Signal, Image and Video Processing (2024) 18:4821–4829
https://doi.org/10.1007/s11760-024-03118-z

ORIGINAL PAPER

A lightweight algorithm for small traffic sign detection based on improved YOLOv5s

Kunhui Cai[1,2] · Jingmin Yang[1,2,3] · Jinghui Ren[1,2] · Wenjie Zhang[1,2]

Received: 18 November 2023 / Revised: 24 February 2024 / Accepted: 26 February 2024 / Published online: 6 April 2024

Abstract
With the rise of deep learning technology, significant progress has been made in object detection. Traffic sign detection is a research hotspot for object detection tasks. However, due to small size of traffic signs, there is room for further improvement in the comprehensive performance of the existing technology. In this paper, we propose a lightweight network based on yolov5s to achieve real-time localization and classification of small traffic signs. First, we improve the bottleneck transformers with 3 convolution (Bot3) module to enhance the backbone network's ability to extract features from small targets, improving the accuracy while reducing the number of parameters and giga floating-point operations per second (GFLOPs). Second, we introduce ghost convolution (GhostConv) to obtain redundant feature maps with cheap operations to further improve the model's efficiency. Finally, we use soft non-maximum suppression (Soft-NMS) in the detection phase to improve the model accuracy again without additional computational overhead for training. According to the tests on the Tsinghua-Tencent 100 K (TT100K) dataset, the proposed method outperforms the original YOLOv5s in small traffic sign detection, with an increase of 8.7% in mAP_{50}, a reduction of 22.5% in parameter count, and a 17.2% reduction in computational complexity.

Keywords Traffic sign detection · Small object · Lightweight network · YOLOv5s

1 Introduction

The detection of traffic signs is crucial in computer vision as it provides essential real-time road traffic information to drivers, contributing to safer driving and reduced traffic accidents. This has led to increased attention and research from scholars in the field of object detection.

Traffic sign detection can be broadly categorized into traditional machine learning-based methods and deep learning-based methods. Traditional methods rely on color and shape for feature extraction [1–3], which is labor-intensive and often yields weak detection capabilities. On the other hand, deep learning-based methods are divided into two categories: two-stage and one-stage. While two-stage algorithms offer high recognition accuracy, they are slow and unsuitable for real-time detection, making one-stage detection more suitable for traffic sign detection due to its higher real-time requirements.

The YOLO series [4–11], particularly YOLOv5 [8], is a prominent model in object detection known for its accuracy and fast detection speed. However, its performance is better suited for large objects, presenting limitations in detecting small objects such as traffic signs. Given the small size and the need for fast detection, satisfactory results in traffic sign detection using YOLOv5 are still challenging to achieve.

In summary, existing small traffic sign detection algorithms do not strike a balance between accuracy and lightweight and are difficult to apply in real-world scenarios. In this research, we propose a new algorithm for small traffic sign detection based on the current excellent lightweight YOLOv5s. The proposed method enhances detection accuracy while reducing the size of the model parameters, which leads to better application to real-world scenarios. In this paper, our contributions are as follows:

Jingmin Yang, Jinghui Ren and Wenjie Zhang have contributed equally to this work

✉ Jingmin Yang
yjm1758@mnnu.edu.cn

[1] School of Computer Science, Minnan Normal University, Zhangzhou 363000, Fujian, China

[2] Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou 363000, Fujian, China

[3] Department of Electronic Engineering, Taipei University of Technology, Taipei, Taiwan, China

- We have improved the BoT3 module to improve the image feature extraction through Multi-Head Self-Attention (MHSA), which both improves the detection ability of the model and reduces the number of parameters.
- To achieve model lightweight, we use GhostConv instead of standard convolution, reducing the size and computation of the model.
- We use Soft-NMS in the detection phase to further improve the detection performance without adding parameter counts.
- The method proposed in this article was trained and evaluated on the TT100K dataset. The experimental results demonstrate the effectiveness of the proposed network improvement method.

The remainder of this paper is organized as follows: Section 2 presents an overview of the related work on traffic sign detection. Section 3 outlines the implementation process of the method proposed in this paper. Section 4 delves into the experimental results and their analysis. Finally, Sect. 5 provides the conclusion of this paper.

2 Related works

In recent years, deep learning has become a mainstream method with superior performance in various tasks. The current object detection methods based on convolutional neural networks (CNN) have achieved good results on public data (e.g., Pascal VOC, MS COCO), and many researchers have started applying these methods to detect traffic signs.

2.1 Traffic sign detection

Object detection methods based on CNN can be broadly categorized into two groups: regression-based and region-proposal based [12]. With the development of these methods, they are also used to apply in traffic sign detection.

Due to the problem of small pixel share of traffic signs in photos, Zhang et al. proposed cascaded R-CNN networks to detect small traffic signs by fusing multi-scale features [13]. Liang et al. proposed an improved multi-scale sparse R-CNN algorithm to improve the network's focus on the target region [14]. Li et al. enhanced contextual information exchange based on Faster R-CNN using attention-improving feature pyramids [15]. However, these region proposal-based methods suffer from slow detection speed and large model sizes.

Due to the faster detection speed of regression-based algorithms, many scholars have researched this method. Hu et al. proposed a parallel deformable convolutional module applied on YOLOv5 to increase the feature extraction capability of the deep network [16]. Wang et al. proposed an improved feature pyramid model to enhance the fusion of multi-scale information in the model [17]. Chen et al. located the position of the traffic signs more accurately by refining the grid of the detector head [18]. The regression-based method requires only one reading of the image to identify the category and position of the objects in the image, meeting the real-time requirements of traffic object detection.

2.2 Small object detection

Detecting small objects has always posed a challenge in object detection. Relative to regular-sized objects, small objects usually lack sufficient appearance information, posing challenges in recognizing them from the background or similar objects. Currently, data augmentation, multi-scale fusion, anchor-free mechanisms, and attention mechanisms are the leading solutions for small object detection. Data augmentation is increasing the sample size to enhance the precision of detecting objects of smaller size without substantially expanding the data [19–21]. Multi-scale feature map fusion technology enriches features by combining shallow position information and profound contextual information in the network [22–24]. Currently, the mainstream detection models have shifted to the anchor-free mechanism [25–27], which eliminates the need to set anchor hyperparameters in advance and avoids many redundant boxes, allowing for flexible prediction of objects of varying sizes.

Since introducing the self-attention mechanism, many researchers have adopted it as a solution for small object detection due to its excellent performance [28–31]. Wang et al. proposed BoTNet [32], which replaces the 3×3 convolution of ResNet Bottleneck with MHSA, which leads to excellent outcomes in the object detection task. It is worth noting that the detection accuracy significantly improves when the transformer self-attention mechanism is incorporated into the deep network structure. This inspires us to improve YOLOv5s in this paper.

2.3 Model lightweight

Lightweighting of the model is another difficulty in traffic sign detection. Network pruning, knowledge distillation, and efficient network structure design play an important role in lightweighting. Network pruning is the process of reducing the model size by removing redundant connections and parameters from the network without compromising accuracy or sacrificing a small degree of accuracy [33–35]. Knowledge distillation is the use of predictions from complex teacher models as object labels to improve the performance of the student models and reduce the number of parameters in the model [36–38].

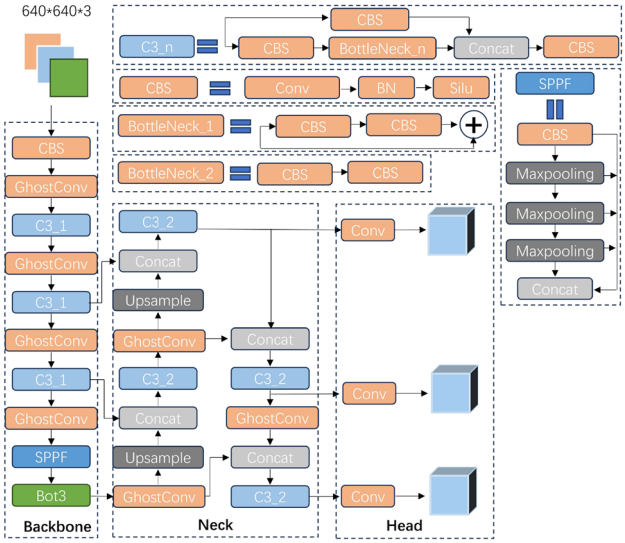


Fig. 1 Architecture of our proposed method

In addition, efficient network structure design is key to lightweight. For example, in the ShuffleNet [39, 40] and the GhostNet series [41, 42], designing a lightweight network structure suitable for a specific task reduces the complexity of the model and the number of parameters while maintaining high accuracy. Among them, GhostNet's GhostConv replaces the standard convolutional approach by combining a small number of convolutional kernels with an inexpensive linear change operation, thus effectively reducing the demand for computational resources without compromising model performance.

3 Methods

In this section, we introduce the proposed YOLOv5s based model. Figure 1 shows the structure of the proposed model, which can be divided into three parts: backbone responsible for feature extraction, neck responsible for feature fusion and head responsible for prediction. In the backbone of our network, we use a Bot3 module to extract the deepest feature maps using MHSA. In addition, GhostConv is used to thin the model.

3.1 A backbone network with multi-head attention

In recent years, visual self-attention has shown remarkable advancements across various computer vision tasks. The mechanism allows the model to focus more on critical features of the target object, thereby enhancing detection accuracy and robustness. However, introducing additional attention mechanisms into CNN inevitably increases computational cost.

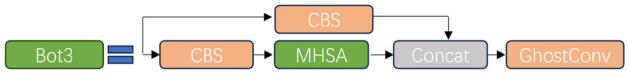


Fig. 2 The structure of the Bot3 module. MHSA refers to Multi-Head Self-Attention

Bottleneck transformer [32] addresses this dilemma by providing excellent performance while balancing computational costs. In our work, the last layer of the original backbone network is replaced with an improved Bot3 module, which improves the model's feature extraction capability and optimizes the network's global information relationship. In Bot3 Block, the MHSA captures long-range dependencies with less computation. The MHSA can process the input feature map with multiple heads, allowing the model to attend to information in different dimensions of the semantic space. The operation of MHSA is as follows:

$$MHSA(Q, K, V) = Concat(head_0, head_1, \ldots, head_H) \tag{1}$$

where Q, K, and V denote three linear layers used for computing queries, keys, and values in a standard self-attention task. The head represents the head of the self-attention mechanism. Concat denotes the concatenation operation. The operation of the head is as follows:

$$head_i = attention(Q_i, K_i, V_i) \quad i \in [0, H-1] \tag{2}$$

$$attention(Q_i, K_i, V_i) = Softmax\left(Q_i K_i^T + qr^T\right) \times V_i \tag{3}$$

$$qr = (R_h + R_w) \times Q, \tag{4}$$

where R_h and R_w represent height and width relative position codes, respectively. Attention means a standard self-attention operation. Figure 2 presents the structure of the Bot3 Block.

3.2 Ghost convolution instead of standard convolution

The conventional feature extraction approach for deep CNN utilizes multiple convolutional kernels to carry out convolutional mapping operations across all channels within the input feature map. However, stacking many convolutional layers in deep networks produces rich or even redundant feature maps and requires a huge amount of parametric and computational effort. Han et al. believe that similar information in feature maps can provide models with a more comprehensive understanding of input data [41]. Therefore, they proposed GhostConv to obtain redundant feature maps with a cheaper operation.

Figure 3 illustrates the GhostConv mechanism. First, perform a convolution operation on the input data with a channel count of half of the target. Secondly, it is subjected to depth-

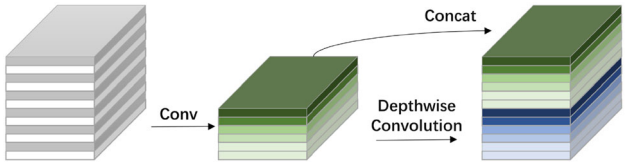


Fig. 3 An illustration of the GhostConv

wise separable convolution and finally stitched together. The computational effort for standard convolution is

$$FLOPs_{conv} = h \times w \times c' \times k \times k \times c \tag{5}$$

where h and w refer to the dimensions of the output feature map. c and c' represent the number of input and output channels, respectively. $k * k$ represents the size of the standard convolutional kernel. The computational complexity of GhostConv is

$$FLOPs_{GC} = FLOPs_{GC1} + FLOPs_{GC2} \tag{6}$$

$$FLOPs_{GC1} = h \times w \times \frac{c'}{2} \times k \times k \times c \tag{7}$$

$$FLOPs_{GC2} = \left(h \times w \times \frac{c'}{2}\right) \times \left(d \times d + \frac{c'}{2}\right)^{\leftarrow} \tag{8}$$

where $FLOPs_{GC1}$ and $FLOPs_{GC2}$ represent the computational complexity of GhostConv's standard convolution and depthwise separable convolution, respectively. $d * d$ represents the size of the depthwise separable convolutional kernel. The theoretical computational compression ratio for upgrading standard convolutions using GhostConv is

$$r_c = \frac{FLOPs_{GC}}{FLOPs_{conv}} \approx \frac{1}{2} \tag{9}$$

3.3 Soft-NMS

The post-processing step in the object detection process is NMS, mainly used to suppress redundant boxes in a set of detection boxes to obtain a window for optimal detection. Initially, the detection boxes are sorted based on their scores, and subsequently, the box with the highest score is preserved, while redundant boxes with an overlap area exceeding a specific percentage of that box are eliminated. However, if two objects are within a preset overlap threshold, it may result in not detecting that object to be detected. In other words, the box with the lower score is discarded when two object boxes overlap, owing to its excessively large overlap area.

Soft-NMS [43] decreases the scores of the detection boxes that overlap with the highest scoring box rather than removing them outright. The detection box with the highest score is first selected. Then, the score is attenuated for the detection boxes whose overlapping region is more significant than a preset threshold. Then, the box is removed from the set of detection boxes. After that, the highest-scoring detection box within the set is processed repeatedly until the set of detection boxes becomes empty, and finally, the detection boxes with confidence less than the set threshold are removed. Soft-NMS operates as follows:

$$S_i = S_i e^{-\frac{iou(M,b_i)^2}{\sigma}} \quad \forall b_i \notin D \tag{10}$$

where S_i is the confidence score, M is the highest-scoring suggestion box, b is the set of suggestion boxes, D is the set of outputs, and σ is the Gaussian penalty function. In addition, Soft-NMS does not require additional training and has the same algorithmic complexity as traditional NMS, making it efficient to use. Therefore, we chose to use Soft-NMS only in the testing phase.

4 Experiment

4.1 Dataset and experimental environment

In this paper, the TT100K [44] dataset is selected as the experimental object. The image size of this dataset is 2048 × 2048, encompassing diverse lighting and weather conditions, with traffic sign dimensions ranging from 8 × 8 to 400 × 400, which is about 0.001% to 4% of the whole picture. These attributes make the TT100K dataset more suitable for small traffic sign recognition tasks. To mitigate the issue of inadequate samples, we dropped the categories with less than 100 traffic signs, finally leaving 45 categories. And the dataset is partitioned randomly into training data, validation data, and test data at a ratio of 7:2:1.

We trained and tested on a Windows PC with an Inter-core i5 12400f CPU, NVIDIA GeForce GTX2080ti GPU, and 32GB of RAM. In addition, we primarily utilized Python 3.8, OpenCV, Pytorch, and other required libraries to implement our model. We set the training epochs to 300, batch size to 32, and the size of the input image was 640 × 640.

4.2 Evaluation metrics

Samples of results in object detection can be broadly classified into three categories. True positive (TP) denotes correctly detected targets, false negative (FN) denotes non-detected objects, and false positive (FP) denotes incorrectly detected objects. This experiment mainly uses mAP (mean average precision) to evaluate the performance. Precision (P) is used to assess the percentage of correct predictions in the results. Recall (R) is used to assess how many positive samples were

correctly detected. These two criteria are defined below:

$$P = \frac{TP}{TP + FP} \tag{11}$$

$$R = \frac{\mathrm{TP}}{TP + FN} \tag{12}$$

mAP is a commonly utilized metric for evaluating object detectors, quantifying an object detector's precision and recall at different intersections over union (IoU) thresholds to gauge its accuracy. IoU gauges the overlap between the predicted box and the ground truth box.

$$AP = \sum_{n} (R_n - R_{n-1})\, P_n \tag{13}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{14}$$

In addition, we are measuring the speed and efficiency of the object detector by parameters, GFLOPs, and FPS. Lower values of the first two metrics indicate higher model efficiency and higher FPS values indicate faster inference of the model.

4.3 Result

In order to validate the effectiveness of the model put forward in this paper, the detection outcomes of various mainstream object detection networks are replicated on the enhanced TT100K dataset and contrasted with YOLOv5s and our model. The results of the comparison are shown in Table1. It is obvious to see that our model achieves good results in the synthesis of Params, GFLOPs, mAP_{50}, and FPS. The Params and GFLOPs of our model are only 30%, 15% and 50% of those of YOLOv6s, YOLOv7 and YOLOv8s, respectively.

Table2 presents the mAP_{50} results of our model and other models across different classes on the enhanced TT100K dataset. These signs fall into the following three types: warning, prohibition, and compulsory. Of these, signs beginning with "w" are classified under the warning category, signs beginning with "p" signify prohibition, and signs beginning with "i" indicate compulsion. The maximum mAP_{50} of our model is 99.4%, while YOLOv5s, YOLOv7 and YOLOv8s can achieve maximum mAP_{50} of 98.4%, 96.8% and 99%. Our model attains a higher maximum mAP_{50} compared to other models. In addition, our model obtains 42 categories with mAP_{50} greater than 60.0%, accounting for 93.3% of the total number of categories, while YOLOv5s, YOLOv7, and YOLOv8s account for 77.8%, 55.6%, and 97.8%, respectively. For certain categories with relatively small icons, such as pn, pne, and pg, the detection performance of our model outperforms the other models.

Table 1 The recognition results of different methods on TT100K dataset

Method	Params(M)	GFLOPs	mAP_{50} (%)	FPS
Faster-RCNN	–	–	56.3	2
YOLOv5s	7.1	16.3	71.4	**256**
YOLOv6s	17.2	44.1	84.6	207
YOLOv7	37.4	105.9	58.3	68
YOLOv8s	11.2	28.7	**86.1**	238
Our	**5.5**	**13.5**	80.1	161

Input image size is 640 × 640
Bold values indicate the best performing values

4.4 Ablation experiment

In this section, we perform ablation experiments on the enhanced TT100K dataset to verify the impact of the model components on the final performance. We chose original YOLOv5s as the baseline. As shown in Table 3, mAP_{50}, Params, GFLOPs, and FPS were used as indicators and compared. In comparison with the baseline, our model exhibits an 8.7% improvement in mAP_{50}, along with a 22.5% decrease in the number of parameters and a roughly 17.2% reduction in computational load, as well as a decrease in the speedup (from 256 to 161 FPS), which is within acceptable limits. In short, our model enhances detection accuracy while reducing its weight. Although the FPS decreases slightly, it maintains a high frame rate and enables real-time detection.

Each module used in this paper improves the comprehensive performance of the model. Compared to YOLOv5s, the mAP_{50} of the improved Bot3 module increases from 71.4 to 79.4%, while the number of parameters and computation decreases by 5.6% and 2.4%, respectively. This indicates that MHSA can better utilize the detailed information of traffic signs in images, thereby more accurately detecting small traffic signs. When applying GhostConv to our model, it decreases the mAP_{50} by 1.5% but also decreases the parameters count by 17.9% and the computation workload by 15.1%. This shows that GhostConv can effectively achieve a lighter model without losing too many features of the smaller objects. When Soft-NMS is used as a post-processing, the mAP_{50} increases by 2.2%, while the FPS decreases from 256 to 161. This shows that using Soft-NMS in the detection phase helps to increase the sensitivity to small target positional deviations, thus improving the inference accuracy but increasing the inference time.

For further validation of the improved Bot3 module's effectiveness, we conduct experiments on the YOLOv5s model. We substituted the convolutional layers at various positions within the Bot3 module with GhostConv. The detailed configuration is shown in Fig. 4. The results are shown in Table 4.

Table 2 Comparison of mAP_{50} for each class in TT100K dataset

Method	Total (%)	pl80 (%)	p6 (%)	p5 (%)	pm55 (%)	pl60 (%)	ip (%)	p11 (%)	i2r (%)	p23 (%)
YOLOv5s	71.4	72.9	54.8	90.9	61	68.1	92.4	75.3	81.3	86.5
YOLOv7	58.3	38.9	26.9	68.1	27.3	23.3	90.3	64.9	76.8	77.6
YOLOv8s	**86.1**	83.1	81.3	96	91.6	83.1	94.8	86.5	87.7	97.3
our	80.1	**83.5**	67.2	87.1	68	63.2	93.2	81.6	85	66

Method	pg (%)	il80 (%)	ph4 (%)	i4 (%)	pl70 (%)	pne (%)	ph4.5 (%)	p12 (%)	p3 (%)	pl5 (%)
YOLOv5s	76.6	86	56.9	89.5	35.1	93.7	64.8	51.3	81.4	70.6
YOLOv7	84.7	90	15.5	85.5	12.2	92.4	54.4	24.3	68.8	45.6
YOLOv8s	91.9	98.7	84.8	93.3	58.7	92.3	75.9	85.9	80.5	86.8
Our	**96.4**	93.6	**94.6**	**94.5**	58.4	**96.7**	**82.2**	62.2	**87.8**	82.3

Method	w13 (%)	i4l (%)	pl30 (%)	p10 (%)	pn (%)	w55 (%)	p26 (%)	p13 (%)	pr40 (%)	pl20 (%)
YOLOv5s	31.9	87.3	71	59.2	93.4	82.2	77.7	73.8	96.7	36.2
YOLOv7	54.8	84.8	32	32.5	93.7	70.4	67.8	75	58.3	33.6
YOLOv8s	79.2	89.2	82.5	86.6	92.1	82.4	89.7	87.6	98.9	75.1
Our	77.8	86.9	79.9	59.6	**94.4**	77.5	80.7	79.7	80.9	**82.4**

Method	pm30 (%)	pl40 (%)	i2 (%)	pl120 (%)	w32 (%)	ph5 (%)	il60 (%)	w57 (%)	pl100 (%)	w59 (%)
YOLOv5s	23.8	68.9	77.8	77.4	66.2	60.5	98.4	80.1	87.8	63.8
YOLOv7	18.1	37.4	68.6	72.4	70.2	33.4	96.8	61.8	83.5	65.8
YOLOv8s	60.4	81.3	80.6	97.4	84.3	72	99	94.2	96.5	77.6
Our	54.7	73.3	**86.4**	75.9	**87.3**	69.8	**99.4**	83.2	87.8	**81**

Method	il100 (%)	p19 (%)	pm20 (%)	i5 (%)	p27 (%)	pl50 (%)
YOLOv5s	95.7	31.1	47.3	95.2	66.8	71.8
YOLOv7	92.4	28.9	19.9	90.7	69.2	42.8
YOLOv8s	94.7	82.1	81.9	89.6	89.1	81.1
Our	87.3	71.3	60.9	**96.1**	67.7	80.5

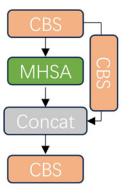
Bold values indicate the best performing values

Table 3 Ablation studies of the proposed Bot3 module

model	Params(M)	mAP_{50} (%)	GFLOPs	FPS
(1):YOLOv5s	7.1	71.4	16.3	256
(2):(1)+Bot3	6.7	79.4	15.9	256
(3):(2)+GhostConv	**5.5**	77.9	**13.5**	256
(4):(3)+Soft-NMS	**5.5**	**80.1**	**13.5**	161

Bold values indicate the best performing values

The best result is Table 4c, with a 1.1% improvement in mAP_{50} and a 15% decrease in the number of parameters compared to the original module. The lowest number of parameters is Fig. 4f, but the difference between the two mAP_{50} and the number of parameters is 1.4% and 15% compared to Fig. 4c, so we choose the structure with higher accuracy.



GhostConv position					
✓			✓		✓
	✓		✓	✓	✓
		✓		✓	✓
(a)	(b)	(c)	(d)	(e)	(f)

Fig. 4 Planned GhostConv Bot3.✓ is the position we replace CBS by GhostConv

Table 4 Ablation study on planned GhostConcatenation model

Model	mAP_{50} (%)	Parameter (M)
Base(Bot-3)	78.9	0.86
Fig. 4a	78.3	0.79
Fig. 4b	78.1	0.79
Fig. 4c	**79.4**	0.73
Fig. 4d	78.5	0.73
Fig. 4e	78.7	0.67
Fig. 4f	78	**0.60**

Bold values indicate the best performing values

Fig. 5 Comparison of detection results between YOLOv5s and our model. **a** Represents the original image, **b** represents the detection result of YOLOv5s, and **c** represents the detection result of our algorithm

4.5 Visualization

In order to show the detection ability of our model, we present visualizations of the results. As shown in Fig. 5, (a) represents the original image, (b) and (c) represent the detection results of YOLOv5s and our model, respectively. We zoomed in on the detection results of the images for ease of observation. It can be seen that our model has higher recall than YOLOv5s because our model detects objects that YOLOv5s do not detect. In addition, our model enhances the precision of small object detection and surpasses YOLOv5s in identifying distant and smaller traffic signs, which is beneficial for tasks related to detecting small traffic signs and ensuring autonomous safety.

5 Conclusion

In this paper, we proposed a lightweight network based on YOLOv5s aimed at tackling the challenges related to traffic sign detection. First, we improve the feature extraction module using multi-head attention to reduce redundant parameters and computation and improve the feature representation of small objects through the improved Bot3 module. Secondly, by constructing an efficient network through GhostConv, generating redundant feature maps through linear computation will significantly decrease both the quantity of parameters and computational load, achieving a lightweight network. Finally, experimental results on the TT100K dataset show that the approach strikes a balance between accuracy and efficiency and excels in detecting small traffic signs. In future work, we will focus on the impact of various post-processing techniques on model performance and design models with better overall performance.

Author Contributions KC was involved in contributed to conceptualization, methodology and writing-original draft preparation. JY contributed to conceptualization, funding, supervision, writing—reviewing and editing. JR contributed to data curation, validation and visualization. WZ was involved in supervision writing—review and editing.

Data availability Data will be made available on reasonable request

Declarations

This work was supported by the Fujian Province Nature Science Foundation under Grant Nos. 2020J01813 and 2021J011002, the Research Project on Education and Teaching Reform of Undergraduate Colleges and Universities in Fujian Province under Grant Nos. FBJG20210070 and FBJY20230170, and the 2022 Annual Project of the Fourteenth Five-Year Plan for Fujian Educational Science under Grant No. FJJKBK22-173.

References

1. Benallal, M., Meunier, J.: Real-time color segmentation of road signs. In: CCECE 2003-Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology (Cat. No. 03CH37436). IEEE, vol. 3, pp. 1823–1826 (2003)
2. Kuo, W.-J., Lin, C.-C.: Two-stage road sign detection and recognition. In: 2007 IEEE International Conference on multimedia and expo. IEEE, pp. 1427–1430 (2007)
3. Liu, H.X., Ran, B.: Vision-based stop sign detection and recognition system for intelligent vehicles. Transp. Res. Rec. **1748**(1), 161–166 (2001)
4. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788 (2016)

5. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263–7271 (2017)
6. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
7. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: Yolov4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
8. Jocher., G, Stoken., A, Chaurasia., A, Borovec., J, Kwon., Y, Michael., K, Changyu., L, Fang., J, Skalski., P, Hogan., A, Nadar., J.: Ultralytics/yolov5: V6.0 - YOLOv5n 'Nano' models. Roboflow Integration, TensorFlow Export, OpenCV DNN Support. https://doi.org/10.5281/zenodo.5563715
9. Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., et al.: Yolov6: a single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976 (2022)
10. 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. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 7464–7475 (2023)
11. Reis, D., Kupec, J., Hong, J., Daoudi, A.: Real-time flying object detection with yolov8. arXiv preprint arXiv:2305.09972 (2023)
12. Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J.: Object detection in 20 years: a survey. Proceedings of the IEEE (2023)
13. Zhang, J., Xie, Z., Sun, J., Zou, X., Wang, J.: A cascaded r-cnn with multiscale attention and imbalanced samples for traffic sign detection. IEEE Access **8**, 29742–29754 (2020)
14. Liang, T., Bao, H., Pan, W., Pan, F.: Traffic sign detection via improved sparse r-cnn for autonomous vehicles. J. Adv. Transp. **2022**, 1–16 (2022)
15. Li, X., Xie, Z., Deng, X., Wu, Y., Pi, Y.: Traffic sign detection based on improved faster r-cnn for autonomous driving. J. Supercomp. (2022). https://doi.org/10.1007/s11227-021-04230-4
16. Hu, J., Wang, Z., Chang, M., Xie, L., Xu, W., Chen, N.: Psg-yolov5: a paradigm for traffic sign detection and recognition algorithm based on deep learning. Symmetry **14**(11), 2262 (2022)
17. Wang, J., Chen, Y., Dong, Z., Gao, M.: Improved yolov5 network for real-time multi-scale traffic sign detection. Neural Comput. Appl. **35**(10), 7853–7865 (2023)
18. Chen, J., Jia, K., Chen, W., Lv, Z., Zhang, R.: A real-time and high-precision method for small traffic-signs recognition. Neural Comp. Appl. **34**(3), 2233–2245 (2022)
19. Kisantal, M., Wojna, Z., Murawski, J., Naruniec, J., Cho, K.: Augmentation for small object detection. arXiv preprint arXiv:1902.07296 (2019)
20. Zhang, X., Izquierdo, E., Chandramouli, K.: Dense and small object detection in uav vision based on cascade network. In: Proceedings of the IEEE/CVF international conference on computer vision workshops, pp. 0–0 (2019)
21. Wang, X., Zhu, D., Yan, Y.: Towards efficient detection for small objects via attention-guided detection network and data augmentation. Sensors **22**(19), 7663 (2022)
22. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117–2125 (2017)
23. Li, H., Xiong, P., An, J., Wang, L.: Pyramid attention network for semantic segmentation. arXiv preprint arXiv:1805.10180 (2018)
24. 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)
25. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European conference on computer vision (ECCV), pp. 734–750 (2018)
26. Dong, Z., Li, G., Liao, Y., Wang, F., Ren, P., Qian, C.: Centripetalnet: pursuing high-quality keypoint pairs for object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10519–10528 (2020)
27. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet++ for object detection. arXiv preprint arXiv:2204.08394 (2022)
28. Lim, J.-S., Astrid, M., Yoon, H.-J., Lee, S.-I.: Small object detection using context and attention. In: 2021 International conference on artificial intelligence in information and communication (ICAIIC), pp. 181–186 (2021). IEEE
29. Yang, X., Yang, J., Yan, J., Zhang, Y., Zhang, T., Guo, Z., Sun, X., Fu, K.: Scrdet: Towards more robust detection for small, cluttered and rotated objects. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 8232–8241 (2019)
30. Fu, J., Sun, X., Wang, Z., Fu, K.: An anchor-free method based on feature balancing and refinement network for multiscale ship detection in sar images. IEEE Trans. Geosci. Remote Sens. **59**(2), 1331–1344 (2020)
31. Yi, K., Jian, Z., Chen, S., Zheng, N.: Feature selective small object detection via knowledge-based recurrent attentive neural network. arXiv preprint arXiv:1803.05263 (2018)
32. Srinivas, A., Lin, T.-Y., Parmar, N., Shlens, J., Abbeel, P., Vaswani, A.: Bottleneck transformers for visual recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 16519–16529 (2021)
33. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. Advances in neural information processing systems **28** (2015)
34. Luo, J.-H., Wu, J., Lin, W.: Thinet: a filter level pruning method for deep neural network compression. In: Proceedings of the IEEE international conference on computer vision, pp. 5058–5066 (2017)
35. Lee, N., Ajanthan, T., Torr, P.H.: Snip: single-shot network pruning based on connection sensitivity. arXiv preprint arXiv:1810.02340 (2018)
36. Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M.: Learning efficient object detection models with knowledge distillation. Advances in neural information processing systems **30** (2017)
37. Sun, R., Tang, F., Zhang, X., Xiong, H., Tian, Q.: Distilling object detectors with task adaptive regularization. arXiv preprint arXiv:2006.13108 (2020)
38. Yang, Z., Li, Z., Jiang, X., Gong, Y., Yuan, Z., Zhao, D., Yuan, C.: Focal and global knowledge distillation for detectors. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4643–4652 (2022)
39. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6848–6856 (2018)
40. Ma, N., Zhang, X., Zheng, H.-T., Sun, J.: Shufflenet v2: practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV), pp. 116–131 (2018)
41. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C.: Ghostnet: more features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1580–1589 (2020)
42. Tang, Y., Han, K., Guo, J., Xu, C., Xu, C., Wang, Y.: Ghostnetv2: enhance cheap operation with long-range attention. Adv. Neural. Inf. Process. Syst. **35**, 9969–9982 (2022)

43. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms–improving object detection with one line of code. In: Proceedings of the IEEE International conference on computer vision, pp. 5561–5569 (2017)
44. Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., Hu, S.: Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2110–2118 (2016)