scientific reports

OPEN

YOLO-BS: a traffic sign detection algorithm based on YOLOv8

Hong Zhang[1,2]✉, Mingyin Liang[1] & Yufeng Wang[1]

Traffic signs are pivotal components of traffic management, ensuring the regulation and safety of road traffic. However, existing detection methods often suffer from low accuracy and poor real-time performance in dynamic road environments. This paper reviews traditional traffic sign detection methods and introduces an enhanced detection algorithm (YOLO-BS) based on YOLOv8 (You Only Look Once version 8). This algorithm addresses the challenges of complex backgrounds and small-sized detection targets in traffic sign images. A small object detection layer was incorporated into the YOLOv8 framework to enrich feature extraction. Additionally, a bidirectional feature pyramid network (BiFPN) was integrated into the detection framework to enhance the handling of multi-scale objects and improve the performance in detecting small objects. Experiments were conducted on the TT100K dataset to evaluate key metrics such as model size, recall, mean average precision (mAP), and frames per second (FPS), demonstrating that YOLO-BS surpasses current mainstream models with mAP50 of 90.1% and FPS of 78. Future work will refine YOLO-BS to explore broader applications within intelligent transportation systems.

Keywords Traffic sign detection, Deep learning, YOLO, TT100K, BiFPN

Traffic sign detection utilizes computer vision and artificial intelligence technologies to recognize and interpret traffic signs on the road automatically. This technology is critical for enhancing driving safety, guiding traffic behavior and aiding in the decision-making processes of autonomous vehicles. Effective traffic sign detection not only helps drivers adhere to traffic regulations, but also reduces traffic accidents and improves traffic flow management.

As transportation networks expand and urbanization progresses rapidly, traffic signs become increasingly vital for managing traffic flow and ensuring road safety. Yet, the growing variety and complexity of urban traffic signs present significant challenges for traditional traffic sign recognition systems, which struggle to adapt to dynamic and complex traffic conditions. Consequently, enhancing the accuracy and real-time performance of automatic traffic sign recognition has emerged as a critical issue in contemporary traffic management.

This paper proposes a traffic sign detection algorithm based on an improved YOLOv8[1]. By incorporating a small object detection layer and integrating BiFPN structure into the neck network, the algorithm enhances the accuracy and efficiency of traffic sign detection, providing a more reliable solution for traffic management and autonomous driving systems.

The main contributions are as follows:

1. We explored the principles of small object detection layers for detecting traffic signs. By integrating a small object detection layer into the YOLOv8 framework, we verified its effectiveness in detecting traffic signs.
2. We discussed the advantages of the BiFPN network and integrated it within the YOLOv8 model. Experiments confirmed that the improved model significantly enhanced the accuracy of traffic sign detection.

Related works

Traffic signs are essential elements on road surfaces that indicate traffic rules, warn of road hazards, and provide important information for drivers. They play a critical role in ensuring road safety, maintaining order, and improving traffic efficiency. Vehicle-mounted traffic sign recognition systems effectively guide and regulate driver behavior, ensuring safe driving and reducing traffic accidents. Additionally, with the ongoing development of intelligent transportation systems, detecting and recognizing traffic signs further enhances autonomous driving technology, improving road commuting efficiency.

The process of traffic sign detection involves two main steps: localization and recognition of the signs. During the localization phase, the system identifies the presence and exact location of traffic signs within complex road

[1]Transportation Institute of Inner Mongolia University, Hohhot 010070, China. [2]Inner Mongolia Engineering Research Center for Intelligent Transportation Equipment, Hohhot 010070, China. ✉email: imu_hongzhang@outlook.com

environments. In the recognition and classification phase, the system interprets the detected signs to determine their types, such as stop signs or speed limits.

Traditional methods for traffic sign detection have primarily relied on image processing and machine learning techniques, using detection algorithms such as Histogram of Orientation Gradients (HOG) and Scale-Invariant Feature Transformations (SIFT), together with classification algorithms such as Support Vector Machines (SVM) and Random Forests (RF).

The traditional approach primarily relies on distinctive features of traffic signs, such as their specific colors (red, blue, yellow, etc.) and prominent shapes (triangular, circular, rectangular, etc.). These unique attributes are used to extract features for detection, followed by classification with a trained classifier. De La Escalera et al.[2] elected the color and shape of the sign's corners as features to extract traffic signs from the environment and used neural networks to classify the identified signs. Gómez-Moreno et al.[3] proposed a color segmentation method using SVM and increased the speed with the Look-Up Table (LUT) while maintaining the quality. Yuan et al.[4] introduced a robust recognition method for traffic signs based on Color Global and Locally Oriented Edge Magnitude Patterns (Color Global LOEMP). This technique effectively integrates color, global spatial structure, global orientation structure, and local shape information, significantly enhancing the efficiency of traffic sign recognition. Berkaya et al.[5] utilized the EDCircles circle detection algorithm combined with an RGB-based color thresholding technique to detect traffic signs. This approach used a feature extraction method that integrates GABOR, LBP, and HOG techniques, and classification was subsequently performed with a SVM. This methodological combination effectively enhanced the traffic sign detection process.

Whether based on color features or shape features, these features are prone to interference from external objects of similar color and shape in complex road conditions, resulting in poor generalization. The emergence of machine learning offers a new approach to traffic sign detection, with advantages such as strong generalization and high robustness. Sun et al.[6] introduced a traffic sign recognition method that skillfully combines a HOG for feature extraction with an Extreme Learning Machine (ELM) classifier for rapid classification. The model not only has high recognition accuracy in the GTSRB dataset but also exhibits significant advantages in computational efficiency.

Aiming to strike a balance between computational efficiency and recognition accuracy, Huang et al.[7] introduced a traffic sign recognition method based on ELM. This method integrates an enhanced version of the HOG for feature extraction with the ELM classifier.The approach achieved high recognition accuracy both on the GTSRB and the Belgium traffic sign classification dataset (BTSC), while maintaining very high computational efficiency. Ellahyani et al.[8] developed a method for traffic sign recognition by initially employing the HSI color space for color threshold segmentation to pinpoint potential traffic sign regions. This approach was augmented with the use of SVM and Random Forest classifiers for enhanced accuracy, and utilized HOG features to recognize the traffic signs effectively.

As cities expand and road networks become more complex, traditional traffic sign recognition systems are increasingly challenged by issues such as lighting variations, occlusions, and the similarities among different types of signs. Traditional computer vision techniques often struggle in these complex environments, lacking the robustness and accuracy needed. Consequently, deep learning-based traffic sign recognition algorithms have gained prominence.

Deep learning algorithms utilize neural networks to model intricate relationships between inputs and outputs. These algorithms have gained popularity in traffic sign recognition due to their ability to autonomously learn high-level features directly from raw data. This capability significantly diminishes the necessity for manual feature extraction, streamlining the process and enhancing the effectiveness of recognition systems. Li et al.[9] utilized CNN to specifically target the detection and recognition of traffic signs in the United States, concentrating particularly on speed limit signs. Their proposed method demonstrated impressive detection performance on the LISA-TS dataset, highlighting its effectiveness in identifying these critical signs.

Li & Wang[10] combined Faster R-CNN with MobileNets to precisely locate and classify small traffic signs. This innovative approach leveraged the strengths of both technologies: Faster R-CNN for its efficient and accurate detection capabilities and MobileNets for its lightweight, mobile-friendly architecture, resulting in enhanced performance in recognizing smaller traffic signs. Tabernik & Skočaj[11] enhanced the Mask R-CNN framework to better recognize small traffic signs and introduced a novel data augmentation technique to improve the model's generalization capabilities. Evaluations on both the DFG and the Swedish traffic sign datasets demonstrated significant performance gains, with the refined Mask R-CNN model achieving metrics such as mAP50 up to 95.5%. Zhang et al.[12] proposed a cascaded R-CNN model with multiscale attention, which improves detection accuracy by focusing on multiscale feature extraction and balancing imbalanced datasets, thus enhancing the model's performance in detecting small-sized traffic signs. Wang et al.[13] developed an enhanced lightweight traffic sign recognition algorithm based on YOLOv4-Tiny. The algorithm refines the K-means clustering method to generate anchor frames tailored to the traffic sign dataset, which significantly improves detection recall and target localization precision. When evaluated on the TT100K dataset, the improved algorithm achieved a mean Average Precision (mAP) at 0.5 of 52.07% and demonstrated enhanced real-time performance. Dewi et al.[14] combined YOLOv3 and Densenet models, incorporating SPP to optimize the feature extraction. This innovation significantly boosted the recognition accuracy of small traffic signs.

The comparison of traffic sign detection algorithms is presented in Table 1. Traditional traffic sign detection algorithms depend on manually features and tend to be sensitive to lighting conditions and complex backgrounds. While machine learning techniques can automate feature extraction from images, their detection performance is generally inferior. In contrast, deep learning approaches achieve high accuracy rates; however, they struggle with detecting small targets effectively. Therefore, the paper will focus on leveraging deep learning to enhance the accuracy of traffic sign detection, improving the robustness and real-time performance of detection algorithms.

Method	Algorithms	Advantages	Disadvantages
Traditional methods	Algorithms based on colors or shapes	Simple to implement, low computational resources	Sensitive to complex backgrounds and lighting changes
Machine learning	HOG, SIFT, SVM, RF	Require less data compared to deep learning methods	poor accuracy, time-consuming
Deep learning	CNN	Automatically extracts features, high accuracy	Requires a significant amount of labeled data for training
	Faster R-CNN	High accuracy, particularly for small objects	Higher computational and time costs
	SSD, YOLO	Fast speed with relatively high accuracy	Poor performance on small object detection

Table 1. Comparison of traffic sign detection algorithms.

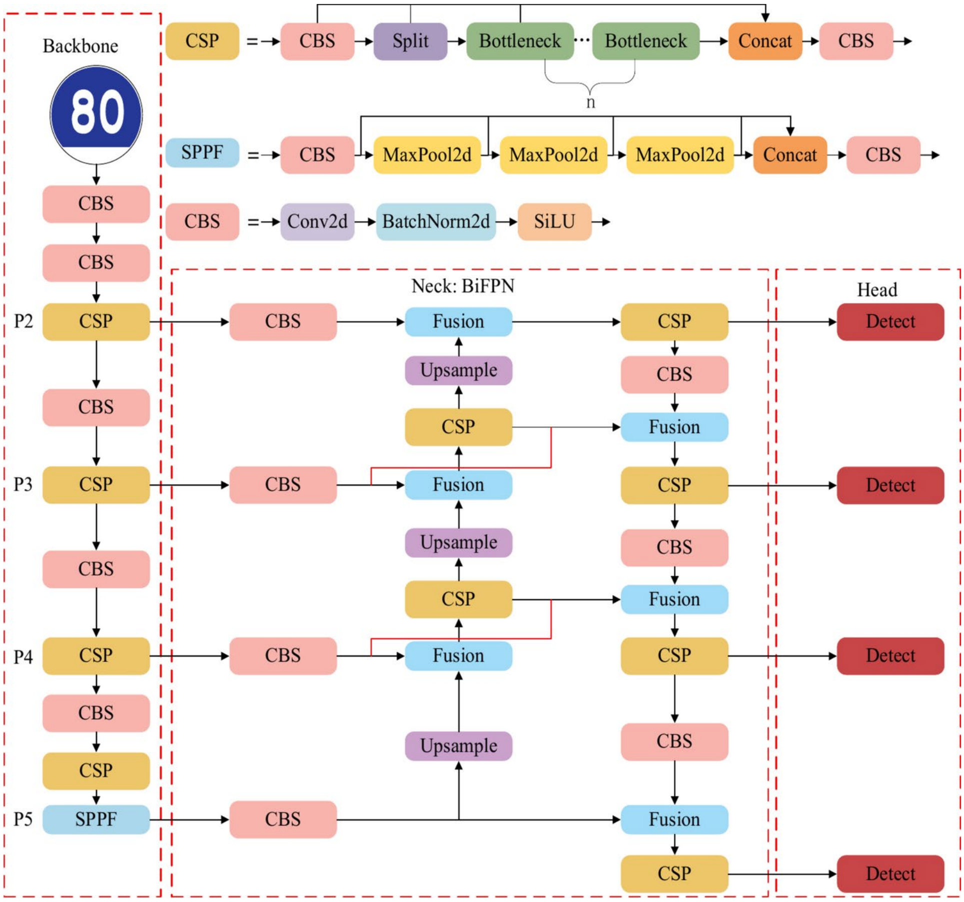


Fig. 1. The framework of YOLO-BS.

Methodology

Method overview

The YOLO-BS detection algorithm presented in this paper is a one-stage traffic sign detection algorithm, and its framework is shown in Fig. 1. The algorithm consists of three main parts: the backbone, the neck and the head.

Initially, images undergo preprocessing through input part data augmentation and other operations before being fed into the backbone network. The backbone network extracts features from the images, producing feature maps at four different scales. These feature maps are then processed by the neck network for feature fusion, resulting in four scaled features. Finally, these features are input into the detection head network for prediction, which outputs the position, confidence, and classification information of the detection boxes at each scale.

Small object detection layer

In traffic sign detection, the traffic sign dataset categorizes the signs into three groups—large, medium, and small—based on their apparent sizes from varying distances. As vehicles travel, they encounter traffic signs at different ranges, from distant to close, which typically results in a predominance of smaller targets in the traffic sign images. To address this issue, we have added a small object detection layer to the YOLO architecture to enhance its ability to detect small targets. The dimensions of the detection feature maps and the corresponding receptive fields of the improved detection layer are shown in Table 2.

Detection layer	Feature map dimensions	Receptive field	Detect object size
P2	160 × 160	Small	Very small
P3	80 × 80	Medium	Small
P4	40 × 40	Large	Medium
P5	20 × 20	Larger	Large

Table 2. Detection layer feature map information.

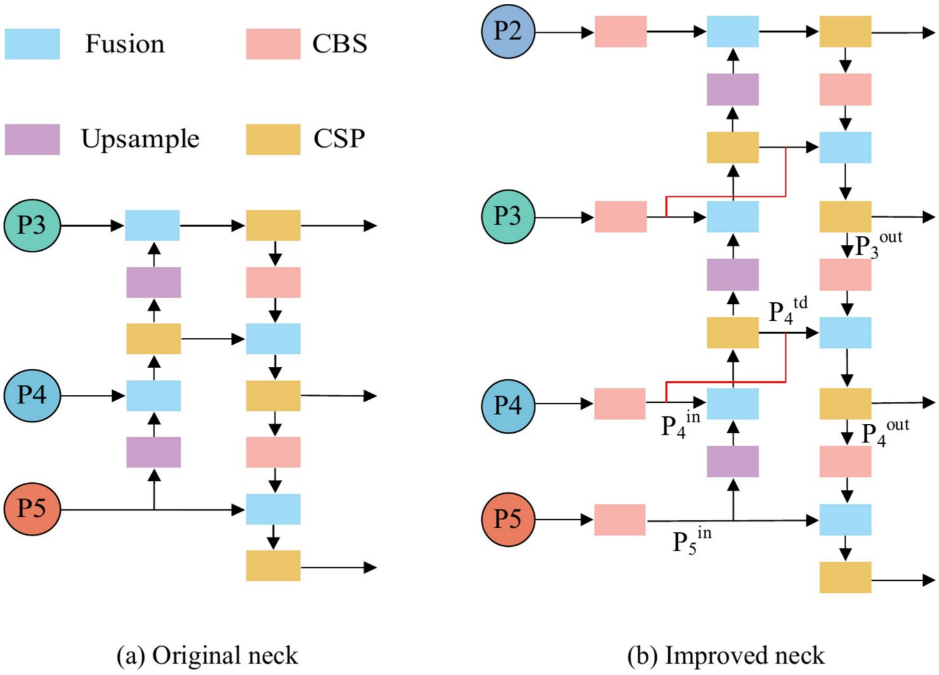


Fig. 2. Comparison of the neck before and after integrating small object layer and BiFPN.

The small object detection layer is introduced mainly by adding higher-resolution feature maps. These feature maps can retain more spatial detail information, thus increasing the network's sensitivity to small targets.

Traditional YOLO conducts target detection using a single-scale feature map, which limits its efficacy in detecting smaller objects. By incorporating a small object detection layer, the architecture can simultaneously engage multiple-scale feature maps for detection. Specifically, the P2 feature layer in the backbone is convolved to obtain scale features rich in small target information, and then fused with the Upsample layer, followed by input to the lower CSP module for multi-scale feature fusion. Finally, the detection head detects the fused multi-scale features.

YOLOv8 introduces mesoscale and small-scale feature maps in addition to the original scale, enabling multi-level detection across these varying scales. This multi-scale detection strategy enhances the network's ability to comprehensively capture a wider range of targets within the image, including those that are notably small.

BiFPN

The Bidirectional Feature Pyramid Network[15] represents an advanced feature pyramid structure aimed at bolstering the multi-scale feature fusion capabilities of CNN for target detection tasks. BiFPN has demonstrated remarkable value in the field of target detection, becoming a key component in various cutting-edge detection frameworks such as EfficientDet and YOLO.

Unlike traditional Feature Pyramid Networks (FPN), which enhance the detection of various-sized targets by merging features of different scales via a top-down path, FPNs typically employ a homogeneous and unidirectional information transfer. BiFPN enhances this approach by introducing a bidirectional information flow, significantly optimizing feature utilization and representation through a weighted feature fusion mechanism, thereby improving the overall efficacy of target detection.

Figure 2a is the original neck of YOLOv8, while the neck network structure after integrating BiFPN into YOLO is shown in Fig. 2b. BiFPN realizes top-down and bottom-up bidirectional information flow through a bidirectional feature pyramid structure. This design not only enhances the information transfer between different layers of features, but also enables a fuller fusion of features from different scales, thus improving the network's ability to detect multi-scale targets.

BiFPN employs a fast normalized fusion method that improves upon the traditional feature fusion methods used in FPN[16]. In conventional FPN, feature maps of different scales are typically merged using a simple addition operation, which does not adequately account for the varying importance of features across scales. In contrast, BiFPN introduces learnable weight coefficients for each scale's feature maps during the fusion process, as shown in Eq. (1):

$$O = \sum_{i} \frac{\omega_i \, I_i}{\varepsilon + \sum_{j} \omega_j} \tag{1}$$

where I_i is the input feature, O is the output feature, ω_i and ω_j are the learnable weights, and $\varepsilon = 0.0001$ is a small amount to mitigate potential numerical instability.

Given a list of multiscale features $\overrightarrow{P}^{in}$ = (P_1^{in}, P_2^{in}, …), where P_i^{in} denotes the feature at layer i. The transformed BiFPN effectively aggregates the different features to obtain a new feature list $\overrightarrow{P}^{out}$ = f($\overrightarrow{P}^{in}$) as the output. The BiFPN feature fusion process at layer 4 can be described as Eq. (2 and 3).

$$P_4^{td} = \mathrm{Conv}\left(\frac{\omega_1 \cdot P_4^{in} + \omega_2 \cdot \mathrm{Resize}(P_5^{in})}{\omega_1 + \omega_2 + \varepsilon}\right) \tag{2}$$

$$P_4^{out} = \mathrm{Conv}\left(\frac{\omega_1' \cdot P_4^{in} + \omega_2' \cdot P_4^{td} + \omega_3' \cdot \mathrm{Risize}(P_3^{out})}{\omega_1' + \omega_2' + \omega_3' + \varepsilon}\right) \tag{3}$$

where P_4^{td} denotes the intermediate features of layer 4 on the top-down path, while P_4^{out} denotes the output features of layer 4 on the bottom-up path. Resize is an upsampling or downsampling operation for resolution matching. Meanwhile, 'Conv' usually refers to a convolution operation for feature processing.

Experiments

TT100K dataset

The TT100K traffic signs dataset[17], a collaborative creation by the joint lab of Tsinghua University and Tencent, stands as the first large-scale traffic signs and signals dataset in China. It comprises over 100,000 traffic-related images, encompassing a diverse range of traffic signs and traffic lights. The categorization of traffic signs within the dataset is detailed in Fig. 3.

Some traffic signs in the TT100K traffic sign dataset have a myriad of labels, while others have fewer than 100 labels, leading to an imbalance in the sample distribution and making detection challenging. To address this issue, a refined traffic sign dataset containing 45 categories was created by isolating those with more than 100 samples through a screening procedure. The number of corresponding labels for the screened traffic signs is illustrated in Fig. 4. The refined traffic sign dataset is divided into a training set and a test set in an 8:2 ratio, with the default validation set being the same as the test set.

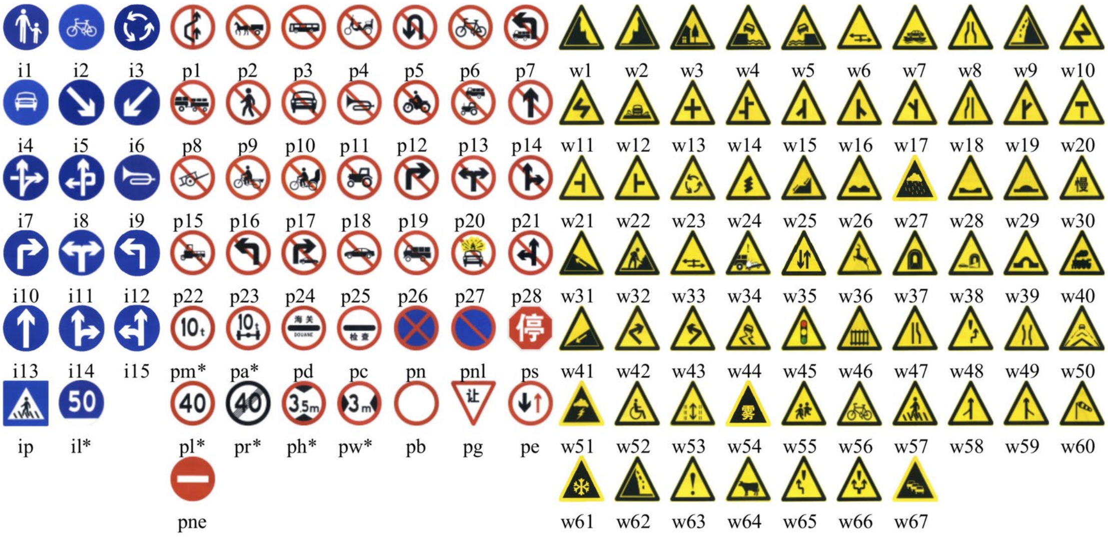

i1 i2 i3 p1 p2 p3 p4 p5 p6 p7 w1 w2 w3 w4 w5 w6 w7 w8 w9 w10
i4 i5 i6 p8 p9 p10 p11 p12 p13 p14 w11 w12 w13 w14 w15 w16 w17 w18 w19 w20
i7 i8 i9 p15 p16 p17 p18 p19 p20 p21 w21 w22 w23 w24 w25 w26 w27 w28 w29 w30
i10 i11 i12 p22 p23 p24 p25 p26 p27 p28 w31 w32 w33 w34 w35 w36 w37 w38 w39 w40
i13 i14 i15 pm* pa* pd pc pn pnl ps w41 w42 w43 w44 w45 w46 w47 w48 w49 w50
ip il* pl* pr* ph* pw* pb pg pe w51 w52 w53 w54 w55 w56 w57 w58 w59 w60
pne w61 w62 w63 w64 w65 w66 w67

Fig. 3. Traffic sign categories in TT100K dataset.

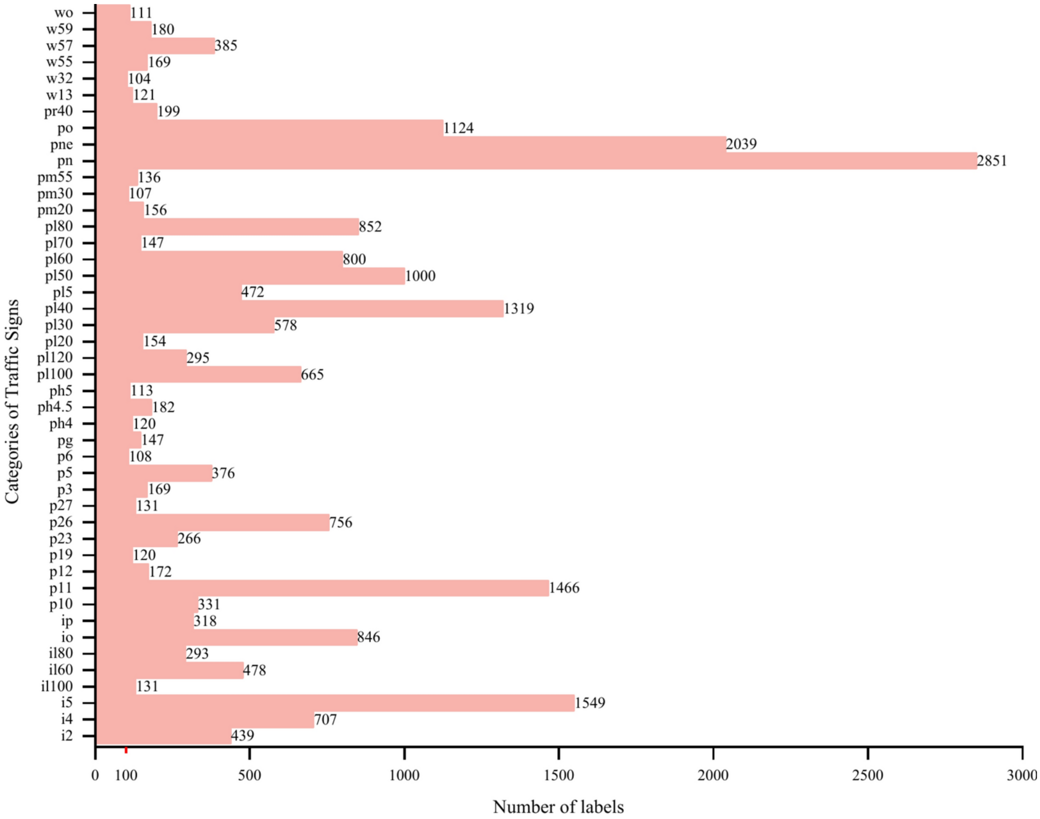


Fig. 4. Number of traffic sign labels screened out.

		Prediction	
		Positive	**Negative**
Ground truth	True	TP	TN
	False	FP	FN

Table 3. Confusion matrix.

Evaluation indicators

The metrics for evaluating the performance of the YOLO algorithm are P(precision), R(recall) and mAP (mean Average Precision). These evaluation metrics are explained based on the confusion matrix as shown in Table 3.

Precision indicates the ratio of correctly predicted positive samples (TP) to all predicted positive samples (TP + FP), and the accuracy rate mainly depends on whether the prediction results are accurate or not. Its formula is shown in Eq. (4):

$$P = \frac{TP}{TP + FP} \tag{4}$$

Recall indicates the ratio of correctly predicted as positive samples (TP) to all true cases as positive samples (TP + FN), and recall mainly depends on whether the prediction results are comprehensive or not. Its formula is shown in Eq. (5):

$$R = \frac{TP}{TP + FN} \tag{5}$$

AP (Average Precision) refers to the area of the region below the PR curve plotted with recall as the horizontal coordinate and precision as the vertical coordinate, AP is used to measure the performance of the algorithm in recognizing each category. Its formula is shown in Eq. (6):

$$AP = \int_0^1 PdR \tag{6}$$

Configuration information	Parameters
CPU	Intel(R) Core(TM) i9-9900 K
Frequency	3.60 GHz
Cores	16
Threads	32
RAM	64G
GPU	NVIDIA RTX 2080Ti
Video random access memory	12G

Table 4. Computer hardware configuration.

Key parameters	Parameter value
Epochs	300
Image size	640 × 640
Batch	8
Patience	10
Model	YOLOv8s.pt
Workers	8

Table 5. Key parameter values.

Model	GFLOPs	P/%	R/%	mAP50/%	mAP50–95/%	FPS
YOLOv8	28.5	81.7	73.8	81.8	63.5	91
YOLOv8-Small object	38.0	86.3	79.2	87.3	67.8	83
YOLOv8-Small object-BiFPN (YOLO-BS)	55.2	87.9	80.5	90.1	70.3	78

Table 6. Results of ablation experiments.

The value of mAP is taken equal to the average value of AP over all categories and is used to measure the performance of the algorithm in recognizing all categories.

Ablation experiments

The system used for the experiment was Windows 10 Professional, and the computer hardware configuration is shown in Table 4. The deep learning framework used was Pytorch2.3, and the commonly used experimental environments such as CUDA11.8, cudnn11.x and opencv4.6.0 were installed.

The training-related parameters are set according to Table 5 to train the YOLOv8 model on the traffic sign dataset.

Results discussion

Results of ablation experiments

The ablation experiments were conducted to evaluate the performance improvements of the proposed YOLO-BS model over the baseline YOLOv8 and YOLOv8 with a small object detection layer. The results are presented in Table 6, showcasing the impact of each enhancement on key metrics such as GFLOPs, P, R, mAP50, mAP50-95, and FPS. Meanwhile, the variation curves of precision, recall, and mAP50 for the three models throughout the training process are presented in Fig. 5.

As shown in Fig. 5, YOLOv8 exhibits the lowest precision, recall, and mAP50 values, while YOLO-BS achieves the highest precision, recall, and mAP50 values. Specifically, the baseline YOLOv8 achieved a precision of 81.7%, recall of 73.8%, and mAP50 of 81.8%. By adding the small object detection layer, the model's precision increased to 86.3%, recall to 79.2%, and mAP50 to 87.3%. The full YOLO-BS model, incorporating both the small object detection layer and BiFPN, further improved these metrics to 87.9% precision, 80.5% recall, and 90.1% mAP50. The results indicate that the YOLO-BS significantly outperforms the baseline YOLOv8 in all metrics.

Although the FPS is slightly lower than the baseline, the improved model maintains a good balance between speed and accuracy. The integration of the BiFPN and small object detection layer into the YOLOv8 architecture has improved the performance of the YOLO-BS model. By incorporating BiFPN, these frameworks significantly enhance their capabilities in detecting small and multi-scale targets. Particularly in real-time detection tasks, the efficient feature fusion mechanism of BiFPN allows the detection system to sustain high operational speeds without compromising accuracy. This balance of speed and precision underscores BiFPN's pivotal role in improving the effectiveness of detection systems. The bidirectional information flow and weighted feature fusion

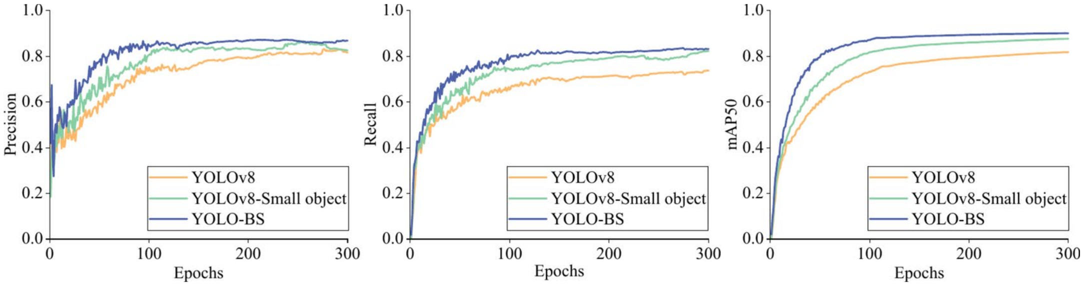


Fig. 5. The variation curves of precision, recall, and mAP50.

Model	GFLOPs	P/%	R/%	mAP50/%	mAP50–95/%	FPS
Faster R-CNN	134	58.8	52.3	55.7	42.4	45
Mask R-CNN	187	55.0	52.8	53.9	40.5	37
TOOD	125	61.3	53.7	71.4	51.5	47
YOLOv5	**16.1**	82.5	77.1	83.1	63.4	39
YOLOv7	103.9	52.1	55.4	52.9	39.4	65
YOLOv9	314.0	87.3	80.2	87.9	68.8	31
YOLOv10	24.6	84.9	72.3	82.4	64.6	83
Wang et al. [18]	17.9	–	–	65.1	–	**95**
Wang et al. [13]	–	–	64.5	52	–	87
Li et al. [19]	–	–	–	89.7	–	65
Gong et al.[20]	–	–	–	77.3	–	33
Ours(YOLO-BS)	55.2	**87.9**	**80.5**	**90.1**	**70.3**	78

Table 7. Results of comparison with other algorithms on the TT100K dataset.

mechanism in BiFPN allow the model to better manage multi-scale features, enhancing detection accuracy for small targets.

Additionally, the small object detection layer increases the network's sensitivity to smaller traffic signs, which are often encountered in traffic sign detection scenarios.

Comparison with other algorithms

As shown in Table 7, the YOLO-BS model demonstrates competitive performance compared to other advanced algorithms. It achieves the highest precision (87.9%) and recall (80.5%) among the models listed, along with an impressive mAP50 of 90.1%.

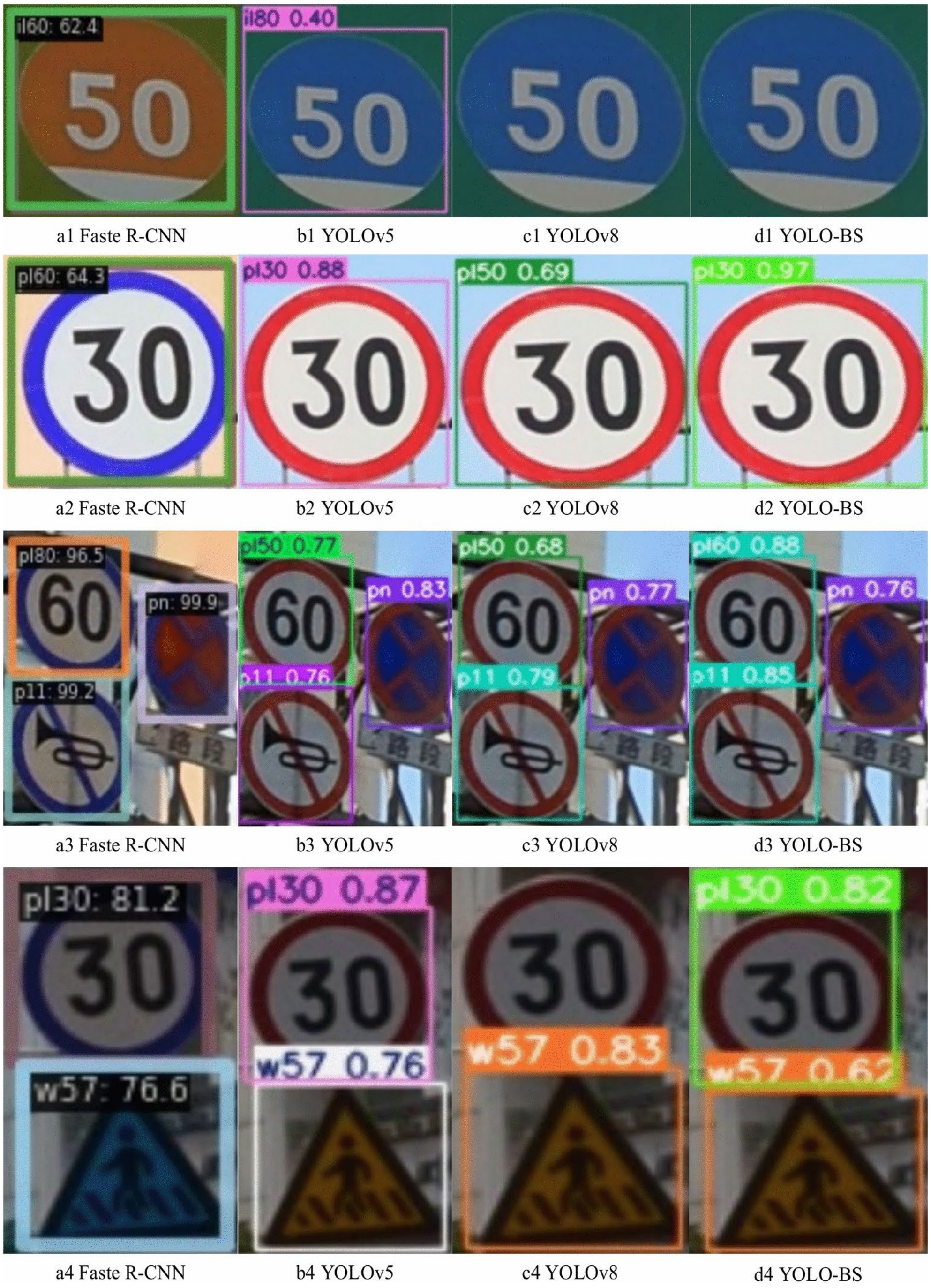
We visualized the performance of Faster R-CNN, YOLOv5, YOLOv8, and YOLO-BS on the validation set, as shown in Fig. 6, where Fig. 6a1–a4 presents the detection results of Faster R-CNN, Fig. 6b1–b4 presents the detection results of YOLOv5, Fig. 6c1–c4 presents the detection results of YOLOv8, and Fig. 6d1–d4 displays the detection results of YOLO-BS. As depicted in Fig. 6a1, a2, a3, b1, b3, c2, c3, false positive samples are observed in the detection results of the other three models, except for YOLO-BS. Additionally, a missed detection is observed in the YOLOv8 model's results, as shown in Fig. 6c4. It is important to note that Fig. 6c1, d1 does not represent missed detections. This is because the number of "il50" is less than 100, and the filtered dataset does not include traffic signs for the minimum speed limit of 50 km/h.

The occurrences of missed detections and false positive samples are mainly attributed to the small size of the traffic signs in the images, image blurriness, and strong lighting conditions. Our proposed YOLO-BS addresses these challenges effectively, successfully identifying small traffic signs in real-world traffic scenarios with high detection accuracy, and exhibiting almost no missed or false detections.

The inclusion of BiFPN and the small object detection layer significantly enhances the model's capability to accurately detect and classify small and multi-scale traffic signs. Despite the slight reduction in FPS compared to some baseline models, YOLO-BS strikes a favorable balance between speed and accuracy, making it suitable for real-time detection tasks where precision is critical. Future work will focus on further optimizing this balance, potentially through hardware acceleration techniques or more efficient network architectures.

Conclusion

This paper proposed YOLO-BS, a traffic sign detection algorithm based on an improved YOLOv8 framework. The small object detection layer enhances the network's sensitivity to smaller objects, which are commonly encountered in traffic sign detection scenarios. Additionally, the bidirectional information flow and weighted feature fusion mechanism in BiFPN enable the model to better handle multi-scale features, improving the detection accuracy for small targets.



a1 Faste R-CNN b1 YOLOv5 c1 YOLOv8 d1 YOLO-BS

a2 Faste R-CNN b2 YOLOv5 c2 YOLOv8 d2 YOLO-BS

a3 Faste R-CNN b3 YOLOv5 c3 YOLOv8 d3 YOLO-BS

a4 Faste R-CNN b4 YOLOv5 c4 YOLOv8 d4 YOLO-BS

Fig. 6. The detection results of Faster R-CNN, YOLOv5, YOLOv8, and YOLO-BS.

By incorporating a small object detection layer and integrating the BiFPN, the algorithm significantly enhances the accuracy and robustness of traffic sign detection. Experimental results on the TT100K dataset demonstrate that YOLO-BS outperforms current mainstream models, achieving high mAP and FPS metrics, making it a promising solution for real-time traffic sign detection in intelligent transportation systems. Future research will focus on further optimizing the YOLO-BS model, potentially through hardware acceleration techniques and more efficient network architectures, to enhance its real-time performance further.

Data availability

The datasets used and analyzed during the current study are publicly available and can be accessed from https://cg.cs.tsinghua.edu.cn/traffic-sign/

Received: 6 September 2024; Accepted: 24 January 2025

Published online: 04 March 2025

References

1. Varghese, R., Sambath, M. 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, pp. 1–6. (2024).
2. De La Escalera, A., Moreno, L. E., Salichs, M. A. & Armingol, J. M. Road traffic sign detection and classification. *IEEE Trans. Industr. Electron.* **44**(6), 848–859 (1997).
3. Gómez-Moreno, H., Maldonado-Bascón, S., Gil-Jiménez, P. & Lafuente-Arroyo, S. Goal evaluation of segmentation algorithms for traffic sign recognition. *IEEE Trans. Intell. Transp. Syst.* **11**(4), 917–930 (2010).
4. Yuan, X., Hao, X., Chen, H. & Wei, X. Robust traffic sign recognition based on color global and local oriented edge magnitude patterns. *IEEE Trans. Intell. Transp. Syst.* **15**(4), 1466–1477 (2014).
5. Berkaya, S. K., Gunduz, H., Ozsen, O., Akinlar, C. & Gunal, S. On circular traffic sign detection and recognition. *Expert Syst. Appl.* **48**, 67–75 (2016).
6. Sun, Z.-L., Wang, H., Lau, W.-S., Seet, G. & Wang, D. Application of BW-ELM model on traffic sign recognition. *Neurocomputing* **128**, 153–159 (2014).
7. Huang, Z., Yu, Y., Gu, J. & Liu, H. An efficient method for traffic sign recognition based on extreme learning machine. *IEEE Trans. Cybern.* **47**(4), 920–933 (2016).
8. Ellahyani, A., El Ansari, M. & El Jaafari, I. Traffic sign detection and recognition based on random forests. *Appl. Soft Comput.* **46**, 805–815 (2016).
9. Li, Y., Møgelmose, A. & Trivedi, M. M. Pushing the “Speed Limit”: high-accuracy US traffic sign recognition with convolutional neural networks. *IEEE Trans. Intell. Vehicles* **1**(2), 167–176 (2016).
10. Li, J. & Wang, Z. Real-time traffic sign recognition based on efficient CNNs in the wild. *IEEE Trans. Intell. Transp. Syst.* **20**(3), 975–984 (2018).
11. Tabernik, D. & Skočaj, D. Deep learning for large-scale traffic-sign detection and recognition. *IEEE Trans. Intell. Transp. Syst.* **21**(4), 1427–1440 (2019).
12. Zhang, J. M., Xie, Z. P., 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).
13. Wang, L., Zhou, K., Chu, A., Wang, G. & Wang, L. An improved light-weight traffic sign recognition algorithm based on YOLOv4-tiny. *IEEE Access* **9**, 124963–124971 (2021).
14. Dewi, C., Chen, R.-C., Yu, H. & Jiang, X. Robust detection method for improving small traffic sign recognition based on spatial pyramid pooling. *J. Ambient Intell. Human. Comput.* **14**(7), 8135–8152 (2023).
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. Chen, J. Q. et al. Efficient and lightweight grape and picking point synchronous detection model based on key point detection. *Comput. Electron. Agricult.* **217**, 108612 (2024).
17. 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).
18. Wang, J. F., Chen, Y., Dong, Z. K. & Gao, M. Y. Improved YOLOv5 network for real-time multi-scale traffic sign detection. *Neural Comput. Appl.* **35**(10), 7853–7865 (2023).
19. Li, Z. S. et al. Toward effective traffic sign detection via two-stage fusion neural networks. *IEEE Trans. Intell. Trans. Syst.* https://doi.org/10.1109/TITS.2024.3373793 (2024).
20. Gong, C. P., Li, A. J., Song, Y. M., Xu, N. & He, W. K. Traffic sign recognition based on the YOLOv3 algorithm. *Sensors* **22**(23), 9345 (2022).

Author contributions

Hong Zhang and Mingyin Liang wrote the main manuscript text and Validated the method proposed in the article. Mingyin Liang. prepared Figs. 1, 2, 3 and 4. Yufeng Wang Collated experimental data. All authors reviewed the manuscript.

Funding

This research is supported by National Natural Science Foundation of China (NSFC) (62362053), Program for Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region of China (NJYT23060), 2024 Basic Research and Applied Basic Research of Hohhot (2024-G-J-29), "Inner Mongolia Science and Technology Achievement Transfer and Transformation Demonstration Zone, University Collaborative Innovation Base, and University Entrepreneurship Training Base" Construction Project (Supercomputing Power Project) (21300-231510).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to H.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.