Journal of Real-Time Image Processing (2024) 21:24
https://doi.org/10.1007/s11554-023-01403-7

RESEARCH

A novel lightweight real-time traffic sign detection method based on an embedded device and YOLOv8

Yuechen Luo[1] · Yusheng Ci[1] · Shixin Jiang[2] · Xiaoli Wei[1]

Received: 6 February 2023 / Accepted: 15 December 2023 / Published online: 26 January 2024

Abstract
Traffic sign recognition, as one of the key steps of intelligent driving technologies, effectively avoids most traffic accidents by detecting the location and type of traffic signs in real time and providing the information to drivers or autonomous vehicles promptly. In addition, edge devices close to users has become an inevitable requirement for the development of IoT technology, real-time computing and the realization of network edge intelligence. Nowadays, the YOLO algorithm in object detection, developed to YOLOv8, is accompanied by various lightweight networks and lightweight methods to win by "fast", so this paper will propose an algorithm, fusing Ghost module and Efficient Multi-Scale Attention Module into YOLOv8, so that the model can improve the computing speed while maintaining the original characteristics. Furthermore, we choose Raspberry Pi as the object detection device due to its many characteristics such as lightweight, low power consumption. Through experiments, the model is trained on CCTSDB dataset, and the improved algorithm is tested on Raspberry Pi 4B. The results show that for three types of traffic signs, namely prohibited, indication and warning, the recognition accuracy mAP reaches 93.5% on the poor weather test set and 82.9% on the original test set, and the inference delay of Raspberry Pi reaches 0.79 s, which is effective in actual road test experiments. The improved model's accuracy has increased by 6.4% and 3.8% on two separate test sets compared to the original model, while the detection time has been reduced by 0.12 s. This research is of great significance to the maturation of assisted driving and autonomous driving technologies.

Keywords Traffic sign recognition · GhostNet · Efficient multi-scale attention · Yolov8 · Raspberry Pi · Deep learning

1 Introduction

In assisted driving and autonomous driving, one of the most critical parts is the detection of traffic signs. In a very short time, the detection of the specific location and type of traffic signs some distance ahead of the current driving road plays a crucial role in the decision-making of the driver and the autonomous vehicle. For high-speed vehicles, real-time detection results not only facilitate the driver to pre-judge the road and make quick decisions in advance but also play an important role in the real-time decision-making of autonomous vehicles, so the time and accuracy of traffic sign detection directly affect whether autonomous vehicles can land [1]. Traffic signs play an important role in driving safety, and as a core issue for assisted, autonomous driving, traffic sign detection has received a lot of attention from researchers. Traffic sign detection mainly collects images through vehicle cameras, uses vehicle computers as edge computing devices, uses artificial intelligence, deep learning, and other technologies to identify and detect traffic signs in images, and provides information to drivers or autonomous vehicles in real-time to achieve pre-determination of road traffic, thereby increasing the driver's reaction time and providing key information for autonomous vehicles' decision making to ensure safe driving. The recognition of traffic signs drives the advancement of assisted, autonomous driving

✉ Yusheng Ci
ciyusheng1999@126.com

Yuechen Luo
120L052117@stu.hit.edu.cn

Shixin Jiang
120L021329@stu.hit.edu.cn

Xiaoli Wei
120L052101@stu.hit.edu.cn

1 School of Transportation Science and Engineering, Harbin Institute of Technology, Harbin, China

2 Faculty of Computing, Harbin Institute of Technology, Harbin, China

technology, accelerates the process of landing autonomous driving technology, enables the early realization of intelligent travel, and makes people's daily life more convenient.

The vision task of traffic sign detection started decades ago, and there is some research focused on classical computational vision methods by manually extracting features, such as color features [2, 3], and shape features [21], however, these algorithms are not able to obtain the expected results stably, since they face various challenges, such as internal and external conditions of the traffic sign environment, and the external conditions are some environmental factors conditions such as weather conditions, complex backgrounds, which are unchangeable; on the other hand, the internal conditions are variables such as response time, detection accuracy, etc. that can be controlled by the algorithm, and these challenges promote the development of traditional traffic sign detection. With the continuous development of deep learning, convolutional neural networks appeared in people's view, and the field of computer vision has since bid farewell to the era of manual feature extraction. For the traffic sign detection problem, first, CNN is used to classify the background and extract the region of interest to obtain the candidate object, then it is developed to extract features using CNN, and then support vector machine [13] is used to classify the extracted features and finally determine the object location, in this stage, the object detection is divided into two different types of methods: two stage and one stage, the former algorithm needs to obtain the candidate frame area first and then perform classification, this type of algorithm is represented by R-CNN algorithm, mainly including Fast R-CNN [4], etc., such as T Liang et al. achieved certain effect by implementing R-CNN to identify traffic signs [25]. one stage algorithm, in contrast, directly provides the bounding box and classification stage, treating the whole process as a regression process, and this type of algorithm is represented by YOLO [16–18], which has had a great impact on the field of state-of-the-art real-time object detection, and it has been developed to the latest YOLOv8 [35], which is a new stage for the field of object detection. Yolov8 has been widely applied in various fields of object detection tasks. Researchers have adopted various solutions to address specific issues in different domains. Lou et al. utilized the improved DC-YoloV8 algorithm to enhance the detection capability for small objects [36]. Other researchers have also contributed to improving the detection accuracy of YoloV8 [37–39]. In addition, some researchers have made significant efforts in the field of traffic sign detection [40–42]. Huang et al. employed an enhanced YOLOv5 for detecting traffic signs on the TT100k dataset, achieving commendable results [40]. Liang et al. proposed the improved Sparse R-CNN to enhance the accuracy of traffic sign detection [41]. However, these methods are limited by hardware and require specific hardware to train the model and inference model.

In the twenty-first century, data is exploding, especially in the field of artificial intelligence, which requires more and more computational power, and the field of cloud platforms is rapidly developing to enhance the human computational power to a new stage, and the existing artificial intelligence greatly relies on cloud platforms. The implementation of autonomous driving relies on a large amount of data processing, and the lag processing generated by cloud platform services in the face of such a large amount of data will lead to uncertainty in the transmission process, thus bringing huge transmission pressure and unstable delays to communication networks, which will undoubtedly pose a huge threat to the safety of intelligent driving technologies such as driver-assisted unmanned driving. As a result, people have started to focus on edge computing, distributing the tasks of cloud computing to individual edge devices. Meanwhile, due to the substantial limitation of computing power, researchers have turned their attention to the field of lightweight network research, such as MobileNet [7, 8, 19], in addition to which, GhostNet [6] introduced by Huawei has gained the favor of researchers. Zhang et al. Introducing Ghost into Yolov5 to Improve the Accuracy of Traffic Sign Detection [32]. At the same time, a series of lightweight methods have also developed rapidly, such as weight quantization [26] and knowledge distillation [27], for embedded devices on the implementation of real-time detection [9–12, 14, 15, 20, 23] laid the foundation, such as Ayachi et al. incorporated the lightweight network squeezeNet into YOLO, which achieved a breakthrough in speed and accuracy [24]. Furthermore, many researchers have also made contributions to lightweight approaches [33, 34].

To address the detection of small objects and achieve high accuracy and high-speed detection, Yolov8-ghost-EMA is used in this study. The rest of this paper contains several sections following several contents. In Sect. 2, the research methodology is analyzed, which contains the selection of the data set and the network structure of Yolov8. In Sect. 3, the training experimental model comparison is described and the model effect is tested in practice. In Sect. 4, the research content of this paper is summarized and discussed.

2 Research methods

2.1 YOLOv8

2.1.1 Structure

The structure of yolov8 is shown in Fig. 1.

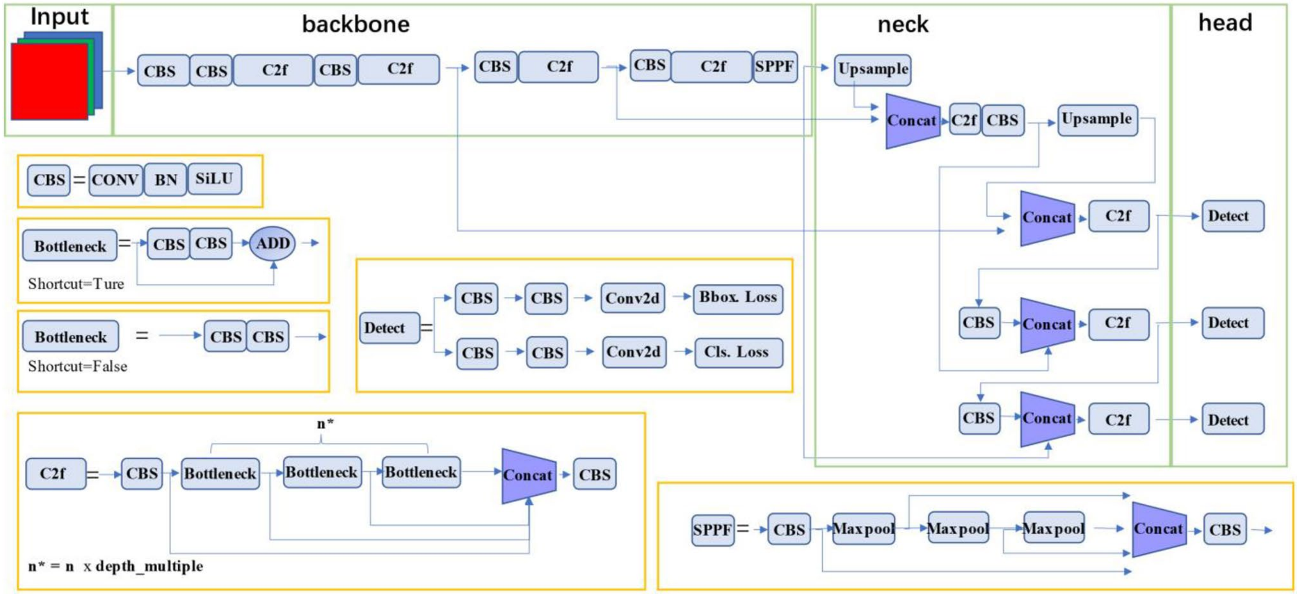


Fig. 1. Structure of YOLOv8

2.1.2 Loss function

The loss calculation process consists of two parts: positive–negative sample allocation strategy and loss computation.

The loss function of YOLOv8 consists of two components: classification loss and localization loss.

•Classification loss is used to calculate whether the anchor is correctly classified with the corresponding calibration.

•Regression loss shows the error in the position between the predicted anchor and the calibrated anchor. It includes CIoU Loss and Distribution Focal Loss.

YOLOv8 uses BCEWithLogitsLoss to calculate the classification loss, which is calculated as follows:

$$\mathrm{Loss} = -\frac{1}{n}\sum_{i}^{n}\left[y_i \log\left(\sigma\left(x_i\right)\right) + \left(1-y_i\right)\log\left(1-\sigma\left(x_i\right)\right)\right] \tag{1}$$

$$\sigma(a) = \frac{1}{1+\exp(-a)} \tag{2}$$

In the regression loss function, one component is the CIoU Loss [22], which takes into account many factors and more comprehensively describes the regression of the bounding box. It is calculated as follows:

$$L_{\mathrm{CIoU}} = 1 - \mathrm{IoU} + \frac{\rho^2\left(b, b^{\mathrm{gt}}\right)}{c^2} + \alpha v \tag{3}$$

$$\mathrm{IoU} = \frac{\left|b \cap b^{gt}\right|}{\left|b \cup b^{gt}\right|} \tag{4}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{5}$$

where b, $b$gt are the prediction box and label box, respectively. $w$gt, $h$gt, w, h are the width and height of the label box and the width and height of the prediction box, respectively. ρ represents the distance between the centroids of the two boxes calculated. c is the farthest distance between the boundaries of the two boxes. α is the weighting factor.

At the same time, yolov8 also employs Distribution Focal Loss as part of its regression loss.

2.2 Improved algorithm model YOLOv8-ghost-EMA

2.2.1 Ghostnet

Due to the advantage of Ghostnet in executing algorithms on embedded devices, in this paper, the backbone part in Yolov5s is replaced with GhostNet to convert the original model into a compact model while maintaining comparable performance. In the backbone of Yolov5s, Ghost Module is added, as seen in Fig. 2. First, the input features are convolved to generate feature maps, then the generated feature maps are convolved with depthwise separable convolution, and finally the generated feature maps are in contact with the result of depthwise separable convolution to get the output. Ghost Module divides the original convolution layer into two parts and generates multiple intrinsic feature maps using fewer filters; then Ghost Module is introduced into Ghost Bottlenecks, as in Fig. 3. When stride is 1, the input goes through Ghost Bottlenecks + BN Relu module once, then

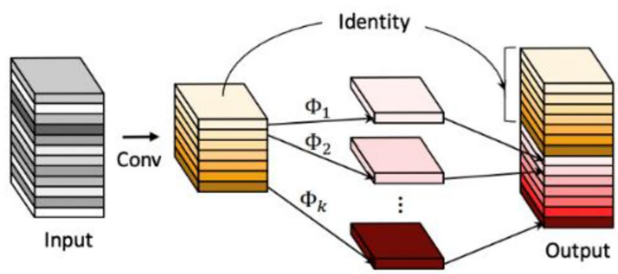


Fig. 2. The ghost module

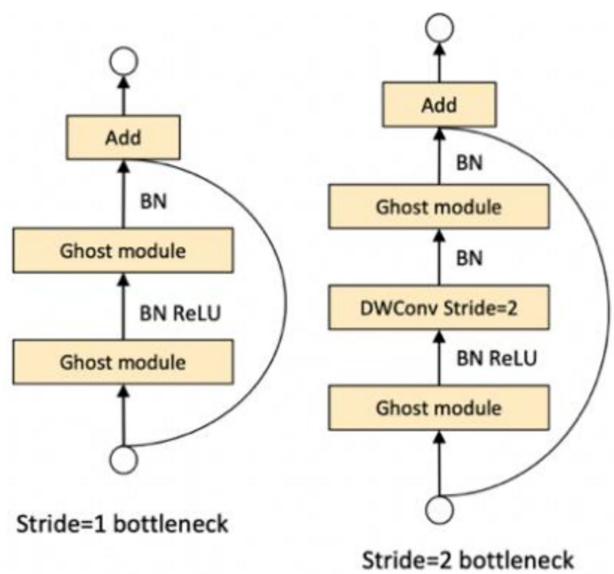


Fig. 3. Ghost bottleneck. Left: Ghost bottleneck with stride = 1; right: Ghost bottleneck with stride = 2

Ghost Bottlenecks + BN once, and finally, the obtained feature map is added with the initial input; When stride is 2, the input goes through the Ghost Bottlenecks + BN Relu module once, then through a depthwise separable convolution with stride 2, then through Ghost Bottlenecks + BN once, and finally, the obtained feature map is added with the initial input to get the output.

2.2.2 Efficient multi-scale attention module with cross-spatial learning

The attention mechanism is similar to our human eyes. Our eyes are always able to extract local, crucial information from complex environments. The attention mechanism shifts the focus from global information to local, critical information. In practice, attention mechanisms help deep learning models in computer vision tasks to prioritize local crucial information while ignoring less important details. Some of the most commonly used

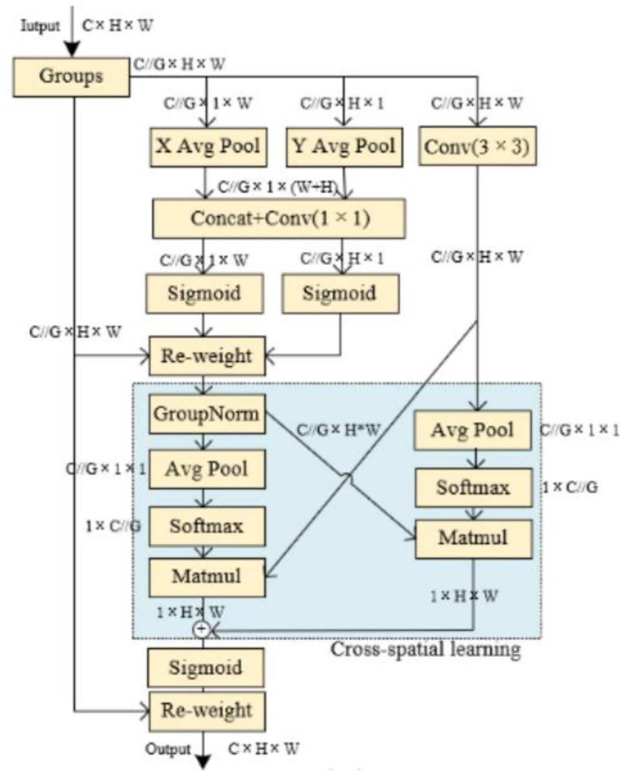


Fig. 4 Efficient multi-scale attention

attention mechanisms include Senet, CBAM, Coordinate Attention (CA), among others. CA considers relationships between channels and spatial position information. However, it overlooks the importance of interactions among spatial positions across the entire space, and the limited receptive field of 1 × 1 kernel convolution hinders the modeling and utilization of local cross-channel interactions and context information. Therefore, we propose efficient multi-scale attention (EMA).

Figure 4 illustrates the composition framework of EMA, and detailed principles can be found in the paper [28]. The research demonstrates that EMA outperforms other attention mechanisms [29–31]. EMA aims to preserve information from each channel while reducing computational redundancy. It reshapes some channels into the batch dimension and groups the channel dimension into multiple sub-features, allowing spatial semantic features to be well distributed within each feature group. In addition to encoding global information to recalibrate channel weights in each parallel branch, it further aggregates the output features of the two parallel branches through cross-dimension interactions to capture pixel-level pairwise relationships (Fig. 5).

In summary, the improved YOLOv8 model structure is shown in Fig. 5 after combining the above improvements.

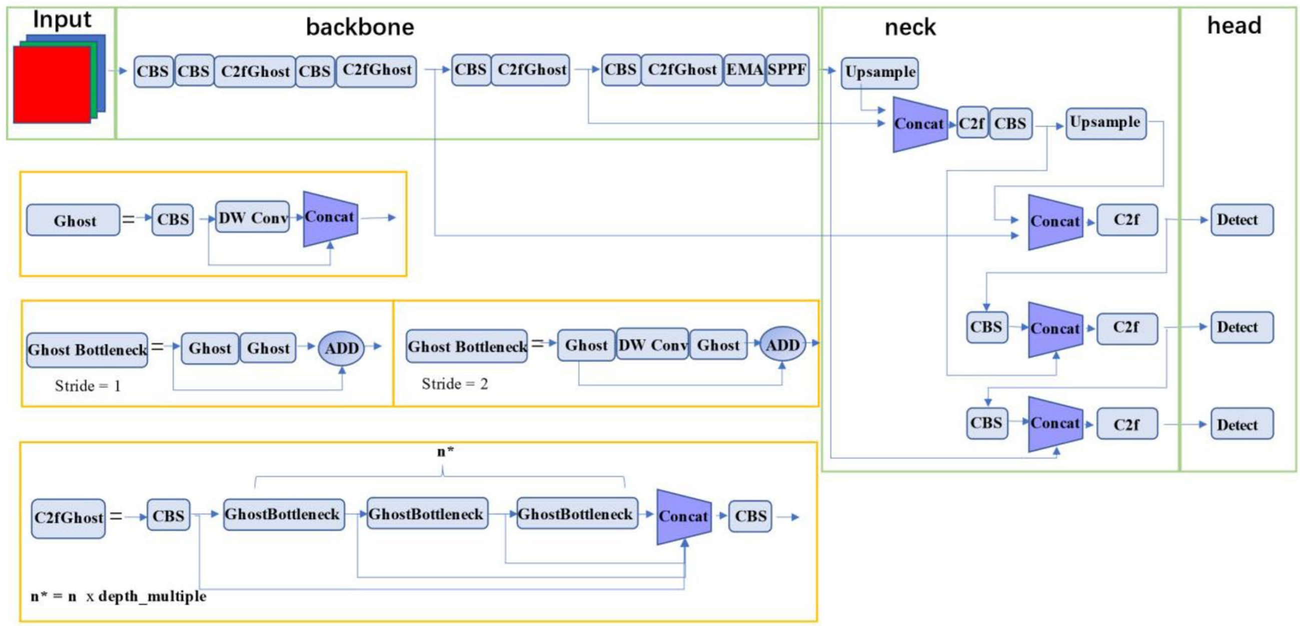


Fig. 5 Structure of YOLOv8-ghost-EMA

3 Experiment and conclusion

In this paper, after completing the improvement of all YOLOv8, the proposed traffic sign recognition algorithm is Yolov8-ghost-EMA, and the traffic sign recognition system with Raspberry Pi as the development platform is completed. This chapter first introduces the datasets, the experimental environment, and then the training results are described.

3.1 Traffic sign detection dataset

The traffic sign dataset comes from CCTSDB China traffic dataset, which is produced by Zhang Jianming's team at the Hunan Provincial Key Laboratory of Comprehensive Transportation Big Data Intelligent Processing of Changsha University of Science and Technology, and its images are taken from Chinese street scenes by car recorders with a resolution of 1280 × 720. A sample CCTSDB traffic sign dataset is shown in Fig. 6.

There are 17,856 images in the CCTSDB dataset, including 16,356 images in the training set and 1500 images in the test set, which calibrates the locations of three categories of traffic signs: prohibition, indication, and warning. The distribution of the number of each category in the training set is shown in Fig. 7.

Fig. 6. Examples of CCTSDB dataset

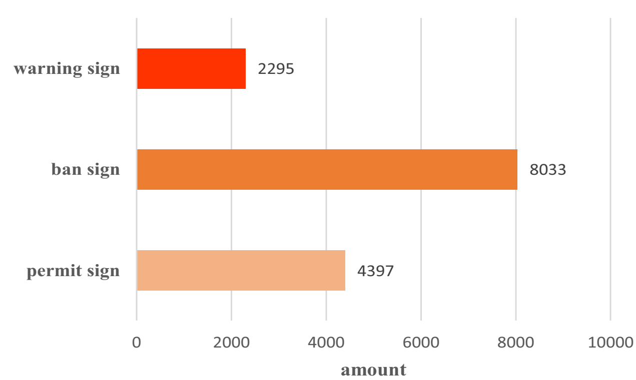


Fig. 7. CCTSDB dataset number of signs by type

After filtering the test set of the dataset, some images with high repetition and useless images were removed, and finally, 1 test set containing 325 images of traffic signs under poor weather conditions was filtered out.

3.2 Experimental environment

3.2.1 Training equipment

This paper uses the cloud platform autodl as the training device, the graphics card model used in this paper is NVIDIA GEFORCE RTX 2080 Ti, 11 GB of current memory, python 3.8 development, Pytorch 1.9.0 deep learning framework, GPU acceleration tool is Cuda 11.1, and the compilation environment is pycharm.

3.2.2 Embedded devices selection

With the increase in computing power of lightweight embedded devices, they have been able to meet most of the inference tasks of deep learning. The YOLOv5s-ghost traffic sign recognition algorithm designed in this paper is applied to a low-cost, low-power, small embedded device, which is then loaded on a car, thus contributing to the popularity of this system.

At present, the mainstream embedded products in the market include Raspberry Pi, NVIDIA series products, etc. Since the traffic sign detection algorithm in this paper is real-time detection of video streams, it is difficult to realize the computational power of a single-chip computer. In addition, although the NVIDIA series products have integrated NVIDIA CUDA-based GPU, which has a very fast computing speed, it comes with a high cost.

Raspberry Pi [3, 5] is a general-purpose embedded device with microcomputer control in the industry, which also integrates various resources such as sensing and communication, with higher performance than microcontrollers and lower cost than NVIDIA products, featuring lightweight, low-power consumption, powerful performance, and low cost. In addition, Raspberry Pi edge computing nodes can offload tasks to their neighboring nodes to reduce their load, lower latency, and improve resource utilization. Raspberry Pi can be optionally combined with Inter Neural Computation Stick 2(NCS 2) at a later stage, which can significantly increase the computational power, so Raspberry Pi 4B is selected as the final device in this study.

Fig. 8. Raspberry Pi 4B

Table 1 Raspberry Pi 4B hardware parameters

Parameters	Specifications
CPU	64-bit ARMv8
Bluetooth	Bluetooth 5.0
GPU	500 MHz VideoCore VI
USB port	2 USB 3.0 ports; 2 USB 2.0 ports
Wired network	Gigabit Ethernet
Wireless	80211ac(2.4/5 GHz)
Input power	3A,5 V
Memory	1-4 GB DDR4

Table 2 Raspberry Pi 4B hardware parameters

Model	mAP (%)	*P* (%)	*R* (%)
Baseline	79.1	89.0	70.0
Ghost	79.7	87.5	71.2
EMA	79.9	88.6	71.2
Ghost + EMA	82.9	89.4	71.9

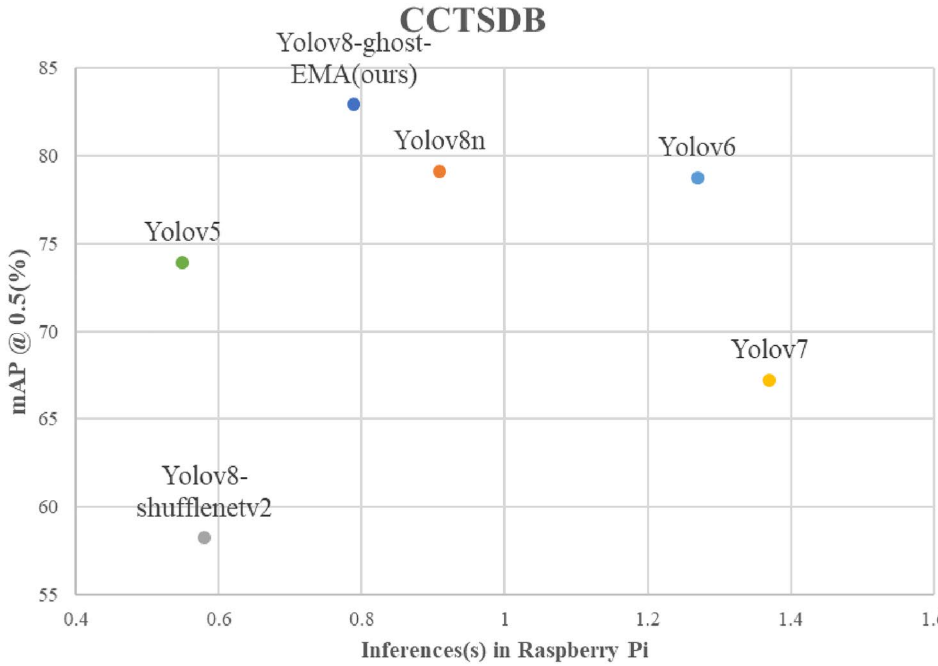
The Raspberry Pi 4B development board is shown in Fig. 8, and its hardware parameters are shown in Table 1.

3.3 Experimental results

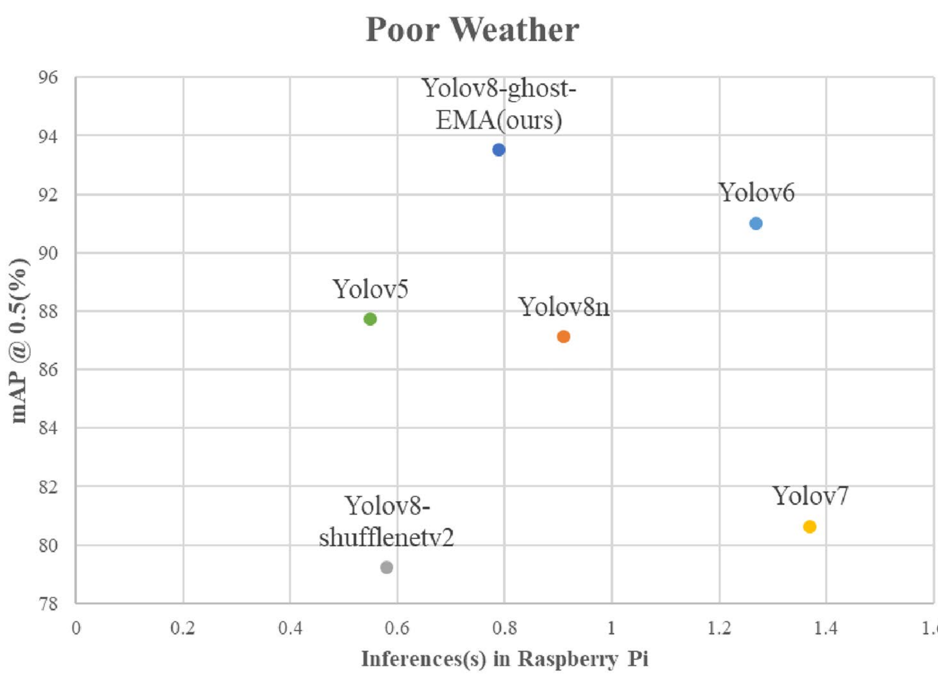
3.3.1 Ablation experiment

To validate the effectiveness of the two improvement methods, GhostNet and EMA, we conducted ablation experiments based on the CCTSDB dataset. We used the YOLOv8n network as the baseline and combined

Fig. 9. Examples of traffic sign recognition



(a)



(b)

Fig. 10. Detection speed in Raspberry Pi comparison of different test sets: **a** CCTSDB and **b** FPS performance

GhostNet and EMA to assess their respective contributions to network enhancement. The evaluation metrics included Precision, Recall, and mAP @0.5, with a resolution set at 640×384. The experimental results are presented in Table 2.

Comparing YOLOv8 with the Ghost and EMA models, it is evident that both Ghost and EMA contribute to the improvement of the model's performance, with increases of 0.6% and 0.8% in mAP (mean Average Precision) respectively. YOLOv8, when incorporating all of these

Table 3 Comparison of traffic sign detection algorithm indicators

Methods	mAP @ 0.5(%)		FLOPs (G)	Weight size (MB)	Inferences(s) in Raspberry Pi
	Poor weather	CCTSDB			
Yolov8-ghost-EMA(ours)	93.5	82.9	6.9	5.2	0.79
Yolov8n	87.1	79.1	8.1	6.0	0.91
Yolov8-Shufflenetv2	79.2	58.2	5.1	3.7	0.58
Yolov7	80.6	67.2	6.9	4.8	1.37
Yolov6	91.0	78.7	11.3	4.6	1.27
Yolov5	87.7	73.9	4.1	3.7	0.55

The size of the input images are 480*640 when reasoning in the text

improvement methods, achieves the best detection results, showing optimal Precision, Recall, and mAP scores. Compared to the baseline model, YOLOv8 demonstrates a 3.8% increase in mAP. Based on the results of ablation experiments, it is clear that the improvements brought by Ghost and EMA are effective.

3.3.2 Comparative experiments

To compare the Yolov8-ghost-EMA proposed in this paper with some of the current table mainstream object detection algorithms, Yolov8, Yolov8-shufflenetv2, Yolov7, Yolov6, and Yolov5 were trained, respectively. Figure 9 shows an example of traffic sign recognition, Fig. 10 compares the inference speed and accuracy and Table 3 gives a comparison of traffic sign detection algorithm metrics.

It can be seen that the accuracy of Yolov8-ghost-EMA proposed in this paper reaches 82.9% on the original CCTSDB test set with mAP@0.5 and up to 93.5% on the test set with poor weather conditions, and the inference time on the Raspberry Pi reaches 0.79 s. We can see that the detection accuracy is highest among all models and the inference time in Raspberry Pi 4 is only 0.79 s. Moreover, the computational complexity and parameter count of our model have reduced by 14.9% and 13.3% compared to the original model. In comparison to Yolo7, Yolov6, and Yolov5 models, our model stands out as the top choice in terms of both detection accuracy and speed. This lays a solid foundation for real-time traffic sign detection.

In addition, to confirm the generalization capability of our model in this study, we conducted tests on the TT100k dataset's test set. We selected 300 relatively common images and achieved an accuracy of 99.4%. This confirms that our model exhibits strong generalization performance in typical traffic scenarios and can adapt to variations in dataset sizes. It further validates the effectiveness of our algorithm.

4 Conclusion and discussion

In this paper, to solve the safety problems of autonomous driving and meet the inevitable demand of edge computing, first, to address the problems of large computational volume and slow computational speed of Yolov8 feature network, an improved algorithm Yolov8-ghost-EMA is proposed, replacing the main modules of the network with lightweight Ghost Bottlenecks modules and introducing the attention mechanism; secondly, based on CCTSDB dataset, the training optimization of the model is completed on the cloud platform; then, the inference process is implemented on the Raspberry Pi 4B platform. The results show that the Yolov8-ghost-EMA model has a memory occupation of 5.2 M, mAP of 82.9%, and inference time of 0.79 s on Raspberry Pi. Compared with the current popular lightweight Yolov8, the model compression is 13.3%, mAP is 3.7% upper, and inference time is 10.9% lower, which is an obvious advantage. On the poor weather test set, the new algorithm achieves 93.5% recognition accuracy. The tests of dynamic traffic video and real-time camera acquisition also achieved good results and verified the effectiveness of the new model.

In this paper, the Raspberry Pi embedded traffic sign detection algorithm is proposed and performance advantages are obtained, but the actual traffic environment is complex and the inference delay of the algorithm proposed in this paper is still some distance away from real-time detection, so based on this paper, future research outlook is for example:

Use NVIDIA series products as the test platform, and use the unique GPU acceleration advantage of NVIDIA products to improve the inference speed.

Raspberry Pi is widely used in the field of IoT, with good generality and low price. In the future, we can introduce Intel Neural Compute Stick 2 to accelerate its inference speed, and consider the synergy between multiple devices to jointly form an IoT network to improve edge computing.

Consider complex environmental factors, introduce some image pre-processing means to do some processing of images under abnormal light, blur, etc. With the popularity of the Transformer algorithm, some related algorithms should be introduced in the future.

Acknowledgements This work was financially supported by Natural Science Foundation of Heilongjiang Province of China (Grant no. LH2023E055).

Author contributions YL: methodology, software, data curation, writing—original draft preparation. YC: conceptualization, methodology, writing—reviewing, investigation. SJ: software, data curation. XW: Wring.

Data availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest All authors declare that they have no conflicts of interest affecting the work reported in this article.

References

1. Artamonov, N.S., Yakimov, P.Y.: Towards real-time traffic sign recognition via YOLO on a mobile GPU. J. Phys. Conf. Ser. **1096**, 012086 (2018). https://doi.org/10.1088/1742-6596/1096/1/012086
2. Bahlmann, C., Zhu, Y., Visvanathan, R., Pellkofer, M., Koehler, T.: A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. Paper presented at the IEEE Proceedings. Intelligent Vehicles Symposium. (2005)
3. Chakravarthy, K.K., Krishna, M.V.: Color objects detection in real-time with raspberry Pi and image processing. SAMRIDDHI **13**(01), 5–7 (2021)
4. Girshick, R.: Fast r-cnn. Paper presented at the Proceedings of the IEEE international conference on computer vision (2015)
5. Gunnarsson, A.: Real time object detection on a Raspberry Pi. In (2019)
6. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C.: Ghostnet: more features from cheap operations. Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2020)
7. Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Vasudevan, V.: Searching for mobilenetv3. Paper presented at the Proceedings of the IEEE/CVF international conference on computer vision (2019)
8. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Adam, H.: Mobilenets: efficient convolutional neural networks for mobile vision applications (2017)
9. Jana, S., Borkar, S., Student, M.: Autonomous object detection and tracking using Raspberry Pi. Int. J. Eng. Sci. 141415 (2017)
10. Jiang, Z., Zhao, L., Li, S., Jia, Y.: Real-time object detection method for embedded devices. Paper presented at the computer vision and pattern recognition (2020)
11. Khoi, T. Q., Quang, N. A., Hieu, N. K.: Object detection for drones on Raspberry Pi potentials and challenges. Paper presented at the IOP Conference Series: Materials Science and Engineering (2021)
12. Lopez-Montiel, M., Orozco-Rosas, U., Sánchez-Adame, M., Picos, K., Ross, O.H.M.J.I.A.: Evaluation method of deep learning-based embedded systems for traffic sign detection. IEEE Access **9**, 101217–101238 (2021)
13. Maldonado-Bascon, S., Lafuente-Arroyo, S., Gil-Jimenez, P., Gomez-Moreno, H., Lopez-Ferreras, F.: Road-sign detection and recognition based on support vector machines. IEEE Trans. Intell. Transp. Syst. **8**(2), 264–278 (2007). https://doi.org/10.1109/TITS.2007.895311
14. Othman, N. A., Aydin, I.: A new deep learning application based on movidius ncs for embedded object detection and recognition. Paper presented at the 2018 2nd international symposium on multidisciplinary studies and innovative technologies (ISMSIT) (2018)
15. Othman, N. A., Salur, M. U., Karakose, M., Aydin, I.: An embedded real-time object detection and measurement of its size. Paper presented at the 2018 International Conference on Artificial Intelligence and Data Processing (IDAP) (2018)
16. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition (2016)
17. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition (2017)
18. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement (2018)
19. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: Mobilenetv2: Inverted residuals and linear bottlenecks. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition (2018)
20. Souki, M. A., Boussaid, L., Abid, M.: An embedded system for real-time traffic sign recognizing. Paper presented at the 2008 3rd International Design and Test Workshop (2008)
21. Xie, Y., Liu, L.-f., Li, C.-h., Qu, Y.-y.: Unifying visual saliency with HOG feature learning for traffic sign detection. Paper presented at the 2009 IEEE Intelligent Vehicles Symposium (2009)
22. Yu, J., Jiang, Y., Wang, Z., Cao, Z., Huang, T.: Unitbox: an advanced object detection network. Paper presented at the Proceedings of the 24th ACM international conference on Multimedia (2016)
23. Ayachi, R., Afif, M., Said, Y., Ben Abdelali, A.: An edge implementation of a traffic sign detection system for advanced driver assistance systems. Int. J. Intell. Robot. Appl. (2022). https://doi.org/10.1007/s41315-022-00232-4
24. Li, C.L., Su, C.Y.: Traffic signs detection based on enhanced YOLOv5 network model. In 2022 IEEE International Conference on Consumer Electronics-Taiwan (pp. 449–450). IEEE (2022)
25. Liang, T., Bao, H., Pan, W., Pan, F.: Traffic sign detection via improved sparse R-CNN for autonomous vehicles. J. Adv. Trans. (2022). https://doi.org/10.1155/2022/3825532
26. Han, S., Mao, H., Dally, W.J.: Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149. (2015)
27. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531. 2(7) (2015)
28. Ouyang, D., He, S., Zhang, G., Luo, M., Guo, H., Zhan, J., Huang, Z.: Efficient multi-scale attention module with cross-spatial learning. In ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1–5). IEEE (2023)
29. Woo, S., Park, J., Lee, J.Y., Kweon, I.S. Cbam: convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV) (pp. 3–19) (2018)
30. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7132–7141) (2018)
31. Hou, Q., Zhou, D., Feng, J.: Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 13713–13722) (2021)
32. Zhang, S., et al.: A real-time and lightweight traffic sign detection method based on ghost-YOLO. Multimed. Tools Appl. (2023). https://doi.org/10.1007/s11042-023-14342-z
33. Golcarenarenji, G., et al.: Robust real-time traffic light detector on small-form platform for autonomous vehicles. J. Intell. Trans. Syst. (2023). https://doi.org/10.1080/15472450.2023.2205018
34. Ou, Y. et al.: "Traffic signal light recognition based on transformer." International Conference on Computer Engineering and Networks. Singapore: Springer Nature Singapore (2022)

35. Lou, H., Duan, X., Guo, J., Liu, H., Gu, J., Bi, L., Chen, H.: DC-YOLOv8: small-size object detection algorithm based on camera sensor. Electronics **12**(10), 2323 (2023)
36. Terven, J., Cordova-Esparza, D.: A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. arXiv preprint arXiv:2304.00501 (2023)
37. Talaat, F.M., ZainEldin, H.: An improved fire detection approach based on YOLO-v8 for smart cities. Neural Comput. Appl. Appl. (2023). https://doi.org/10.1007/s00521-023-08809-1
38. Aboah, A., Wang, B., Bagci, U., Adu-Gyamfi, Y.: Real-time multi-class helmet violation detection using few-shot data sampling technique and yolov8. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5349–5357) (2023)
39. Yang, G., Wang, J., Nie, Z., Yang, H., Yu, S.: A lightweight YOLOv8 tomato detection algorithm combining feature enhancement and attention. Agronomy **13**(7), 1824 (2023)
40. 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)
41. Huang, Z., Li, L., Krizek, G.C., Sun, L.: Research on traffic sign detection based on improved YOLOv8. J. Comput. Commun. **11**(7), 226–232 (2023)
42. Kumar, D., Muhammad, N.: Object detection in adverse weather for autonomous driving through data merging and YOLOv8. Sensors (2023). https://doi.org/10.3390/s23208471