Yolov8-CS: A Traffic Sign Detection Algorithm Based on Improved Yolov8

Jiashu Han [1, *], Yueru Li [2], Danna Wang [3]

[1] College of Mechanical and Electrical Engineering, Hohai University, Jiangsu, China, 213000

[2] College of Traffic and Logistics Engineering, Wuhan University of Technology, Wuhan, China, 430063

[3] College of Software Engineering, Xinjiang University, Xinjiang, China, 830046

* Corresponding Author Email: 2261710219@hhu.edu.cn

Abstract. In the context of the rapid advancements being made in autonomous driving technology, traffic sign detection has emerged as a pivotal technology to ensure safe driving. However, existing traffic sign detection algorithms are beset with challenges, including difficulties in feature extraction and model bias when identifying small targets and dealing with imbalanced data. To address these issues, this paper proposes an enhanced algorithm, termed YOLOv8-CS (CBAM, SlideLoss), which is based on YOLOv8 and incorporates the CBAM (Convolutional Block Attention Module) attention mechanism to enhance the model's capacity to identify key features. This, in turn, leads to an improvement in the detection accuracy for small traffic signs. Furthermore, the loss function in the YOLOv8 model is enhanced, and SlideLoss is employed to augment the model's capacity to handle imbalanced data, ensuring the model focuses more on challenging cases in data training. The experimental results demonstrate that, in comparison with the original YOLOv8 model, the enhanced YOLOv8-CS model exhibits a 6.1% increase in accuracy, a 4.4% increase in recall rate, and a 5.1% increase in mAP. Furthermore, the enhanced model demonstrates a comparable training speed to the original model, thereby substantiating its viability for practical applications.

Keywords: Traffic Sign Detection, YOLOv8, CBAM, SlideLoss.

1. Introduction

With the continuous progress of science and technology, unmanned technology has become the focus of attention, in which target detection technology has gradually become an important area of academic research. There are various research directions in traffic target detection, and one of the most important directions is traffic sign detection and recognition for complex and diverse real traffic environments. In traffic scenarios, target detection algorithms need to consider both efficiency and accuracy, but on roads with complex backgrounds, they often face problems such as a large number of signs with chaotic distributions, as well as long distances between the signs and the camera, small targets, and large changes in angle. The small target problem can lead to misses or false detections, which can seriously affect the detection accuracy. Therefore, it is of great practical significance to solve the small target detection problem in complex road scenes.

To address the above challenges, many researchers have proposed appropriate improvements. Yang et al[1] accelerated the convergence speed of the model in traffic sign processing by improving the image data pre-processing as well as optimising the relevant network parameters based on YOLOv5, and achieved better results in various experiments. Chen[2] proposed a method to improve YOLOv8 by using a lightweight feature extraction module MixGhost and a lightweight recognition head SEG-Head, which improves the accuracy of the model in traffic sign detection. Chen et al[3] proposed a traffic sign detection framework based on improved YOLOv8s, which improves the average microtarget mean accuracy and reduces the number of parameters. Du et al[4], by adopting the Select Kernel attention mechanism and other optimisation methods, proposed the TSD-YOLO model, which improved the feature extraction ability of the model for different traffic signs. Cui et al[5] applied the method of halving the sampling multiplier in the YOLO backbone and introducing a feature fusion module to improve the performance of the model in detecting multiple traffic signs.

Although these methods improve the performance of traffic sign detection to a certain extent, YOLOv8 still has limitations in small target detection and category imbalance problems, especially in complex backgrounds, it is still difficult to effectively detect small traffic signs. Therefore, how to further improve the model's small target detection ability and how to effectively deal with the data imbalance problem are still pressing challenges in current traffic sign detection research.

In this paper, we improve and propose an improved algorithm YOLOv8-CS based on YOLOv8 by introducing the CBAM attention mechanism to enhance the feature representation capability of the model and adopting the SlideLoss loss function approach. This method solves the data imbalance problem and the small target detection problem. YOLOv8-CS has a significant performance improvement on dealing with small target detection and category imbalance problems in complex traffic environments has a significant performance improvement compared to the traditional YOLOv8 model. This method contributes to the development of driverless technology by improving the accuracy of traffic sign detection.

2. YOLOv8 Network Improvement

2.1. YOLOv8 Network Introduction

YOLOv8 is a high-performing SOTA (State-Of-The-Art) model in the YOLO series. It provides a flexible and powerful unified framework that not only supports the training of real-time object detection tasks, but also meets the requirements of other tasks such as instance segmentation and image classification[6]. Compared with previous generations, YOLOv8 has been optimized and innovated in network structure, with characteristics such as depth-widening, lightweighting, and multi-scale feature fusion. These features allow YOLOv8 to maintain high accuracy while further improving detection speed. This paper focuses on its improvement and application in the field of traffic target detection. The YOLOv8 network architecture consists of three main parts: Backbone, Neck and Head.

Backbone is composed of 5 convolutional modules, 4 C2f modules, and 1 SPPF (Spatial Pyramid Pooling Fast) module. The convolutional module includes 2D convolution, BatchNorm2d, and SiLU activation function. The backbone network uses convolutional layers to extract features and optimizes performance by combining residual connections and bottleneck structures. The C2f module and the SPPF module are important for extracting features from the model.

Neck[7] is a key component of the model, playing an important role in feature extraction and fusion. The Neck part of YOLOv8 is based on the FPN-PAN idea. FPN fuses high-level and low-level features through upsampling, while PAN is an improvement on FPN, introducing horizontal connections to enhance feature semantic information. YOLOv8's Neck specifically includes one SPPF module, one PAA module, and two PAN modules to achieve the fusion of feature maps at different stages and enhance feature representation capabilities.

When viewed as the preparatory stages, the Backbone and Neck represent the initial phase, with the Head representing the output phase. This module performs the terminal processing stages of object detection and classification through its dual-head architecture, comprising detection and classification components. The detection component incorporates sequential convolutional and deconvolutional layer stacks to generate bounding box predictions, whereas the classification unit applies global average pooling followed by softmax activation for feature map categorization.

The YOLOv8 network structure diagram is shown in Figure 1.

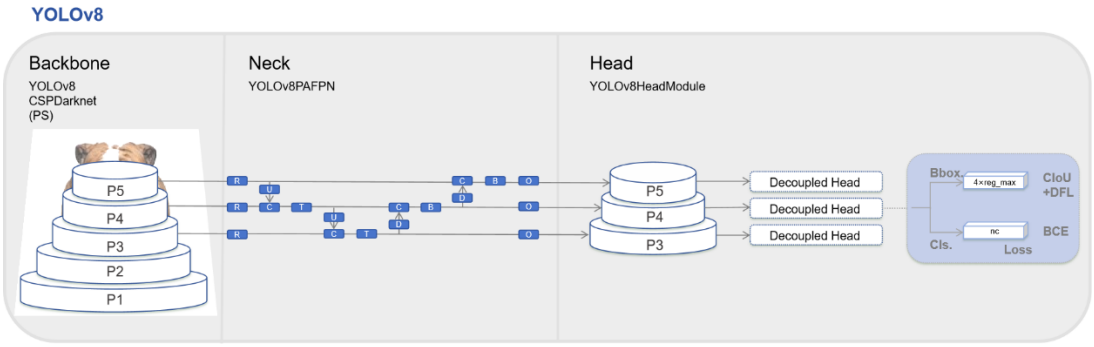

Figure 1. Yolov8 network structure diagram

2.2. CBAM Attention Mechanism

In the task of traffic sign detection, faced with complex traffic scenes, it may be difficult to effectively extract the features of traffic signs due to background noise[8]and various visual disturbances[9], and there are deficiencies in detecting small target traffic signs. This study develops an attention-enhanced YOLOv8 architecture, where convolutional block attention modules (CBAM) are strategically embedded to amplify feature discriminability in critical regions.

Evolving from SENet's channel attention, CBAM (Convolutional Block Attention Module)[10] introduces spatial attention gates that perform cross-dimensional feature modulation, forming the first unified C-S attention framework in computer vision. The channel attention module serves to identify and amplify feature channels that are relevant to traffic signs, computing the average and maximum responses of the feature map. This statistical information is then used to generate attention weights across the channel dimension through a simple neural network. This enables the model to focus on feature channels related to traffic signs. On the other hand, the spatial attention module emphasizes the spatial distribution within the feature map by examining the relationships between different locations and producing attention weights in the spatial dimension. This procedure includes pooling operations on the feature map, followed by convolution layers to generate a spatial attention map, allowing the model to more precisely identify the location of the traffic sign. The CBAM attention mechanism is illustrated in Figure 2.

The output $M_c(F)$ of the channel attention module can be computed using the following equation:

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \quad (1)$$

In Equation 1, F denotes the input feature map, AvgPool and MaxPool represent the global average pooling and maximum pooling operations, respectively, and MLP signifies Multilayer Perceptron, while σ denotes the Sigmoid activation function.

The output Ms(F) of the spatial attention module can be computed using the following equation:

$$M_s(F) = \sigma(f^{7\times7}([AvgPool(F); MLP(MaxPool(F)])) \quad (2)$$

In Equation 2, $f^{7\times7}$ represents the 7×7 convolution operation, while [AvgPool(F); MLP(MaxPool(F))] represents the concatenation of the outputs from average pooling and max pooling along the channel axis.

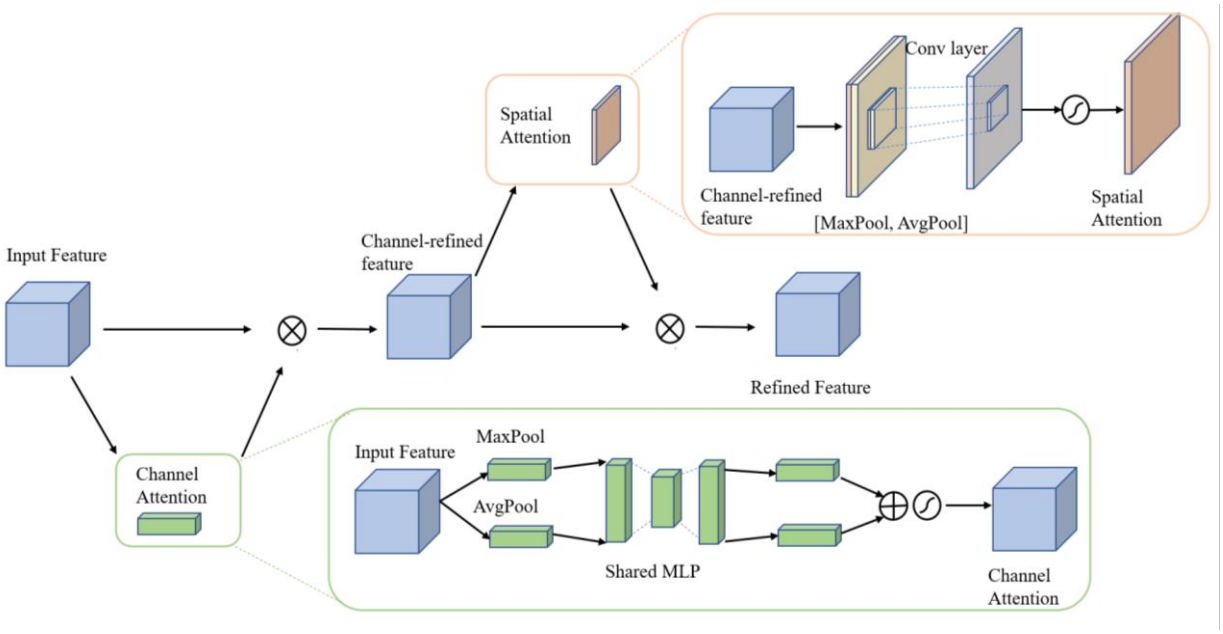

Figure 2. Illustration of the working principle of CBAM Feature Enhancement Module

2.3. Loss Function

During the training process of the traffic sign detection model, the model generally exhibits deficiencies in its handling of class imbalance[11]. In order to resolve this problem, the paper suggests employing SlideLoss as the loss function within the improved YOLOv8 model, aiming to improve the model's ability to manage class imbalance. Initially, the samples are segmented into positive and negative classes through the parameter μ. Subsequently, emphasis is placed on samples at the boundary through the implementation of the weighted function Slide, as illustrated in Figure 3.

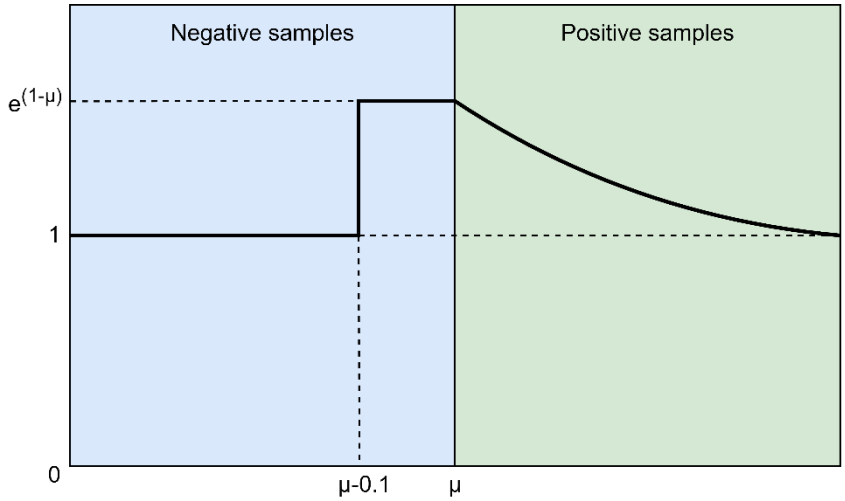

Figure 3. Processing of samples at the boundary

The Slide weighting function can be expressed by formula 3:

$$f(x)=\begin{cases}1 & x\le \mu-0.1\\ e^{1-\mu} & \mu<x<\mu-0.1\\ e^{1-x} & x\ge \mu\end{cases} \tag{3}$$

Therefore, the SlideLoss function can make the model pay more attention to difficult cases in the datasets, so that the model in this paper performs well in handling traffic sign sample imbalance and target smaller sign sample detection.

3. Results and discussion

3.1. Experimental Platform

The experimental platform for the traffic sign recognition algorithm utilises an Intel Xeon Platinum 8352V CPU and two NVIDIA RTX 3080 GPUs (each with 20GB video memory), while the software environment consists of Python 3.8 and the PyTorch 2.0.0 framework operating on an Ubuntu 20.04 system. The CUDA version employed is 11.2. In terms of the training configuration, the starting learning rate is set to 0.01, the momentum for the learning rate is 0.937, the batch size is 16, the weight decay factor is 0.0005, and 200 training epochs are executed in total.

3.2. Description of Experimental Datasets

This paper uses Tsinghua-Tencent 100K(TT100K)[11] as the experimental dataset, which is an unbalanced dataset. It contains a large number of traffic sign types, a total of 45. These 45 data are divided into three categories: yellow signs with letter codes starting with 'w', red signs with letter codes starting with 'p', and blue signs with letter codes starting with 'i'. However, the number of each type is not the same. Some traffic signs, such as the no parking sign with the code 'pn', have more than 2500 marked boxes, while the road construction sign with the code 'w32' has less than 100 marked boxes. Its distribution is roughly as shown in Figure 4.

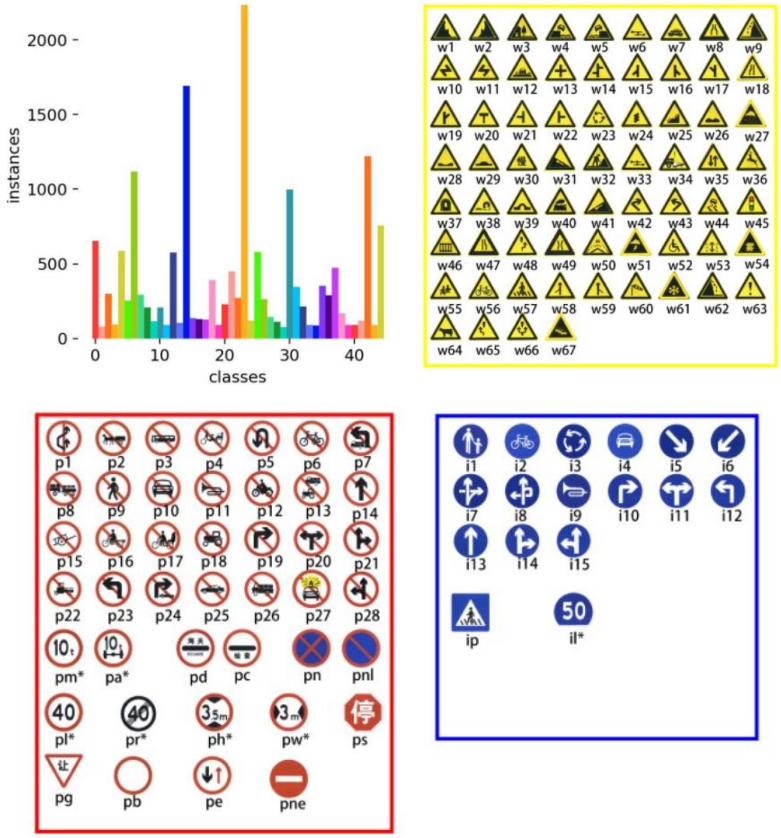

Figure 4. Distribution of datasets categories and display of traffic signs

3.3. Description of Evaluation Indicators

In the field of traffic sign recognition, it is crucial to thoroughly evaluate the performance of the model. This research utilizes various evaluation metrics such as precision (P), recall (R), mean average precision (mAP), mean average precision at 50% Intersection over Union (mAP50), and the F1-Score, which is the harmonic mean of precision and recall. The formulas for these metrics are presented in equations (4) to (8).

$$P = \frac{T_P}{T_P + F_P} \tag{4}$$

$$R = \frac{T_P}{T_P + F_N} \tag{5}$$

$$AP = \int_0^1 P(r)dr \tag{6}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{7}$$

$$F1 = \frac{2P \cdot R}{P + R} \tag{8}$$

The following definitions correspond to Equations (4) and (5): True Positives (TP) refer to cases where the instances are genuinely positive and are accurately classified as positive by the model. False Positives (FP) indicate situations where the actual instances are negative, but the model incorrectly labels them as positive. False Negatives (FN) describe instances that are truly positive but are misclassified as negative. True Negatives (TN) represent instances that are genuinely negative and correctly classified as such by the model.

3.4. Comparative Experiments

To demonstrate the superiority of the enhanced optimized model, a number of classic YOLO series models were selected on merit for comparison tests, comparing the original YOLOv8 (baseline), YOLOv3-tiny, YOLOv5, YOLOv6, YOLOv10, YOLOv11 and the YOLOv8-CS improved model in this paper, in the same dataset, the performance of the same epochs equal to 200 is obtained as shown in Table 1.

Table. 1. Comparative Experimental Results of Evaluation Indicators for Different Models

Algorithm	P	R	mAP50	mAP50-90	GFlops	Params/M	F1
YOLOv8[13]	0.706	0.635	0.706	0.534	8.1	3.0	0.669
YOLOv3-tiny[14]	0.705	0.488	0.552	0.418	19.0	12.2	0.577
YOLOv5[15]	0.675	0.631	0.681	0.516	7.1	**2.5**	0.652
YOLOv6[16]	0.579	0.578	0.600	0.448	12.0	4.3	0.578
YOLOv10[17]	0.706	0.592	0.670	0.515	8.3	2.7	0.644
YOLOv11[18]	0.698	0.633	0.689	0.530	**6.4**	2.6	0.664
YOLOv8-C	0.764	0.669	0.753	0.545	21.4	5.9	0.713
YOLOv8-S	0.728	0.653	0.695	0.520	8.1	3.0	0.688
YOLOv8-CS(Ours)	**0.767**	**0.679**	**0.757**	**0.564**	21.3	5.9	**0.720**

As presented in Table 1, during the comparison of different YOLO models' performance, YOLOv3-tiny has a smaller computational volume of 19.0GFlops, but it has a precision of 0.705, a recall of 0.488, and lower mAP and F1 scores, which is suitable for resource-constrained scenarios. YOLOv5 has a better performance in precision and recall, with a precision of 0.675, a recall of 0.631, mAP and F1 scores of 0.652, computation of 7. 1GFlops and a parameter count of 2.5M, which is fast but with poor precision. YOLOv6 is similar to YOLOv5, with an unsatisfactory balance between precision and efficiency, and does not perform well. YOLOv8 demonstrates a better balance between precision and recall, achieving a precision of 0.706 and a recall rate of 0.635. It has an mAP50 of 0.706, an F1 score of 0.669, a mediocre computational volume of 8.1GFlops and a parameter count of 3.0M. YOLOv10 is similar to YOLOv8, with close precision and recall, but with a slight drop in mAP and F1 scores, a computational volume of 8.3GFlops, and a parameter count of 2.7M. YOLOv11, although more compact, with 6.4GFlops of computation and 2.6M parameters, has a

precision of 0.688, a recall of 0.619, and slightly inferior mAP and F1 scores. In addition, two sets of experiments were conducted on the role of CBAM and SlideLoss. The first set involved the introduction of the CBAM module (YOLOv8-C), while the second set incorporated the introduction of the SlideLoss loss function (YOLOv8-S). The experimental results demonstrate that the P of the two groups of experiments reached 0.764 and 0.728, respectively, with all other indicators also showing improvement. This finding suggests that the incorporation of both the CBAM module and the SlideLoss loss function can enhance the model's performance, and that the combined effect of these two modules is more efficacious, thereby validating the improvements outlined in this study.

YOLOv8-CS excels in several metrics, with a precision of 0.767, a recall of 0.679, a mAP50 of 0.757 and an F1 score of 0.720, which is far superior to other YOLO models, especially for small target detection and complex scenes. Despite having a computational volume of 21.3GFlops and 5.9M parameters, which are larger than those of other models, it outperforms them in both precision and recall. This makes it well-suited for application scenarios that demand high precision and reliability, while also maintaining a good balance between performance and computational efficiency. However, due to the high computational complexity, YOLOv8-CS may affect real-time performance on resource-constrained devices, especially in scenarios requiring high inference speed, and further optimisation (e.g., model compression and knowledge distillation) may be required to achieve a better balance across multiple devices and scenarios.

Figure 5 illustrates the training convergence process of the improved YOLOv8-CS network, where the evolutionary trends of bounding box regression loss (train/box_loss), target confidence loss (train/cls_loss), and distribution focusing loss (train/dfl_loss) during the training phase are characterised by three independent curves, respectively. The corresponding performance metrics of the validation set are presented as three subplots with the prefix val, describing the trajectories of localisation loss, classification confidence loss and probability distribution loss, respectively, during the validation process. The horizontal axis of all loss curves corresponds to the number of model iterations, and the vertical axis scales are uniformly normalised loss values. In the performance evaluation module on the right side of the figure, the four evaluation metrics, namely accuracy, completeness, mean accuracy at 0.5 IoU threshold (mAP50) and multi-threshold integrated accuracy (mAP50-95) for the class B detection task are visualised by four blue curves of convergence characteristics with the number of training rounds, and the scale of their vertical axes is adaptively adjusted according to the range of values of each metric.

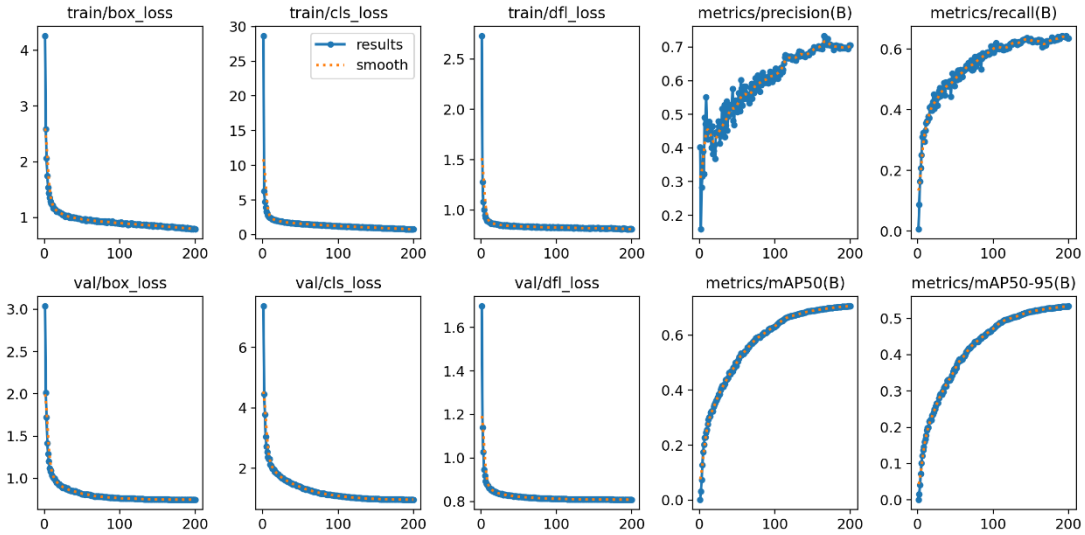

Figure 5. Training results of the YOLOv8-CBAM model

As observed in Figure 5, with the progression of training epochs, the YOLOv8-CS model shows good convergence. Whether it is various losses or accuracy and recall rate and other parameters, there

is no overfitting or underfitting, and they all stabilize at the optimal value around 200 epochs. This also explains the rationality of choosing 200 epochs as the number of training rounds in this paper.

Figure 6 illustrates the comparison of detection results prior to and following the improvements.

(a) Detecting '71559' with YOLOv8

(b) Detecting '71409' with YOLOv8

(c) Detecting '71559' with YOLOv3-tiny

(d) Detecting '71409' with YOLOv3-tiny

(e) Detecting '71559' with YOLOv5

(f) Detecting '71409' with YOLOv5

(g) Detecting '71559' with YOLOv6

(h) Detecting '71409' with YOLOv6

(i) Detecting '71559' with YOLOv10

(j) Detecting '71409' with YOLOv10

(k) Detecting '71559' with YOLOv11

(l) Detecting '71409' with YOLOv11

(m) Detecting '71559' with YOLOv8-CS

(n) Detecting '71409' with YOLOv8-CS

Fig. 6. Comparison of the detection effect of different YOLO series models

It is not difficult to see from Figure 6 that the effect after improvement has been significantly improved. As shown in Figures 6(a) and (m), for the image with the code '71559', the YOLOv8 original model detected the content as 'p11' with a confidence level of 27%; while the YOLOv8-CS improved in this paper also detected 'p11', but the confidence level increased to 47%. In Figures 6(b) and (n), the YOLOv8 original model failed to detect any traffic signs in the road image with the code '71409', but the improved YOLOv8-CBAM accurately detected the traffic sign 'p10'. From Figure 6(c) to Figure 6(l), it can be seen that the performance of other YOLO models is not as good as that of YOLOv8-CS. For example, YOLOv11, the latest SOTA model in the YOLO series, does not detect the traffic sign in '71559' and detects the wrong traffic sign type in '71409'. These improvements fully demonstrate the effectiveness of the improvement.

The confusion matrix is a visualised effect chart that can intuitively reflect the performance of the algorithm. A detailed examination of the confusion matrix allows for a clear understanding of the model's detection accuracy for each target class. The confusion matrix subsequent to training the enhanced model under consideration in this study is presented in Figure 7.

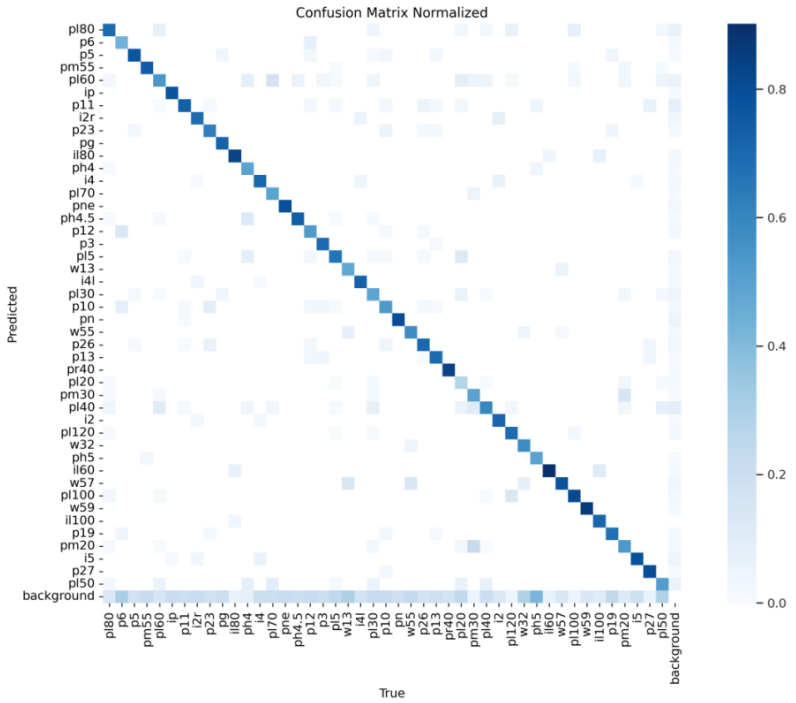

Figure 7. YOLOv8-CBAM confusion matrix

As shown in Figure 7, the enhanced model demonstrates high accuracy in detecting most traffic signs. For example, the accuracy for the 'pne' no entry sign is as high as 94.7%, and the accuracy for the 'pn' no parking sign is as high as 92.6%. This is mainly because these signs have simple graphics and high occurrence frequency, and the model has excellent performance. On the contrary, for the 'pm30' 30t weight limit sign, the accuracy is only 23.2%. This is mainly because the weight limit signs in the dataset are very similar to the height limit signs, and there are only numerical differences between different weight limit signs. In order to be realistic, the photo angle is usually not facing the sign, which leads to poor model training effect and inability to accurately identify these signs. Meanwhile, the sample size of the 'pm30' category is too small, which may also lead to insufficient training and affect the accuracy of the model's detection of this category.

4. Conclusions

This paper proposes an enhanced algorithmic model, termed YOLOv8-CS, to address the challenges posed by the recognition of small targets and category imbalance in traffic sign detection. The incorporation of the CBAM module boosts the model's ability to recognize important features, leading to improved performance in detecting small traffic signs. Concurrently, the enhanced loss function, designated as SlideLoss, has been demonstrated to effectively address the category imbalance issue, thereby prompting the model to prioritize complex cases and enhancing the detection accuracy. The TT100K dataset, which contains 45 different types of traffic signs and suffers from category imbalance, is used to validate the model. The experimental findings show that the upgraded YOLOv8-CS model surpasses the original YOLOv8 and other YOLO models in precision, recall, and average precision. Additionally, the model proves especially effective in identifying small target traffic signs.

Although the computational volume of the model has increased, the performance improvement far outweighs the impact of the increased computational volume. Future studies could focus on developing more lightweight YOLOv8 models to reduce both computational and memory requirements. Additionally, techniques like model compression and knowledge distillation can be explored to enhance the inference speed of YOLOv8-CS.

References

[1] Yang Xiaoling, Jiang Weixin, Yuan Haoran. Traffic Sign Recognition Detection Based on Yolov5 [J]. Information Technology and Informatization, 2021, (04): 28-30.

[2] Chen Guangjing. Traffic Sign Recognition Algorithm Based on Improved YOLOv8 [J]. Smart City, 2024, 10(03): 9-11.

[3] Chen Qibin, Deng Tao, Yang Zhijun, et al. ASPC-YOLOv8 Detection Algorithm for Micro Traffic Signs [J]. Journal of Chongqing University of Technology (Natural Science), 2024, 38(05): 55-60.

[4] Du S, Pan W, Li N, et al.TSD-YOLO: Small traffic sign detection based on improved YOLO v8 [J]. IET Image Processing, 2024, 18(11): 2884-2898.

[5] Cui Y, Guo D, Yuan H, et al.Enhanced YOLO Network for Improving the Efficiency of Traffic Sign Detection [J]. Applied Sciences, 2024, 14(2): 555.

[6] Jing Xie, Yueming Deng, Runmin Wang. An Improved YOLOv8s Algorithm for Traffic Sign Detection [J]. Computer Engineering, 2024, 50(11): 338-349.

[7] Qiangjun Zhu, Bin Hu, Huilan Wang, et al. Traffic Sign Detection Based on Lightweight YOLOv8s [J]. Journal of Graphics, 2024, 45(03): 422-432.

[8] Liu Yuchen, Shi Gang, Cui Qing, et al. Improving MobileNetv3-YOLOv3 traffic sign detection algorithm [J]. Journal of Northeast Normal University (Natural Science Edition), 2022, 54(02): 53-60.

[9] Mou Jiayu, Nan Xinyuan. Real-time detection algorithm of small target of traffic sign based on YOLOv7-tiny [J]. Science Technology and Engineering, 2024, 24(30): 13072-13079.

[10] Woo, SanghyunCAa; Park, JongchanCAb; Lee, Joon-youngCAc; Kweon, InsoCAd. CBAM: Convolutional block attention module [J]. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018, 11211: 3-19.

[11] Quanhe Yao. Research on Natural Traffic Sign and Optical Remote Sensing Data Target Detection for Smart Transportation [D]. Xi'an: Xidian University, 2023.

[12] ZHU Z, LIANG D, ZHANG S H, et al. Traffic-sign detection and classification in the wild [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA: IEEE, 2016: 2110-2118.

[13] Varghese Rejin and M. Sambath. YOLOv8: A Novel Object Detection Algorithm with Enhanced [C]. Performance and Robustness, 2024 International Conference on ADICS, 2024, 1-6.

[14] Adarsh P, Rathi P., and Kumar M., YOLO v3-Tiny: Object Detection and Recognition using one stage improved model [C]. 2020 6th ICACCS, 2020, 687-694.

[15] Khanam R., and Hussain M. What is YOLOv5: A deep look into the internal features of the popular object detector [C]. ArXiv, 2024.

[16] Chuyin Li et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications [C]. ArXiv, 2022.

[17] Ge Z., Liu S., Wang F., Li Z., and Sun J. YOLOX: Exceeding YOLO Series in 2021 [C]. ArXiv, 2021.

[18] Khanam R., and Hussain M. YOLOv11: An Overview of the Key Architectural Enhancements [C]. ArXiv, 2024.