Signal, Image and Video Processing
https://doi.org/10.1007/s11760-024-03252-8

ORIGINAL PAPER

GRFS-YOLOv8: an efficient traffic sign detection algorithm based on multiscale features and enhanced path aggregation

Guobo Xie[1] · Zhijun Xu[1] · Zhiyi Lin[1] · Xingming Liao[1] · Teng Zhou[2]

Received: 29 February 2024 / Revised: 3 April 2024 / Accepted: 28 April 2024

Abstract
Traffic sign detection is a crucial element of advanced driver assistance systems (ADAS) for environmental perception. However, challenges persist in the detection of small-scale targets and susceptibility to adverse weather, varying light conditions, and occlusions. To address this issue, a novel traffic sign detection algorithm, GRFS-YOLOv8, is proposed. GRFS-YOLOv8 introduces an enhanced greater receptive field-SPPF (GRF-SPPF) module to replace the original SPPF module, enabling the capture of richer multiscale features from image feature maps. Additionally, a new SPAnet architecture is designed by introducing two "shortcut" paths and additional smaller target detection layers to enhance path aggregation capabilities. This architecture propagates complete semantic information, alleviating the reduction in resolution for small targets and enhancing the model's capability to detect them. Finally, by employing GhostConv and C2fGhost to replace multiple CBS and C2f modules in Backbone and Neck, cost-effective linear operations are utilized to obtain more feature maps, thus reducing the computational cost of the model. Experimental validation across multiple datasets demonstrates the efficacy and adaptability of GRFS-YOLOv8, achieving an 80.3% mAP and 72.4% Recall in CCTSDB 2021, a 71.2% mAP and 95% Recall in TT100K, and a 94.0% mAP and 96.0% Recall in GTSDB, surpassing mainstream detectors and comparative methods.

Keywords Traffic sign detection · Challenging environments · Small object detection · YOLOv8

1 Introduction

As an integral part of transport systems, traffic signs play a crucial role in providing road information, guidance, and ensuring traffic order and safety. Traffic sign detection constitutes a key element of advanced driver assistance systems (ADAS) and has been extensively researched in recent years. However, when capturing images of traffic signs through in-vehicle devices while driving on actual roads, these images are easily affected by various factors such as weather conditions, lighting, motion, and obstacles. These factors often lead to poor visual appearances, lack of contextual reference information, and since traffic signs usually occupy a small portion of the entire field of view, detecting them becomes more complex, increasing the risks of false positives and missed detections, which could potentially lead to accidents. Therefore, achieving accurate traffic sign detection remains a significant challenge. Presently, traffic sign detection can be broadly categorized into two major types: traditional methods relying on color, shape, and deep learning-based popular detection algorithms.

Traditional methods for detecting traffic signs in images rely on the object attributes of detected traffic signs, such as color and shape features, which are used to identify regions and match traffic signs in images, forming the basis of traffic sign detection. For instance, David et al. [1] proposed an innovative approach that integrates the CIELab + color space with Support Vector Machines (SVM) for traffic sign recognition. However, this method is limited by its ability to recognize specific shapes and computational requirements. In contrast, Natthathida et al. [2] focused on the HSV color model, facilitating the extraction of red and blue traffic signs and organizing them cleverly during the shape extraction process. Garcí-Garrido et al. [3] proposed using the Hough transform

Guobo Xie and Zhijun Xu have contributed equally to this work.

✉ Zhiyi Lin
lzy291@gdut.edu.cn

Zhijun Xu
2112205110@mail2.gdut.edu.cn

[1] School of Computing, Guangdong University of Technology, Guangzhou 510000, Guangdong, China

[2] School of Mechanical and Electrical Engineering, Hainan University, Haikou 570000, Hainan, China

Published online: 20 May 2024

method for detecting traffic signs from image contours. Nevertheless, these traditional methods relying on manual feature extraction suffer from low recognition accuracy, imprecise results, high computational cost, and slow processing speed.

Compared to traditional methods, deep learning techniques based on genuine feature learning have proven to be more effective in extracting traffic sign features and have become the mainstream research direction for traffic sign detection, extensively explored and applied by researchers. For example, Li et al. [4] designed a detector combining Faster R-CNN and MobileNet to locate small-scale traffic signs using color and shape. However, their bounding box localization method was customized for specific categories of traffic signs. Domen et al. [5] adjusted Mask R-CNN to address the detection and recognition of various traffic signs by improving the sample selection mechanism of the region proposal network and modifying the ROI digit passing method, but still faced some issues with network classification missing detections. Yan et al. [6] introduced a new lightweight Feature Detection (FD) model based on SDD to improve traffic sign detection, particularly under complex lighting conditions. However, their experiments were conducted on relatively small datasets. Although the R-CNN or SDD [7] series of deep learning methods have achieved certain successes in traffic sign detection, they lag behind the YOLO algorithm in terms of processing speed, real-time performance, and detection of smaller targets. Consequently, YOLO-based traffic sign detection has become a current research hotspot.

Researchers have proposed several traffic sign detection algorithms based on YOLO series deep learning. For instance, YU et al. [8] introduced a fusion model combining YOLOv3 and VGG19 networks, introducing a multi-image-based traffic sign detection algorithm. This model requires considerable training time in traffic sign detection as it involves extracting multiple images to complete the detection task. Shi et al. [9] introduced a low-parameter SC-YOLOv5 model, incorporating cross-stage attention networks and a dense neck structure to effectively fuse detailed and semantic information, along with the integration of SIOU loss function for training optimization. However, the exploration of this model for the task of traffic sign detection in complex environments remains limited. However, this model has limited exploration for traffic sign detection tasks in complex environments. The latest product in the YOLO series, YOLOv8, introduced new functionalities and improvements based on its predecessors, demonstrating impressive performance in speed and accuracy. Luo et al. [10] proposed a solution for identifying blurry traffic signs by introducing a simple and efficient image fusion method, significantly improving detection accuracy, especially for blurry traffic signs. Soylu et al. [11] provided valuable insights by evaluating and comparing various YOLOv8 models in traffic sign detection, guiding researchers in developing optimized YOLOv8 models. While the YOLOv8 model has shown good performance, practical applications reveal issues. Nicholas et al. [12] found that although YOLOv8 performed well on the Lisa dataset, it struggled on the glare dataset he created, indicating difficulties in traffic sign detection under adverse environmental conditions. Additionally, YOLOv8 detection faces challenges in detecting small targets.

Therefore, to address the challenges of YOLOv8 in detecting small targets and the impact of adverse environmental conditions on detection effectiveness, this paper proposes a traffic sign detection algorithm, GRFS-YOLOv8. Aimed at achieving more accurate traffic sign detection, the experimental results on multiple datasets validate the effectiveness of the proposed model, contributing as follows:

- Replacement of the original SPPF module in the Backbone with an enhanced GRF-SPPF to broaden the model's receptive field. This aids in extracting richer multiscale feature information from feature maps, facilitating accurate target detection in complex environments.
- Introduction of two "shortcut" paths and additional smaller target detection layers to design the new SPANet, aiming to enhance the path aggregation network (PANet) in the traditional YOLOv8 model. SPANet incorporates two "shortcut" paths from lower to higher layers between the backbone and neck networks of PANet, allowing complete semantic information to propagate to subsequent layers, mitigating resolution loss of small targets as the network deepens, while enhancing the model's detection capability for additional smaller targets.
- Effective integration of GhostNet into the feature extraction and fusion network, utilizing cost-effective linear operations to obtain additional feature maps, thereby reducing the model's complexity and computational costs.

2 The principle behind the YOLOv8 network architecture

YOLO shifts the detection problem from classification to regression. The algorithm divides the image into n^2 grids, predicting the presence of objects within each grid by regressing bounding boxes, using a single CNN applied to the input image, directly computing class confidences and their respective positions. The architecture of YOLOv8, as depicted in Fig. 1, comprises Input, Backbone, Neck, and Head components.

Input: In the input phase, YOLOv8 employs the same data augmentation strategy as YOLOv5, including techniques such as random cropping, flipping, and scaling. The key

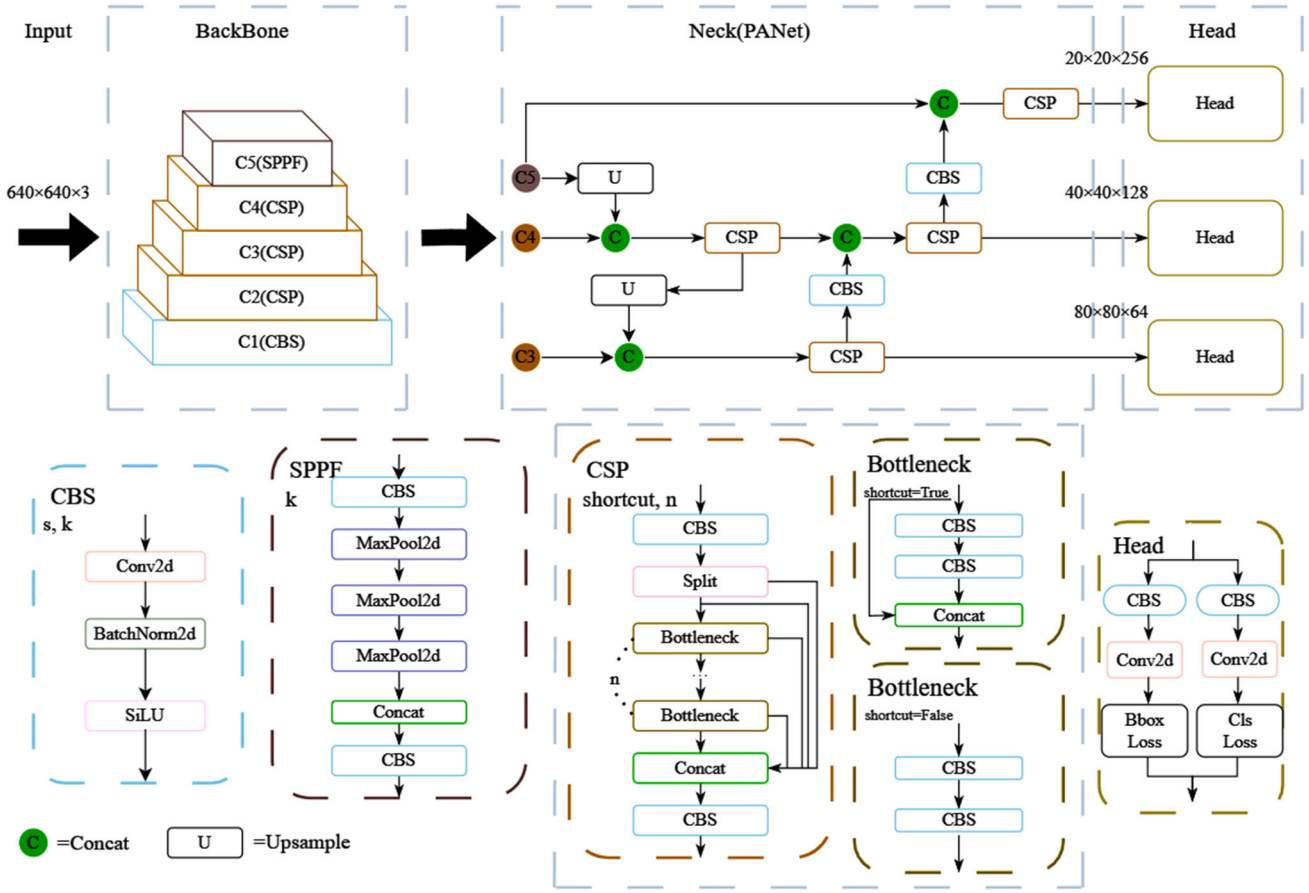

Fig. 1 Original structure of YOLOv8: feature extraction utilizing the CSP-based backbone network. Multiscale feature fusion within the neck network. Detection across various scales in the head network based on a three-channel input image

difference lies in YOLOv8's decision to disable data augmentation in the final ten epochs.

Backbone: The initial part of the network is termed the Backbone, tasked with mapping diverse input images into high-level feature representations that encompass positional and semantic information from various regions of the image. This segment includes CBS modules, CSP modules, and an SPPF module. Within the main network, multiple stacked CBS modules extract various features from the image. The CSP module not only extracts deeper features from image features but also reduces the model's computational cost. The SPPF module is employed to enhance the network's perception of targets of different sizes.

Neck: The middle segment of the network, known as the Neck, serves to connect the Backbone and Head networks, responsible for further processing the features extracted by the Backbone network. YOLOv8 adopts PANet [13] as its Neck network, aiding in feature fusion across different scales and minimizing semantic information loss.

Head: The third part of the network is the Head, responsible for generating the final detection results. Unlike YOLOv5, YOLOv8 employs a state-of-the-art decoupled head structure, separating classification and detection heads. This involves two parallel branches used for independently extracting category and position features. Although YOLOv8 is designed as a general object detector, it may encounter the following challenges when applied to traffic sign detection:

- *Limited generalization*: YOLOv8 utilizes relatively simple data augmentation strategies, typically integrating basic methods like random cropping, flipping, and scaling. This might diminish the model's ability to generalize in complex scenarios, rendering it insufficient for diverse traffic sign detection tasks in varying environments.
- *Weak feature extraction for small targets*: During training, YOLOv8 uses a feature pyramid to extract features at different scales, achieving commendable results in detecting large targets. However, its capability to detect small targets remains suboptimal. Traffic signs generally occupy only

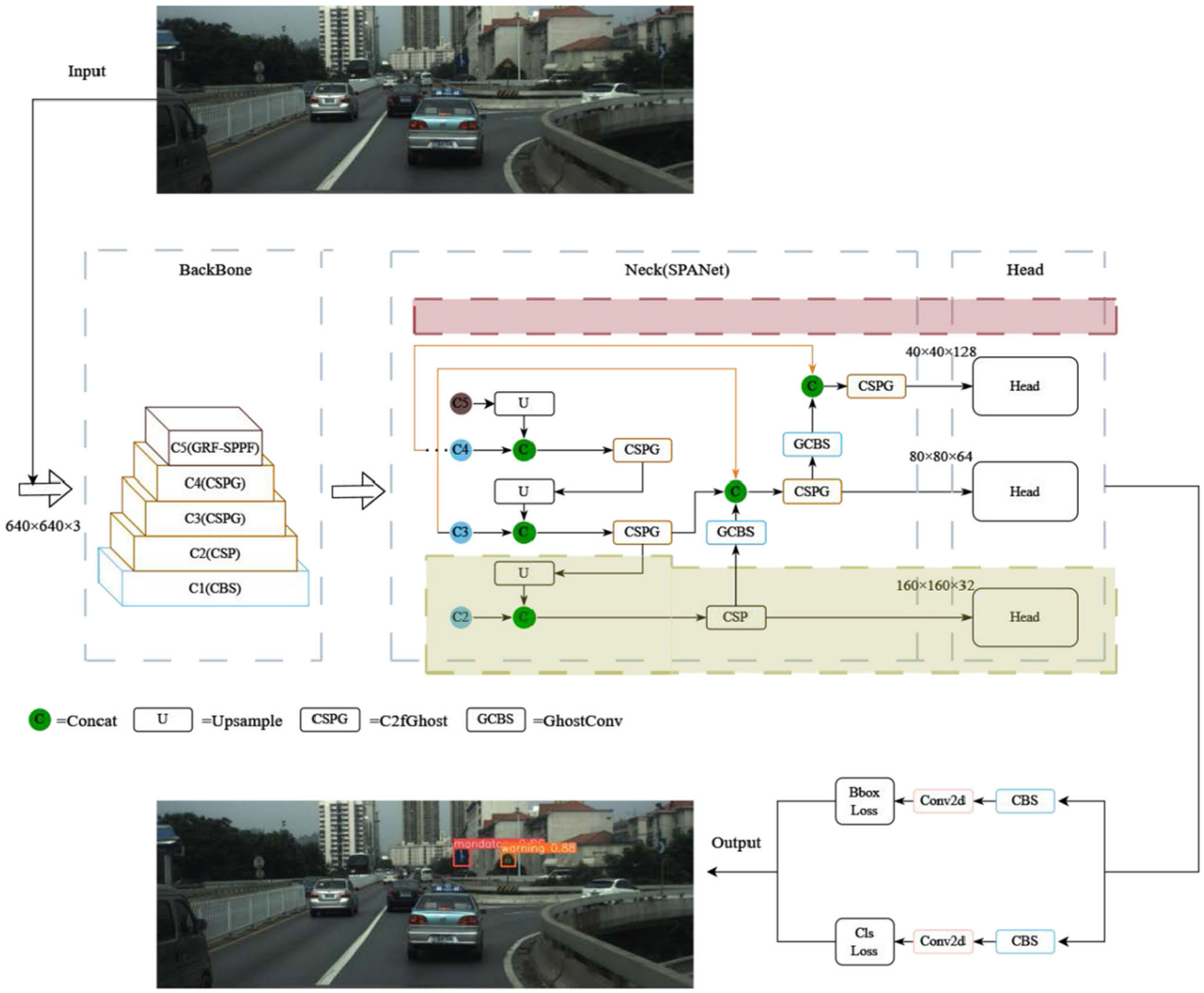

Fig. 2 Overall architecture of GRFS-YOLOv8: feature extraction utilizing CSP and CSPG structures in the backbone, integrating the GRF-SPPF Structure with a larger receptive field. The neck network (SPANet) encompasses additional layers (highlighted in green in the figure). Orange arrow lines in the figure denote two new "shortcut" connections, while the red overlay signifies the removal of the large object detection head, focusing the model on small object detection

a small portion of the entire field of view. Small traffic signs tend to lose crucial feature information across multiple layers of YOLOv8's core network, leading to missed detections or false positives.

- *Excessive redundant computations*: YOLOv8's original architecture includes a considerable number of conventional convolutional modules (CBS) in a relatively deep network structure. The stacking of numerous convolutional layers not only significantly increases the number of parameters and computational load (FLOPs) but also generates redundant feature maps.

3 The detailed architecture of GRFS-YOLOv8 network

In response to the limitations of YOLOv8, this paper introduces an enhanced traffic sign detection algorithm, GRFS-YOLOv8, as illustrated in Fig. 2. Firstly, the enhanced GRF-SPPF (Greater Receptive Field-SPPF) module has replaced the SPPF module to bolster the detection capability of traffic signs in complex scenarios. Secondly, the original PANet in the model has been replaced with an improved version called SPANet ("shortcut" PANet), which incorporates subsequent network layers capable of integrating more semantic information, thereby enhancing the network's detection performance in complex scenarios and with smaller targets.

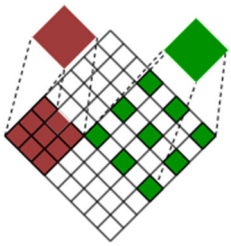

Fig. 3 Standard convolution in red (dilation rate = 1, receptive field = 3); and dilated convolution in green (dilation rate = 2, receptive field = 5)

Finally, GhostNet has been employed to enhance both the Backbone and Neck, reducing model complexity and boosting detection speed.

3.1 GRF-SPPF

YOLOv8 employs the Spatial Pyramid Pooling Fusion (SPPF) structure, which combines serial and parallel pooling structures to expand the receptive field. However, this method may have limitations in certain scenarios because fixed-scale pooling operations may not fully adapt to the diverse scales required for traffic sign detection. Additionally, it may not sufficiently capture the subtle features and contextual information crucial for comprehensive traffic sign analysis.

Integrating a larger receptive field into deep neural networks contributes significantly to enhancing the model's understanding of contextual information, which is crucial for traffic sign detection. Expanding the receptive field further helps to encompass a wider range of surrounding information, reducing misinterpretation and enhancing the model's semantic understanding, and feature extraction capabilities, particularly in complex environments and for detecting small objects, enabling each convolutional output to contain richer information.

Typical methods to expand the receptive field involve additional convolution and pooling operations on feature maps. However, this sequence of operations not only leads to feature loss but also incurs higher computational costs. The widely used "dilated convolution" [14] in image segmentation extends the receptive field while preserving the resolution of image feature maps, replacing the downsampling and upsampling processes. This convolution introduces a parameter called "dilation rate", defining the gap between the values processed by the convolutional kernel. The comparison between regular and dilated convolutions is illustrated in Fig. 3.

Given this, we introduce dilated convolutions to enhance the existing SPPF module, termed as GRF-SPPF shown in Fig. 4. The improvement strategy is as follows:

- Introduce parallel dilated convolution layers after the max-pooling operation in the original SPPF structure, with dilation rates of 2, 4, and 8, constructing convolutional kernels with different receptive fields.
- Incorporate a residual branch to alleviate gradient vanishing issues and capture global image features. The residual branch includes an average pooling layer, a 1×1 convolution layer, and upsampling.
- Employ two different fusion processes for feature maps with different channel numbers.

GRF-SPPF further enhances the network depth and the receptive field, while preserving the model's learning ability in deeper layers and maintaining the resolution of image feature maps. It captures more contextual and background information by extracting multi-scale features of the image and compensates for the model's loss of traffic sign information in the feature extraction process, especially for small-scale signs and signs within complex backgrounds. Hence, it effectively enhances the detection accuracy and robustness of our model.

3.1.1 SPANet

In the original YOLOv8 model, the three detection heads correspond to target resolutions of 80×80, 40×40, and 20×20. Detecting small traffic signs at such resolutions evidently poses challenges.

To enhance the accuracy of small traffic sign detection, a common approach involves adding a small object detection layer. Within the feature fusion network, an additional upsampling is applied to the 80×80 feature map, generating a new output feature map of 160×160, facilitating the utilization of four output feature layers for target detection. However, this method has its drawbacks. Compared to regular-sized objects, small traffic signs are more prone to overlap with other objects and might be partially occluded by objects of different sizes. Moreover, smaller signs may be misinterpreted by larger objects, and features extracted from deeper layers may lack sufficient information about these small objects. These limitations can cause the algorithm to overlook small objects during the learning process, making their differentiation and localization challenging, thus reducing detection performance.

In the YOLOv8 model, images pass through the feature extraction network and the feature fusion network before reaching the detection heads. As the network deepens, image features are gradually abstracted, compressed and represented at higher levels. This process can obscure or distort some semantic features due to information compression and abstraction, potentially affecting the detection of small objects or sensitive subtle features. The processing

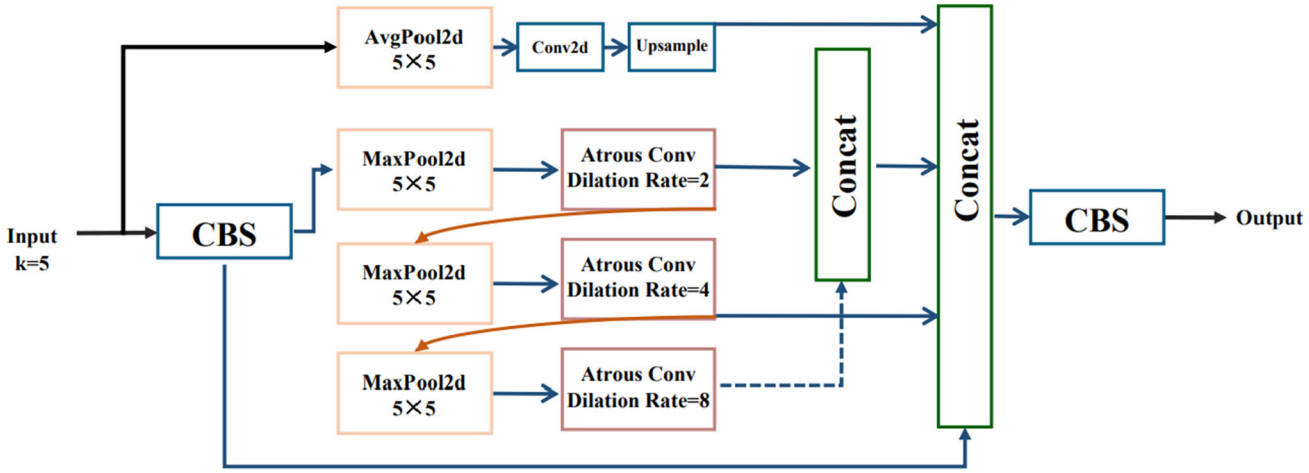

Fig. 4 The GRF-SPPF structure: enhanced image understanding through multiscale feature fusion using parallel dilated convolutions and residual branches

of high-level features may lack sufficient detailed information, reducing detection accuracy.

Consequently, our approach revises the addition of the small object detection layer by removing the 20 × 20 output designed specifically for large targets. This technique minimizes the impact of larger targets on smaller ones, significantly enhancing the accuracy of detecting small traffic signs while considerably reducing the number of model parameters and computational costs. Moreover, inspired by the BIFPN [15] approach, an additional edge was introduced between the original input and output nodes, allowing for the learning of the significance of different input features to achieve unique fusion. Through careful observation, opportunities were identified to improve the flow of information between the P3 and P4 layers by introducing two additional "shortcuts" between P2 and P3 and between P3 and P4. We refer to this redesigned PANet as the SPANet, as shown in Fig. 5.

Since the C3 and C4 layers reside in the lower levels of the Backbone network, containing rich and comprehensive semantic and positional information, the introduction of these two new "shortcuts" enables the YOLOv8 model to better integrate positional and semantic information, thereby enhancing its path aggregation capability and improving its ability to detect small traffic signs in complex environments.

3.2 Ghost module

YOLOv8 employs standard convolutions by convolving all feature channels with corresponding convolutional kernels and then amalgamating all results into the output. Outputs from standard convolutional layers usually contain a surplus of redundant feature maps, some of which might be similar to each other. Hence, generating these redundant feature maps individually with substantial FLOPs and parameters is unnecessary. The Ghost module [16] (shown in Fig. 6) solves this problem by first acquiring a subset of feature maps using standard convolution. Then, it generates additional feature maps using linear operations, and finally concatenates distinct feature maps. This method produces more feature maps with fewer parameters and less computation.

In standard convolution, given an input $x \in R^{c\times h\times w}$, where c denotes the input channel, and h and w represent the height and width of the input, the formula for generating n feature maps in any convolutional layer can be expressed as follows:

$$y = x \otimes f + b \tag{1}$$

Here, $\otimes$ denotes a convolutional operation, b is the bias term, $y \in R^{h'\times w'\times n}$ represents the output feature maps with n channels, and $f \in R^{c\times k\times k\times n}$ is the convolutional filter for this layer. Additionally, h' and w' are the output height and width, and $k \times k$ is the kernel size of filter f. Therefore, in standard convolutional operations, the FLOPs can be expressed by the following formula:

$$n \cdot h' \cdot w' \cdot c \cdot k \cdot k \tag{2}$$

Typically, the values of n and c in filter f are relatively large, thus limiting the parameter scale based on the dimensions of input and output feature maps.

Assuming a linear operation with kernel size $d \times d$ and each base feature having s redundant features, where the number of channels c is significantly larger than the redundant features s, the computational cost using the Ghost module involves an identity mapping and can be represented as:

$$\frac{n}{s} \cdot h' \cdot w' \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot h' \cdot w' \cdot d \cdot d \tag{3}$$

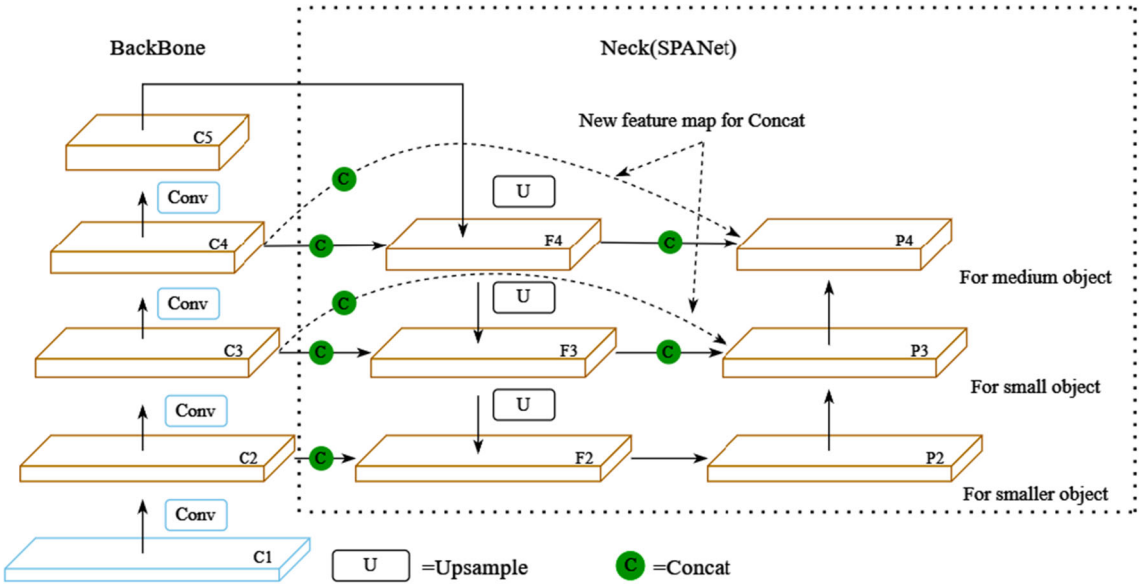

Fig. 5 The SPANet structure: enhancing YOLOv8 path aggregation capability through optimized PANet structure integration of spatial and semantic information

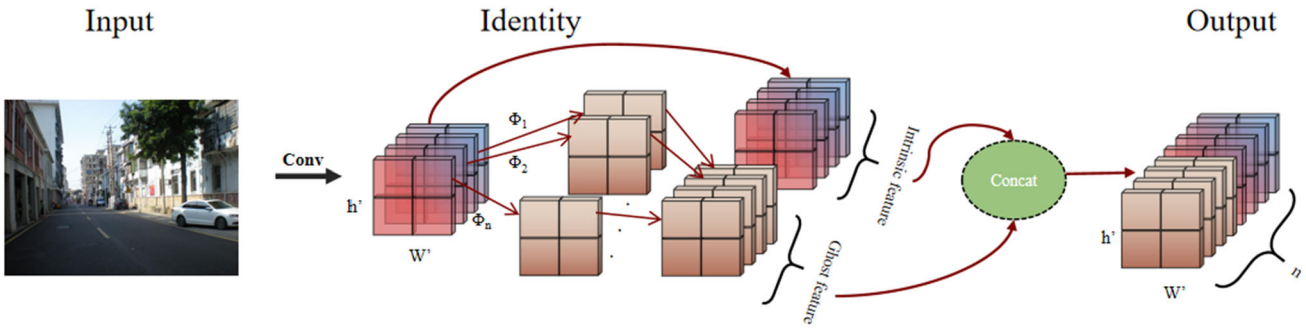

Fig. 6 Implementation of the Ghost Module: 'Identity' denotes an existing identity mapping. 'Φ' indicates a linear operation

The compression ratio of the computational cost compared to standard convolution is:

$$r_1 = \frac{n \cdot h' \cdot w' \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot h' \cdot w' \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot h' \cdot w' \cdot d \cdot d} = \frac{c \cdot k \cdot k}{\frac{1}{s} \cdot c \cdot k \cdot k + \frac{(s-1)}{s} \cdot d \cdot d} \approx \frac{(s \cdot c)}{s + c - 1} \approx s \tag{4}$$

The number of parameters in the Ghost module is:

$$\frac{n}{s} \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot d \cdot d \tag{5}$$

The compression ratio of the number of parameters compared to standard convolution is:

$$r_2 = \frac{n \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot d \cdot d} \approx \frac{s \cdot c}{s + c - 1} \approx \frac{s \cdot c}{c} = s \tag{6}$$

It can be observed that the computational load of the Ghost module is *1/s* compared to the standard convolution. In this paper, building on the concept of Ghost module, we introduce GhostConv (Fig. 7a) and C2fGhost (Fig. 7b).

In GhostConv, a cost-effective linear operation utilizes a 5×5 convolution to generate diverse feature maps, which are then concatenated to form a new output. In the C2fGhost structure, a more efficient GhostBottleneck gradient flow branch is primarily composed of two stacked Ghost modules, increasing channel numbers in the first module and subsequently decreasing them in the second module to match the "shortcut" path. Both GhostConv and C2fGhost not only reduce redundant computations but also maintain similarity recognition performance. Integrating them into the YOLOv8 model maintains model accuracy while minimizing parameters and computational costs.

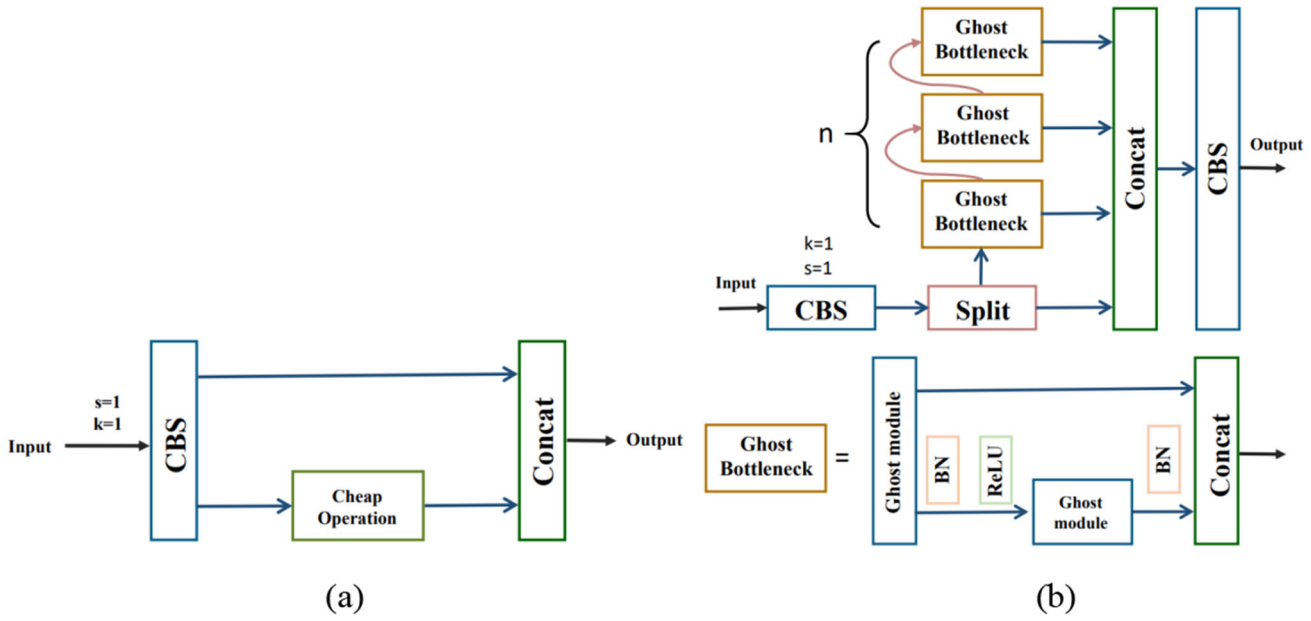

Fig. 7 The GhostConv structure (**a**) and the C2fGhost structure (**b**)

4 Experiments and discussions

In this section, we conducted performance evaluations of GRFS-YOLOv8 on three public datasets: CCTSDB 2021 [17], TT100K [18], and GTSDB [19], for traffic sign detection. Comparative analyses were performed against mainstream object detectors and other methods to demonstrate the superiority of GRFS-YOLOv8.

4.1 Applied datasets

The benchmark dataset utilized in this paper is the CCTSDB 2021 traffic sign dataset released by Changsha University of Science and Technology. To demonstrate the superior generalization capability of GRFS-YOLOv8, we conducted comparative evaluations on the TT100K and GTSDB datasets. Both TT100K and GTSDB datasets were divided into training and testing sets in an 8:2 ratio. During the actual training process, the training sets were further split into training and validation sets using a 9:1 ratio.

4.1.1 CCTSDB 2021 datasets

The CCTSDB 2021 dataset, which expands on CCTSDB 2017, includes more than 4,000 meticulously annotated real-world traffic scene images meticulously annotated. It replaced many of the initially easily detectable images with challenging samples to adapt to intricate and dynamic detection environments. This dataset consists of 17,856 images that depict urban roads and highways, covering three primary traffic sign categories: "mandatory", "prohibitory" and "warning". The training set within CCTSDB 2021 comprises 16,356 images, while the test set contains 1,500 images. The test set's intricate samples are categorized based on weather and environmental conditions, including cloudy, foggy, nighttime, rainy, snowy, and clear weather conditions.

4.1.2 TT00K datasets

The TT100K dataset, a collaboration between Tencent and Tsinghua University, comprises 45 distinct categories of traffic signs. However, due to the limited representation for many categories within the TT100K dataset, there exists a noticeable imbalance in sample distribution. Furthermore, the dataset includes background images lacking specific category labels. After preprocessing, the TT100K dataset consists of 9170 images, of which 7222 images are allocated to the training set and 1948 images to the test set.

4.1.3 GTSDB datasets

The GTSDB (German Traffic Sign Detection Benchmark) is a German traffic sign dataset categorized into four main classes: "Prohibitory", "Danger", "Mandatory", and "Other". This dataset consists of 900 images, of which 600 images are allocated to the training set and 300 images to the test set.

4.2 Experimental environment and settings

The experiments were conducted on a system running Linux, equipped with an AMD EPYC 7543 32-core Processor and

an A5000 (24G) GPU. PyTorch version 1.10.1 served as the development platform. The training parameters were set as follows: an initial learning rate of 0.01, Stochastic Gradient Descent (SGD) as the optimizer, a weight decay of 0.0005 to prevent overfitting, and 200 epochs to ensure adequate learning and parameter convergence. This configuration was consistently used across all experiments to guarantee the comparability of the results.

4.3 Evaluation metrics

The experiment employed Precision (P), Recall (R), Average Precision (AP), and F1 score as evaluation metrics. Precision (P) signifies the ratio of true positive samples among the detected targets. Recall (R) indicates the probability of correctly identifying positive samples among all samples. AP represents the average precision of the detector at various Recall levels, while mAP serves as an indicator of the overall model performance. The F1 score, a harmonic mean, offers a balance between precision and Recall. The formulas for calculating each evaluation metric are as follows:

$$P = \frac{TP}{(TP + FP)} \tag{7}$$

$$R = \frac{TP}{TP + FN} \tag{8}$$

Here, *TP* represents the number of detections that match both the ground truth and the detected results. *FP* denotes detections present in the detected results but absent in the ground truth. *FN* refers to detections present in the ground truth but absent in the detected results.

$$AP = \int_0^1 P(R)dR \tag{9}$$

In general, AP is the average of P, and its value varies with the IoU threshold.

$$mAP = \frac{1}{r}\sum_{j=1}^{r} AP_j \tag{10}$$

where r is the number of categories. AP_j represents the average precision of the j-th category at different IoU thresholds.

$$F1 = \frac{2 \cdot (P \cdot R)}{P + R} \tag{11}$$

4.4 Performance on the CCTSDB 2021 dataset

4.4.1 Dissociation experiment research

In this section, we conducted ablation experiments on the CCTSDB 2021 dataset to confirm the influence of each module on the network. We progressively integrated GRF-SPPF, Ghost modules (GhostConvand C2fGhost), and SPANet into the baseline model. The experimental results are presented in Table 1.

The introduction of GRF-SPPF to YOLOv8 resulted in a 0.9% increase in both mAP and R values. This highlights how our unique multi-scale feature fusion structure can extract richer feature information and better capture the characteristics of small objects. By replacing portions of the standard convolutions and C2f structures in the Backbone and Neck networks with GhostConv and C2fGhost, the number of model parameters decreased by 23%, optimizing the model while maintaining performance in similarity recognition. Additionally, the introduction of SPANet led to a 2% improvement in F1 and a 2.5% increase in mAP, while further reducing the number of model parameters by 28%. This indicates that our designed Neck network can more effectively fuse contextual information while retaining lower-level positional and semantic details.

4.4.2 Comparison of experimental results on difficult sample

To validate the effectiveness and advancement of GRFS-YOLOv8, we compared it to several algorithms including Faster R-CNN [20], SSD, RetinaNet [21], Efficientdet [15], YOLOv5[22], YOLOv6 [23], YOLOv7 [24], YOLOX [25], YOLOv8, Ghostnet-YOLOv5s [26], YOLOv5-TDHSA [27], HIC-YOLOv5 [28], Depth improved-YOLOv5 [29], LF-YOLO [30], TP-YOLO [31], C2net-YOLOv5 [32], SG-YOLO [33], LLTH-YOLOv5 [34], DW-YOLOv7-tiny [35].

Table 2 presents the models' performance across six dimensions: F1, R, mAP, mAP50-95, Par, and G (some specific algorithm metrics are missing as the authors did not provide the required test data for those particular algorithm metrics). Our proposed GRFS-YOLOv8 demonstrates the highest mAP and R indices. It outperforms the second-ranking model by 0.1% and 0.3%, respectively. This highlights the significant performance advantage of GRFS-YOLOv8, mainly attributed to its capability to capture more feature information and efficiently fuse multi-scale data.

While its F1 score is slightly lower than that of YOLOv6, GRFS-YOLOv8 achieves higher accuracy while maintaining fewer parameters and computational loads. Overall, GRFS-YOLOv8 strikes a favorable balance between precision and model size.

Table 1 Impact of different modifications on the accuracy of the proposed model compared to the baseline model YOLOv8 (Unit: %)

Model	Module							
	GRF-SPPF	Ghost module	SPANet	F1	R	mAP	mAP_{50-95}	Par (M)[a]
YOLOv8	✘	✘	✘	77.0	71.1	77.3	48.8	3.01
	✔	✘	✘	77.0	72.0	78.2	49.5	3.28
	✔	✔	✘	77.0	72.0	77.8	49.1	2.31
	✔	✔	✔	**79.0**	**72.4**	**80.3**	**51.8**	1.65

In the table, bold font indicates the optimal performance indicators, "✔" denotes the inclusion of the module, and "✘" signifies its exclusion
[a]Par indicates the number of Parameters

Table 2 Performance comparison of GRFS-YOLOv8 and other methods on the CCTSDB 2021 dataset (Unit: %)

Method	F1	R	mAP	mAP_{50-95}	Par (M)[a]/G[b]
Faster R-CNN(VGG)	33.7	55.0	29.2	10.6	13.80/15.5
SGG(VGG)	7.3	3.8	34.0	13.1	13.80/15.5
RetinaNet (resnet50)	11.0	5.9	12.8	5.5	2.50/4.1
Efficientdet	17.3	9.7	18.5	–	6.60/6.1
YOLOv5	63.0	52.9	61.2	34.6	7.03/16.0
YOLOv6	**79.9**	52.9	61.2	50.3	4.70/11.4
YOLOv7	52.0	44.3	47.7	26.3	6.02/13.2
YOLOX	71.0	<u>72.1</u>	79.4	38.9	5.06/6.5
YOLOv8	77.0	71.1	77.3	48.8	3.01/8.2
Ghostnet-YOLOv5s	–	–	75.5	46.6	5.85/12.4
YOLOv5-TDHSA	72.0	–	69.8	–	12.22/–
HIC-YOLOv5	73.0	67.1	78.4	46.1	9.30/30.7
Depth improved-YOLOv5	–	–	<u>80.2</u>	–	6.30/11.8
LF-YOLO	60.0	49.1	54.1	29.5	7.25/16.3
TP-YOLO	78.0	69.4	77.1	48.1	4.29/9.2
C2net YOLOv5	76.2	69.3	75.2	–	– /–
SG YOLO	–	–	74.5	–	4.82/9.3
LLTH YOLOv5	–	70.1	69.4	**54.3**	43.87/–
DW YOLOV7 tiny	–	–	58.4	–	5.20/12.1
GRFS YOLOv8 (ours)	<u>79.0</u>	**72.4**	**80.3**	<u>51.8</u>	1.65/9.8

In the table, bold font indicates the optimal performance indicators, while font with underlining represents secondary performance indicators. The symbol "–" indicates data not provided by the author
[a]Par indicates the number of parameters
[b]G indicates GFLOPs

To illustrate the outstanding performance of GRFS-YOLOv8 in a visual manner, we present its comparison with other algorithms during the training process. Figure 8 illustrates the comparative results between GRFS-YOLOv8 and other algorithms in terms of Loss, Recall, mAP, and mAP_{50-95} metrics. From Fig. 8b–d, it's noticeable that GRFS-YOLOv8 demonstrates smoother upward trends and consistently outperforms other algorithms on all metrics, highlighting its superior performance.

4.4.3 Experimental results across different categories

To further enrich the content of experimental results, this section delves into an in-depth exploration of the results across different categories ('mandatory', 'prohibitory', 'warning'). Table 3 presents the Precision (P) and Recall (R) results for each category when the IoU threshold is set to 0.5. Clearly, it is evident that our proposed method significantly outperforms other approaches across all categories in terms of both P and R metrics.

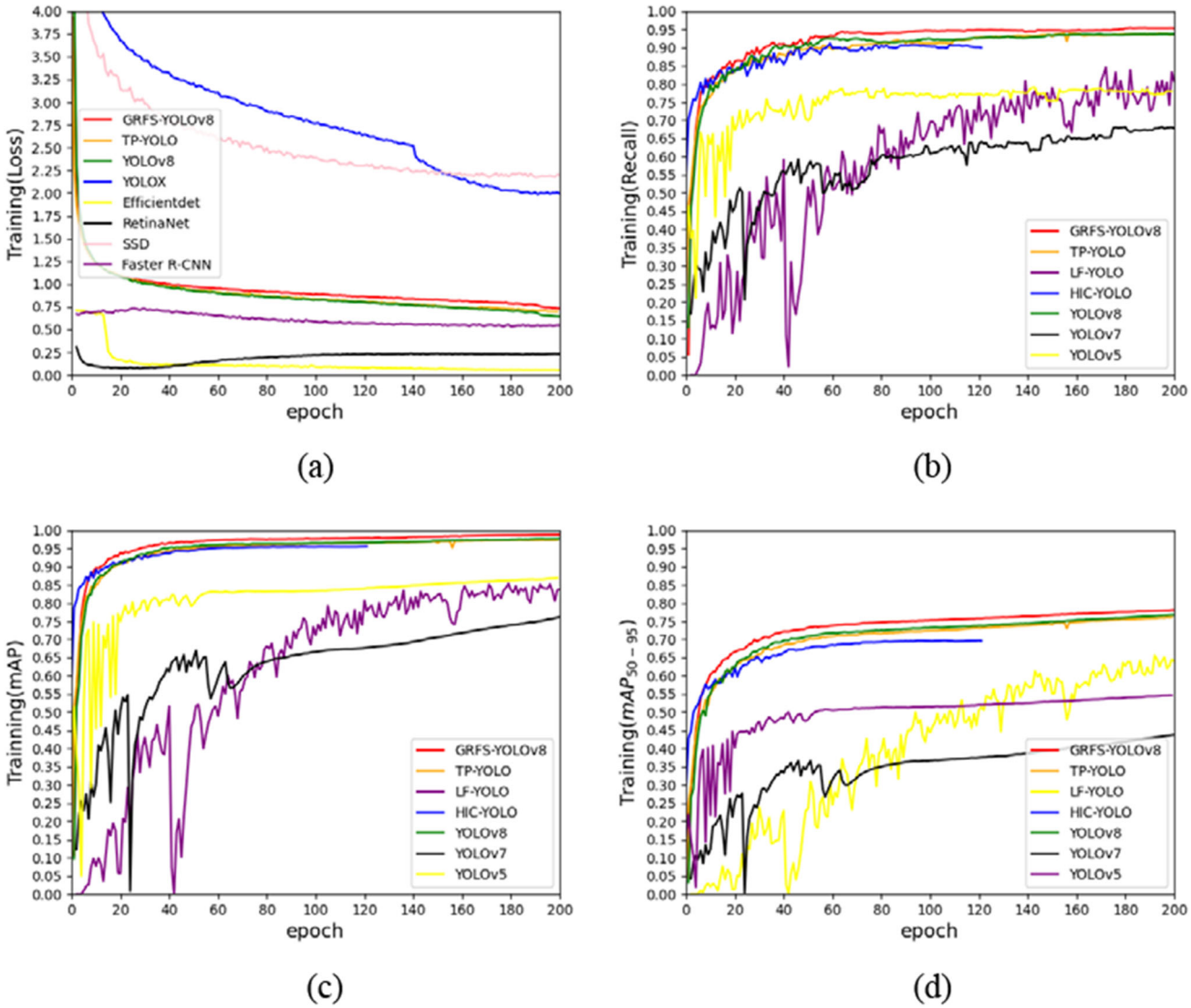

Fig. 8 Comparative Visualization of GRFS-YOLOv8 against Other Methods: **a** Comparative curves of loss variation during training; **b** Comparative curves of Recall variation during training; **c** Comparative curves of mAP variation during training; **d** Comparative curves of $mAP_{50–95}$ variation during training

Table 3 Performance comparison of GRFS-YOLOv8 and other methods across different categories (Unit: %)

Method	Mandatory		Prohibitory		Warning	
	P	R	P	R	P	R
Faster R-CNN	27.4	42.6	25.7	42.2	26.0	60.1
YOLOv5	69.4	49.5	84.0	49.4	79.8	59.8
YOLOv7	50.5	43.7	76.3	38.3	66.2	58.8
YOLOv8	82.0	64.1	87.6	68.8	86.6	**80.5**
LF-YOLO	61.3	49.9	70.6	52.0	58.0	61.7
GRFS YOLOv8 (ours)	**82.8**	**64.2**	**90.8**	**72.8**	**88.8**	80.2

In the table, bold font indicates the optimal performance indicators, while font with underlining represents secondary performance indicators

Table 4 Performance comparison of GRFS-YOLOv8 and other methods in various complex environments (Unit: %)

Method	Foggy		Night		Snowy		Sunny	
	P	R	P	R	P	R	P	R
Faster R-CNN	26.0	48.1	26.1	48.1	26.1	48.1	26.4	48.6
YOLOv5	75.1	47.2	69.7	42.3	86.9	71.9	86.0	70.8
YOLOv6	82.5	62.9	**84.0**	67.6	94.0	**82.5**	92.3	**91.8**
YOLOv7	**89.5**	38.4	62.4	28.5	73.7	55.7	73.7	63.7
YOLOv8	71.5	59.9	79.3	65.8	90.4	65.6	92.2	86.7
LF-YOLO	85.3	42.8	71.9	37.5	75.0	72.2	78.0	71.3
GRFS YOLOv8 (ours)	**89.5**	**72.6**	81.0	**69.1**	**95.0**	72.2	**93.1**	87.0

In the table, bold font indicates the optimal performance indicators, while font with underlining represents secondary performance indicators

4.4.4 Experimental results of samples in complex environments

In practical applications, the performance of detectors can be influenced by varying weather conditions. Therefore, we conducted tests on the CCTSDB 2021 dataset under our weather conditions ('foggy', 'night', 'snowy', 'sunny') to assess detection performance, using an IoU threshold of 0.5. Table 4 displays the detector's Precision (P) and Recall (R) metrics under these weather conditions.

Notably, under 'foggy' weather conditions, while YOLOv7 achieves the same level of Precision as GRFS-YOLOv8, its Recall is 34.2% lower than that of GRFS-YOLOv8. This indicates that GRFS-YOLOv8 has a superior ability to identify positive samples compared to the baseline, showing a significant improvement with a 17.3% increase in Precision and a 12.9% increase in Recall. Under other weather conditions, GRFS-YOLOv8 significantly outperforms other methods in both Precision and Recall metrics.

4.4.5 Inference results of samples in complex environments

To visually illustrate the detection performance of GRFS-YOLOv8 in complex environments, Fig. 9 presents the comparison of the prediction results with the Baseline under different weather conditions. In the first, third, and fifth images of the figure, our method demonstrates a notable ability to identify more traffic signs compared to YOLOv8, with higher confidence levels. In the second and sixth images, both YOLOv8 and our method detect traffic signs, but our method exhibits approximately 10% higher confidence levels than YOLOv8. Notably, in the third image, our method not only detects a less prominent 'prohibitory' traffic sign, which YOLOv8 fails to identify, but also correctly identifies it, while YOLOv8 misidentifies a traffic signal.

4.5 Performance across different datasets

This section presents the performance of GRFS-YOLOv8 on the TT100K and GTSDB datasets, as shown in Tables 5 and 6 respectively. Table 5 illustrates that compared to the Baseline, GRFS-YOLOv8 demonstrates improvements across various metrics, although the magnitude of enhancement is moderate. Conversely, Table 6 shows more pronounced enhancements of GRFS-YOLOv8 compared to the Baseline, particularly in terms of precision. Despite slightly lower F1 scores compared to Improved YOLOv4, it displays higher Recall and precision rates. The performance improvements on both datasets demonstrate the strong generalization capabilities of our proposed algorithm.

5 Conclusion and future work

To address the challenges of low accuracy and missed detections for small traffic signs in complex backgrounds, we propose the GRFS-YOLOv8 algorithm for traffic sign detection. This algorithm not only addresses the limitations of the original algorithm in detecting small objects in complex environments but also reduces parameters by 45%. In this algorithm, the introduction of the GRF-SPPF module enhances the model's ability to handle background interference in complex environments, improving feature extraction and multi-scale fusion capabilities. The GhostConv and C2fGhost modules reduce computational costs while maintaining similar recognition performance. Additionally, the proposed SPANet effectively integrates more comprehensive semantic and positional information, enhancing the model's path aggregation capability and the feature fusion capacity within the neck network, consequently improving the detection ability of small targets. Ablation experiments on the CCTSDB 2021 dataset validate the positive impact of each enhancement module on the initial model. Comparative

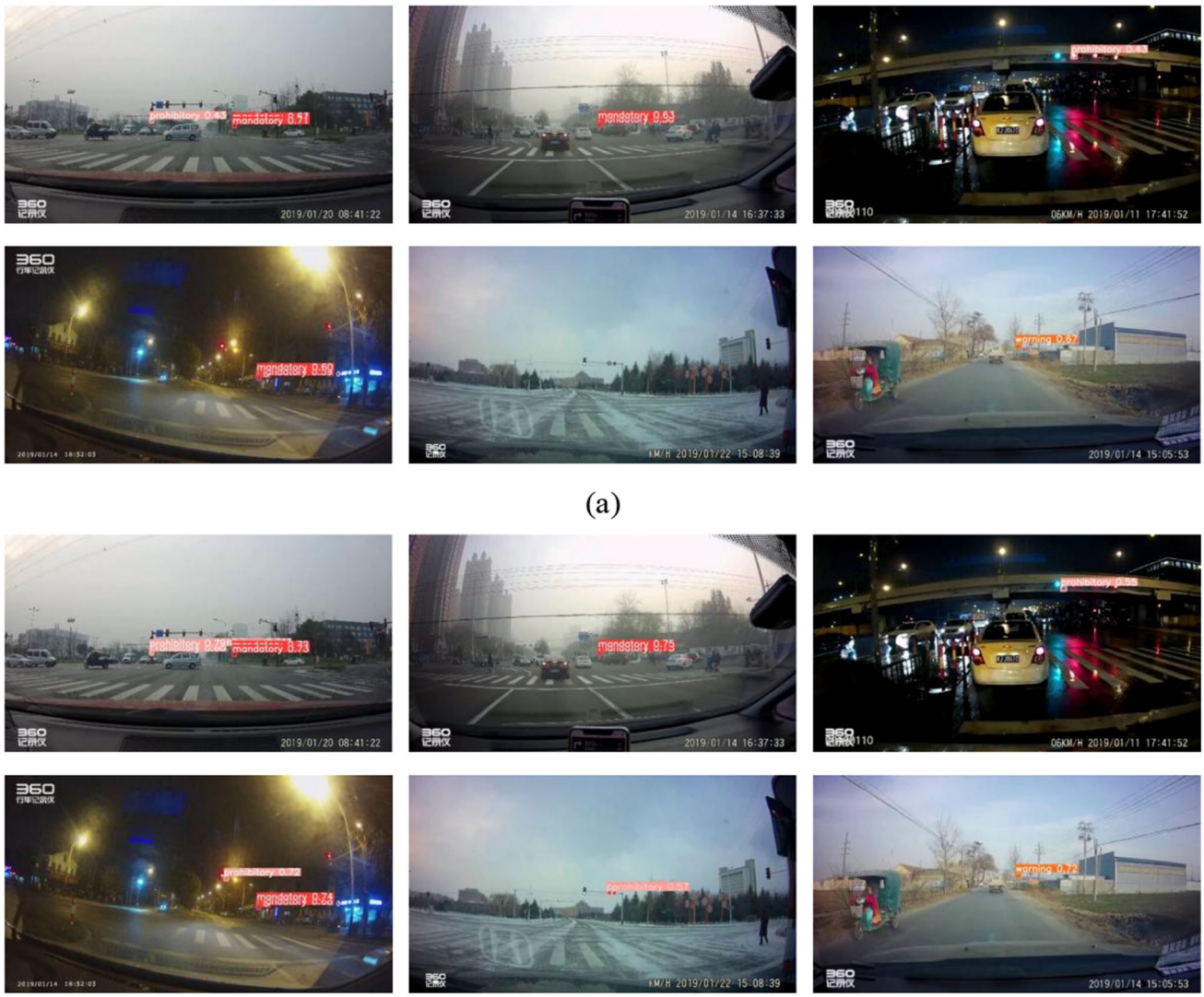

Fig. 9 Detection results of YOLOv8 (**a**) and GRFS-YOLOv8 (**b**) under various environmental and weather conditions, arranged from left to right as cloudy, foggy, night, rainy, snowy, and sunny

experiments on the CCTSDB 2021, TT100K, and GTSDB datasets demonstrate the advantages of GRFS-YOLOv8 in terms of accuracy and model size. In particular, test results under different weather conditions confirm the robustness of our method in complex environments.

Although GRFS-YOLOv8 enhances performance, issues of missed detections or false alarms persist in extreme scenarios (e.g., extensive occlusion, severe distortion). This may be due to misjudgments by the classification network regarding similar objects or backgrounds. Future work will focus on addressing these challenges. Additionally, we aim to further compress the model size while ensuring detection accuracy for easier deployment on mobile devices. This will involve more efficient model design and optimization to adapt to the resource constraints of mobile platforms.

Table 5 Performance of GRFS-YOLOv8 on the TT100K dataset (Unit: %)

Method	F1	R	mAP	mAP_{50-95}	Par (M)[a]/G[b]
YOLOv8	65.0	91.0	69.1	53.1	3.02/8.2
LF-YOLO [30]	37.0	47.8	38.6	25.7	7.35/16.8
TP-YOLO [31]	62.0	58.5	64.0	49.0	4.29/9.2
E-YOLOv4 Tiny [36]	–	–	54.4	–	18.20/–
LLTH YOLOv5 [34]	–	72.2	71.0	54.5	43.87/–
Anchor Free TSDetector [37]	–	–	65.1	–	81.34/95.0
Improved YOLOv5 [38]	–	–	65.1	–	8.04/17.9
YOLOv5 + AFPN + Multi head [39]	–	–	68.1	–	7.42/–
DW YOLOv7 tiny [35]	–	–	51.2	–	5.20/11.8
CR YOLOv8 [40]	–	–	65.1	–	14.60/–
GRFS YOLOv8(ours)	**67.0**	**95.0**	**71.2**	**54.8**	1.71/10.9

In the table, bold font indicates the optimal performance indicators, while font with underlining represents secondary performance indicators. The symbol "–" indicates data not provided by the author
[a]Par indicates the number of parameters
[b]G indicates GFLOPs

Table 6 Performance of GRFS-YOLOv8 on the GTSDB dataset (Unit: %)

Method	F1	R	mAP	mAP_{50-95}	Par (M)[a]/G[b]
YOLOv8	89.0	93.0	91.3	74.5	3.01/8.2
YOLOv4-tiny + AFPN + RFB [41]	87.6	82.4	86.8	–	– /–
Improved YOLOv4 [42]	**93.5**	95.5	91.7	–	– /–
YOLOv7-tiny [43]	–	–	93.5	74.9	23.29/–
YOLOv5s-A2 [44]	–	90.5	94.1	–	7.90/–
LF-YOLO [30]	83.0	81.6	86.5	62.1	7.25/16.3
TP-YOLO [31]	88.0	82.1	90.4	72.2	4.28/9.2
Anchor Free TSDetector [37]	–	94.7	93.4	–	81.34/95.0
SPD YOLOv5s [45]	–	91.6	**95.4**	–	24.30/–
GRFS YOLOv8(ours)	92.0	**96.0**	94.0	**78.9**	1.65/9.9

In the table, bold font indicates the optimal performance indicators, while font with underlining represents secondary performance indicators. The symbol "–" indicates data not provided by the author
[a]Par indicates the number of parameters
[b]G indicates GFLOPs

Author contributions The submission has been received explicitly from all co-authors. And authors whose names appear on the submission have contributed sufficiently to the scientific work and therefore share collective responsibility and accountability for the results.

Funding This work is supported by the National Natural Science Foundation of China (62002070), the Science and Technology Plan Project of Guangzhou City (202102021236).

Availability of data and materials Data will be made available on reasonable request.

Code availability Code availability not applicable.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

1. Soendoro, D., Supriana, I.: Traffic sign recognition with color-based method, shape-arc estimation and SVM. In: Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, pp. 1–6 (2011). https://doi.org/10.1109/ICEEI.2011.6021584
2. Khongviriyakit, N., Paripurana, S.: Traffic sign detection based on color and boundary shape box ratio. In: 2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTICON), pp. 461–464 (2018). https://doi.org/10.1109/ECTICon.2018.8620017
3. Garcí-Garrido, M.A., et al.: Complete vision-based traffic sign recognition supported by an I2V communication system. Sensors **12**(2), 1148–1169 (2012)
4. Li, J., Wang, Z.: Real-time traffic sign recognition based on efficient CNNs in the wild. IEEE Trans. Intell. Transport. Syst. **20**(3), 975–984 (2019). https://doi.org/10.1109/TITS.2018.2843815
5. Tabernik, D., Skočaj, D.: Deep learning for large-scale traffic-sign detection and recognition. IEEE Trans. Intell. Transport. Syst. **21**(4), 1427–1440 (2020). https://doi.org/10.1109/TITS.2019.2913588
6. Yan, Yi., et al.: A traffic sign recognition method under complex illumination conditions. IEEE Access **11**, 39185–39196 (2023). https://doi.org/10.1109/ACCESS.2023.3266825
7. Liu, W., 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, pp. 21–37 (2016). Springer
8. Yu, J., Ye, X., Tu, Q.: Traffic sign detection and recognition in multiimages using a fusion model with YOLO and VGG network. IEEE Trans. Intell. Transport. Syst. **23**(9), 16632–16642 (2022). https://doi.org/10.1109/TITS.2022.3170354
9. Shi, Y., Li, X., Chen, M.: SC-YOLO: a object detection modelfor small traffic signs. IEEE Access **11**, 11500–11510 (2023). https://doi.org/10.1109/ACCESS.2023.3241234
10. Luo, S., Chenghang, Wu., Li, L.: Detection and recognition of obscured traffic signs during vehicle movement. IEEE Access **11**, 122516–122525 (2023). https://doi.org/10.1109/ACCESS.2023.3329068
11. Soylu, E., Soylu, T.: A performance comparison of YOLOv8 models for traffic sign detection in the Robotaxi-full scale autonomous vehicle competition. Multimed. Tools Appl. **8**, 1–31 (2023)
12. Gray, N., et al.: GLARE: a dataset for traffic sign detection in sun glare. IEEE Trans. Intell. Transport. Syst. **24**(11), 12323–12330 (2023). https://doi.org/10.1109/TITS.2023.3294411
13. Liu, S., et al.: Path aggregation network for instance segmentation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8759–8768 (2018). https://doi.org/10.1109/CVPR.2018.00913
14. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions (2015). arXiv preprint arXiv:1511.07122
15. Tan, M., Pang, R., Le, Q.V.: Efficientdet: scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10781–10790 (2020)
16. Han, K., et al.: Ghostnet: more features from cheap operations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1580–1589 (2020)
17. Zhang, J.M., et al.: "Cctsdb 2021: a more comprehensive traffic sign detection benchmark", human-centric comput. Inf. Sci. **12**, 55 (2022)
18. Zhu, Z., et al.: Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2110–2118 (2016)
19. Houben, S., et al.: Detection of traffic signs in real-world images: the German traffic sign detection benchmark. In: The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2013)
20. Ren, S., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. Adv. Neural. Inf. Process. Syst. **5**, 28 (2015)
21. Lin, T.Y., et al.: Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007 (2017). https://doi.org/10.1109/ICCV.2017.324
22. Jocher, G.: Ultralytics/yolov5: v3.1-bug fixes and performance improvements. https://github.com/ultralytics/yolov5. Version v3.1. Oct. 2020. https://doi.org/10.5281/zenodo.4154370. https://doi.org/10.5281/zenodo.4154370
23. Li, C., et al.: YOLOv6: a single-stage object detection framework for industrial applications (2022). arXiv:2209.02976
24. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors (2022). arXiv:2207.02696
25. Ge, Z., et al.: Yolox: exceeding yolo series in 2021 (2021). arXiv preprint arXiv:2107.08430
26. Cao, L., Kang, S., Chen, J.: Improved lightweight YOLOv5s algorithm for traffic sign recognition. In: 2023 3rd International Symposium on Computer Technology and Information Science (ISCTIS), pp. 289–294. IEEE (2023)
27. Bai, W., et al.: Two novel models for traffic sign detection based on YOLOv5s. Axioms **12**(2), 160 (2023)
28. Tang, S., Fang, Y., Zhang, S.: HIC-YOLOv5: improved YOLOv5 for small object detection (2023). arXiv preprint arXiv:2309.16393
29. Xie, Z.Y., Li, T.J.: Traffic sign detection based on depth improved YOLO-V5. In: 2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP), pp. 1896–1899 (2023). https://doi.org/10.1109/ICSP58490.2023.10248695
30. Liu, M., et al.: LF-YOLO: a lighter and faster YOLO for weld defect detectionof X-ray image. IEEE Sens. J. **23**(7), 7430–7439 (2023). https://doi.org/10.1109/JSEN.2023.3247006
31. Di, Y., et al.: TP-YOLO: a lightweight attention-based architecture for tiny pest detection. In: 2023 IEEE International Conference on Image Processing (ICIP), pp. 3394–3398 (2023). https://doi.org/10.1109/ICIP49359.2023.10222202
32. Wang, X., et al.: C2Net-YOLOv5: a bidirectional Res2Net-based traffic sign detection algorithm (2023). Available at SSRN 4406700
33. Wang, Q., et al.: Real time traffic sign recognition algorithm based on SG-YOLO. In: Asian Simulation Conference, pp. 86–99. Springer (2022)
34. Sun, X., et al.: LLTH-YOLOv5: a real-time traffic sign detection algorithm for low-light scenes. Automot. Innov. **7**(1), 121–137 (2024)
35. Jia, Z., Sun, S., Liu, G.: Real-time traffic sign detection based on weighted attention and model refinement. Neural. Process. Lett. **55**(6), 7511–7527 (2023)
36. Xiao, Y., et al.: E-YOLOv4-tiny: a traffic sign detection algorithm for urban road scenarios. Front. Neurorobotics **17**, 34 (2023)
37. Zhang, J., et al.: A robust real-time anchor-free traffic sign detector with one-level feature. IEEE Trans. Emerg. Top. Comput. Intell. **8**, 24 (2024)
38. Wang, J., et al.: Improved YOLOv5 network for real-time multi-scale traffic sign detection (2021). arXiv preprint arXiv:2112.08782
39. Wang, J., et al.: A lightweight vehicle mounted multi-scale traffic sign detector using attention fusion pyramid. J. Supercomput. **80**(3), 3360–3381 (2024)

40. Zhang, L.J., et al.: CR-YOLOv8: multiscale object detection in traffic sign images. IEEE Access **12**, 219–228 (2023)
41. Yao, Y., et al.: Traffic sign detection algorithm based on improved YOLOv4-Tiny. Signal Process. Image Commun. **107**, 116783 (2022)
42. Wang, H., Yu, H.: Traffic sign detection algorithm based on improved YOLOv4. In: 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), vol. 9, pp. 1946–1950 (2020). https://doi.org/10.1109/ITAIC49862.2020.9339181
43. She, F., et al.: Improved traffic sign detection model based on YOLOv7-Tiny. IEEE Access **11**, 126555–126567 (2023). https://doi.org/10.1109/ACCESS.2023.3331426
44. Yuan, Xu., et al.: Faster light detection algorithm of traffic signs based on YOLOv5s-A2. IEEE Access **11**, 19395–19404 (2023). https://doi.org/10.1109/ACCESS.2022.3204818
45. Han, T., Sun, L., Dong, Q.: An improved YOLO model for traffic signs small target image detection. Appl. Sci. **13**(15), 8754 (2023)