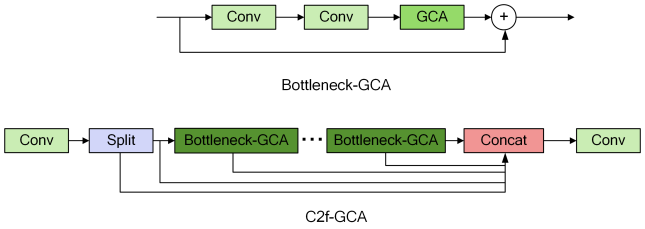

Fig. 5. C2f-GCA module

attention mechanism allows it to be integrated into the Bottleneck structure, forming the Bottleneck-GCA, which is further combined with the C2f module to create the C2f-GCA module, as illustrated in Fig. 5.

E. Focaler-CIoU Loss Function

Bounding box regression is critical in object detection. YOLOv8 employs the CIOU loss function, which accounts for overlap area, center distance, and aspect ratio to provide a comprehensive measure of the difference between the predicted box and the ground truth box. Although the CIOU loss function considers geometric relationships between boxes, it overlooks the impact of sample difficulty on regression, particularly the insufficient focus on hard samples such as small objects, which reduces detection accuracy.

To address the aforementioned issues, the Focaler-IoU bounding box regression loss function [26] has been introduced. Its formula is as follows:

$$L_{focaler-IoU} = 1 - IoU^{focaler},$$
$$IoU^{focaler} = \begin{cases} 0, & IoU < d, \\ \frac{IoU-d}{u-d}, & d \le IoU \le u, \\ 1, & IoU > u. \end{cases} \tag{5}$$

where $IoU^{focaler}$ is reconstructed using a linear interval mapping method that dynamically adjusts the weights of easy and hard samples. The parameters $[d, u] \in [0, 1]$ control the range of d and u, enabling $IoU^{focaler}$ to focus on regression samples with varying levels of difficulty.

To fully exploit the strengths of both approaches, Focaler-IoU and CIoU are combined to form the $L_{focaler-CIoU}$ loss function, effectively addressing sample imbalance in small target traffic sign detection. This enhances bounding box regression for hard small target samples, improves detection accuracy, and ensures precise identification and localization in complex traffic scenarios. Moreover, it boosts the model's robustness and adaptability. The Focaler-CIoU loss function is defined as follows:

$$L_{focaler-CIoU} = L_{CIoU} + IoU - IoU^{focaler} \tag{6}$$

IV. Experimental Results And Analysis

A. Dataset

In this study, the TT100K Chinese traffic sign dataset, jointly released by Tsinghua University and Tencent Lab, was used. After filtering, 45 categories with more than 100 instances were selected, resulting in 9,737 images, which were split into 7,230 for training and 2,507 for validation. However, some categories in the training set had fewer samples, potentially affecting the model's generalization. To address this, data augmentation techniques such as random rotation, brightness adjustment, and noise addition were applied to categories with fewer than 200 samples, ensuring each had over 200 images. This increased the total number of training images to 10,944.

B. Experimental Configurations

This experiment is conducted on a hardware environment consisting of a Windows 10 operating system, an NVIDIA GeForce RTX 3080 (10GB), an Intel Xeon Gold 6148 CPU, and 32GB of memory. The software environment includes Python 3.8, CUDA 11.8, and the PyTorch 1.11.0 deep learning framework. The training setup uses an image size of 640×640, the SGD optimizer, and runs for 150 epochs with a batch size of 32. The initial learning rate (lr_0) is set to 0.01, momentum to 0.937, and weight decay to 0.0005.

The performance metrics used to analyze the experimental results include Precision, Recall, Parameters (Params), Model Size, FPS, and Mean Average Precision (mAP). Among these, the mAP metric is evaluated using mAP@0.5 and mAP@0.5:0.95 [27].

TABLE I
ABLATION EXPERIMENT RESULTS

A	B	C	D	E	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Params (10^6)	Model size (MB)
					70.3	63.7	70.0	52.7	3.019	5.99
✓					75.7	69.5	76.3	57.2	2.073	4.35
✓	✓				76.5	68.1	76.8	57.9	1.885	4.01
✓		✓			78.7	67.0	76.6	57.5	1.995	4.2
✓			✓		76.4	70.7	77.0	57.5	2.073	4.35
✓				✓	78.3	68.8	77.2	57.8	2.088	4.42
✓	✓	✓			76.1	69.8	77.2	58.4	1.839	3.93
✓	✓	✓	✓		76.3	70.3	78.0	58.1	1.839	3.93
✓	✓	✓	✓	✓	78.9	69.8	78.7	58.9	1.854	4.00

TABLE II
COMPARISION EXPERIMENT RESULTS

Model	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Params (10^6)	Model size (MB)	FPS
Faster-R-CNN	–	–	62.8	44.7	28.36	316	12
YOLO-SG [20]	–	–	75.8	–	4.00	8.80	–
Ghost-YOLOv8 [21]	71.7	66.7	71.9	54.7	2.796	6.07	–
YOLOv5n	70.3	61.7	68.2	50.8	2.517	5.06	68
YOLOv7-tiny	43.8	51.8	46.2	32.2	6.134	11.9	33
YOLOv8n	70.3	63.7	70.0	52.7	3.019	5.99	67
YOLOv9-T	69.0	63.9	70.0	52.5	2.014	4.44	55
YOLOv10-N	70.2	59.7	66.6	50.2	2.724	5.53	82
GCA-YOLO (ours)	78.9	69.8	78.7	58.9	1.854	4.00	51

C. Ablation Experiment

To evaluate the impact of the proposed GCA-YOLO, nine ablation experiments are conducted using YOLOv8n as the baseline model. In these experiments, A represents the small target detection layer, B denotes the T-BiFPN multi-scale feature fusion structure, C refers to the C2f-CP module, D represents the Focaler-CIoU loss function, and E denotes the GCA attention mechanism. The results of these experiments are presented in Table I.

From the results in Table I, adding the small target detection layer significantly enhances the model's ability to identify small targets, reflected in notable improvements in Precision, Recall, and mAP. Additionally, the removal of the large target detection layer reduces the model's parameters and size by 31.33% and 27.38%, respectively. Replacing the original PANet structure with the T-BiFPN structure further improves multi-scale feature fusion, yielding increases of 0.5% in mAP@0.5 and 0.7% in mAP@0.5:0.95, along with reductions in parameters and model size. Incorporating the C2f-CP module, which integrates PConv, enhances Precision and mAP while further reducing model parameters and size. The Focaler-CIoU loss function effectively improves the model's ability to focus on hard small target samples, significantly enhancing detection performance. Although the GCA attention mechanism slightly increases model parameters and size, it strengthens the model's ability to focus on key information, leading to higher overall detection accuracy. By combining these improvements, the GCA-YOLO model achieves substantial performance gains over the original YOLOv8n. Specifically, it improves Precision by 8.6%, Recall by 6.1%, mAP@0.5 by 8.7%, and mAP@0.5:0.95 by 6.2%, while reducing model parameters and size by 38.57% and 33.21%, respectively.

D. Comparison Experiment

To evaluate the performance of the GCA-YOLO algorithm, comparative experiments were conducted on the TT100K dataset. The algorithm was compared against Faster R-CNN, YOLOv5n [28], YOLOv7-tiny [29], YOLOv8n, and advanced YOLO series models, including YOLOv9-T [30] and YOLOv10-N [31]. Additionally, comparisons were made with existing improved traffic sign detection models from the literature [20], [21]. The results of these experiments are presented in Table II.

As shown in Table II, while the two-stage Faster R-CNN achieves higher detection accuracy than some YOLO series algorithms, its large parameter count and model size hinder deployment on edge devices. In contrast, the GCA-YOLO algorithm surpasses all other algorithms listed in Table II, including the state-of-the-art YOLOv10-N, with improvements of 8.7%, 10.1%, 12.1%, and 8.7% in precision, recall, mAP@0.5, and mAP@0.5:0.95, respectively. Furthermore,

GCA-YOLO demonstrates significant efficiency, with only 1.854×10^6 parameters and a model size of 4 MB, making it highly suitable for edge device deployment. Additionally, its FPS performance satisfies real-time requirements, further enhancing its practicality.

V. Conclusion

This paper has proposed GCA-YOLO, an optimized version of YOLOv8n. By adding small-object detection layers and removing large-object ones, the model has enhanced small-object detection while reducing parameters. A T-BiFPN structure has been introduced for improved multi-scale feature fusion, and the C2f-CP module has reduced memory access to boost efficiency. The proposed GCA attention mechanism has captured spatial and channel dependencies, while Focaler-CIoU loss has addressed class imbalance by focusing on hard samples. Experiments on TT100K have shown notable gains in precision, recall, and mAP, along with reduced model size, making it suitable for edge-based traffic sign detection. Future work will focus on real-world deployment and further performance optimization.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (Grant Nos. 62072159, 61902112, 62401037) and the Henan Provincial Science and Technology Research Project (Grant No. 252102210218). Part of the work was carried out at ICTFICIAL Oy. It also receives support from the EU Horizon Europe programme under the 6G-Path (Grant No. 101139172, HORIZON-JU-SNS-2023) and 6G-SANDBOX (Grant No. 101096328, HORIZON-JU-SNS-2022) projects. The paper reflects only the authors' views, and the European Commission bears no responsibility for any utilization of the information contained herein.

References

[1] Y.-H. Lin and Y.-S. Wang, "Modular learning: Agile development of robust traffic sign recognition," *IEEE Transactions on Intelligent Vehicles*, 2023.

[2] X. R. Lim and *et al.*, "Recent advances in traffic sign recognition: approaches and datasets," *Sensors*, vol. 23, no. 10, p. 4674, 2023.

[3] C. Wang, H. Yu, and *et al.*, "Dependency-aware microservice deployment for edge computing: A deep reinforcement learning approach with network representation," *IEEE Trans. Mob. Comput.*, 2024.

[4] Z. Qin and W. Q. Yan, "Traffic-sign recognition using deep learning," in *Geometry and Vision: First International Symposium, ISGV 2021, Auckland, New Zealand, January 28-29, 2021, Revised Selected Papers 1*. Springer, 2021, pp. 13–25.

[5] B. R. Solunke and S. R. Gengaje, "A review on traditional and deep learning based object detection methods," in *2023 International Conference on Emerging Smart Computing and Informatics (ESCI)*. IEEE, 2023, pp. 1–7.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[7] S. Ren, K. He, and *et al.*, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[8] J. Cao, H. Cholakkal, R. M. Anwer, F. S. Khan, Y. Pang, and 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. 11 485–11 494.

[9] W. Liu, D. Anguelov, and *et al.*, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[10] T.-Y. Ross and G. Dollár, "Focal loss for dense object detection," in *proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2980–2988.

[11] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023.

[12] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[13] T. Gong, L. Zhu, F. R. Yu, and T. Tang, "Edge intelligence in intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 9, pp. 8919–8944, 2023.

[14] S. Liang, H. Wu, and *et al.*, "Edge yolo: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 345–25 360, 2022.

[15] P. Yuan, Z. Han, and X. Zhao, "Integrating the edge intelligence technology into image composition: A case study," *Peer-to-Peer Networking and Applications*, vol. 16, no. 4, pp. 1641–1651, 2023.

[16] W. Feng, R. Zhang, and *et al.*, "Exploring collaborative diffusion model inferring for aigc-enabled edge services," *IEEE Transactions on Cognitive Communications and Networking*, 2024.

[17] P. Yuan, R. Huang, J. Zhang, E. Zhang, and X. Zhao, "Accuracy rate maximization in edge federated learning with delay and energy constraints," *IEEE Systems Journal*, vol. 17, no. 2, pp. 2053–2064, 2022.

[18] R. Varghese and M. Sambath, "Yolov8: A novel object detection algorithm with enhanced performance and robustness," in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*. IEEE, 2024, pp. 1–6.

[19] J. Zhang, Z. Xie, J. Sun, X. Zou, and J. Wang, "A cascaded r-cnn with multiscale attention and imbalanced samples for traffic sign detection," *IEEE access*, vol. 8, pp. 29 742–29 754, 2020.

[20] Y. Han, F. Wang, W. Wang, X. Li, and J. Zhang, "Yolo-sg: Small traffic signs detection method in complex scene," *The Journal of Supercomputing*, vol. 80, no. 2, pp. 2025–2046, 2024.

[21] E. Xiong and *et al.*, "Ghost-yolov8 detection algorithm for traffic signs," *Comput. Eng. Appl*, vol. 59, no. 20, pp. 200–207, 2023.

[22] S. Chen, Z. Zhang, and *et al.*, "A semi-supervised learning framework combining cnn and multi-scale transformer for traffic sign detection and recognition," *IEEE Internet of Things Journal*, 2024.

[23] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 781–10 790.

[24] J. Chen, S.-h. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H. G. Chan, "Run, don't walk: chasing higher flops for faster neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 12 021–12 031.

[25] P. Su, H. Han, and *et al.*, "Mod-yolo: Rethinking the yolo architecture at the level of feature information and applying it to crack detection," *Expert Systems with Applications*, vol. 237, p. 121346, 2024.

[26] H. Zhang and S. Zhang, "Focaler-iou: More focused intersection over union loss," *arXiv preprint arXiv:2401.10525*, 2024.

[27] T.-Y. Lin, M. Maire, and *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[28] Ultralytics, "Yolov5," 2020, accessed: 2024-12-12. [Online]. Available: https://github.com/ultralytics/yolov5

[29] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 7464–7475.

[30] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao, "Yolov9: Learning what you want to learn using programmable gradient information," in *European Conference on Computer Vision*. Springer, 2025, pp. 1–21.

[31] A. Wang, H. Chen, and *et al.*, "Yolov10: Real-time end-to-end object detection," *arXiv preprint arXiv:2405.14458*, 2024.