

Article

MSGC-YOLO: An Improved Lightweight Traffic Sign Detection Model under Snow Conditions

Baoxiang Chen  and Xinwei Fan *

College of Energy Environment and Safety Engineering & College of Carbon Metrology, China Jiliang University, Hangzhou 310018, China; chenbx@cjlu.edu.cn

* Correspondence: fxw@cjlu.edu.cn; Tel.: +86-135-8875-2279

Abstract: Traffic sign recognition plays a crucial role in enhancing the safety and efficiency of traffic systems. However, in snowy conditions, traffic signs are often obscured by particles, leading to a severe decrease in detection accuracy. To address this challenge, we propose an improved YOLOv8-based model for traffic sign recognition. Initially, we introduce a Multi-Scale Group Convolution (MSGC) module to replace the C2f module in the YOLOv8 backbone. Data indicate that MSGC enhances detection accuracy while maintaining model lightweightness. Subsequently, to improve the recognition ability for small targets, we introduce an enhanced small target detection layer, which enhances small target detection accuracy while reducing parameters. In addition, we replaced the original BCE loss with the improved EfficientSlide loss to improve the sample imbalance problem. Finally, we integrate Deformable Attention into the model to improve the detection efficiency and performance of complex targets. The resulting fused model, named MSGC-YOLOv8, is evaluated on an enhanced dataset of snow-covered traffic signs. Experimental results show that the MSGC-YOLOv8 model is used for snow road traffic sign recognition. Compared with the YOLOv8n model $mAP@0.5:0.95$, $mAP@0.5:0.95$ is increased by 17.7% and 18.1%, respectively, greatly improving the detection accuracy. Compared with the YOLOv8s model, while the parameters are reduced by 59.6%, $mAP@0.5$ only loses 1.5%. Considering all aspects of the data, our proposed model shows high detection efficiency and accuracy under snowy conditions.

Keywords: YOLOv8; traffic sign detection; small target detection; group convolution; data augmentation

MSC: 68T07



Citation: Chen, B.; Fan, X. MSGC-YOLO: An Improved Lightweight Traffic Sign Detection Model under Snow Conditions. *Mathematics* **2024**, *12*, 1539. <https://doi.org/10.3390/math12101539>

Academic Editor: Faheim Sufi

Received: 6 April 2024

Revised: 2 May 2024

Accepted: 14 May 2024

Published: 15 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid development of today's era, the wide application of artificial intelligence, and the continuous maturation of intelligent driving technology, the traditional driving mode is constantly being replaced. In this context, the safety of intelligent driving technology becomes particularly important. With traffic sign recognition technology, as an important part of the intelligent transportation system (ITS) [1], the accuracy of its recognition is directly related to whether the driving system is safe and reliable. Traffic sign recognition (TSR) [2] is mainly used to collect and recognize the traffic signs on the road (e.g., speed limit, no left turn, and high limit) during the process of automobile driving, and the main goal of TSR is to make the driving system able to automatically recognize the various traffic signs on the road, and to provide the driver with a safe, reliable, and reliable driving system through the use of computerized image processing and pattern recognition and other related technologies. The main goal of TSR is to enable the driving system to automatically recognize all kinds of traffic signs on the road by using computer image processing and pattern recognition technologies, so as to provide timely and accurate traffic information for the driver and the vehicle system. Nowadays, the rapid development of deep learning, image recognition and other fields provides strong support for traffic sign recognition technology. The research on traffic sign recognition technology can not only

improve the safety of road driving, but also help to realize the modernization of intelligent driving and traffic flow management. Therefore, the study of automatic recognition of traffic signs is of great significance for road safety, traffic management and scientific and technological development.

At present, the methods for traffic sign recognition can be mainly divided into two categories: traditional methods and methods based on deep learning.

The traditional traffic sign recognition methods are mainly based on color space analysis or shape analysis traffic signs [3,4], which use different color features and shape information among various types of traffic signs for recognition. After recognition, a predefined traffic sign template is matched with a local region in the image, or a classifier is used to classify the sign to detect and recognize the traffic sign. However, both of these methods can significantly reduce the detection effect as the shape of the traffic sign changes or the color of the traffic sign falls off.

The second is the deep learning-based traffic sign recognition method. The rapid development of deep learning in recent years has provided strong support for the research of traffic sign recognition technology. Deep learning detection algorithms are generally categorized into two-stage detection algorithms typically represented by R-CNN [5], Fast-RCNN [6] and Faster-RCNN [7], and one-stage detection algorithms represented by YOLO [8–14], SSD and Overfeat [15]. Due to the cumbersome model structure of two-stage detection algorithms such as R-CNN, which results in slower detection speed, it is difficult to meet the needs of actual traffic sign recognition. In contrast, one-stage detection algorithms have faster detection speed and can maintain good accuracy at the same time. In the detection research on traffic signs in recent years, a large number of scholars have adopted the YOLO series of algorithms to detect and identify traffic signs. And YOLOv8 [16], as the latest open-source detection algorithm in the YOLO series of algorithms, is characterized by fast speed and high accuracy. Therefore, this paper chooses the YOLOv8 algorithm to detect and recognize traffic signs.

However, the current traffic sign recognition task is mainly carried out for simple scenes. In the case of snowy weather, in order to solve the problem of low traffic sign recognition accuracy and difficult recognition, this paper proposes an improved YOLOv8 model, which mainly has the following improvements:

- (a) Multi-Scale Group Convolution is integrated into the original YOLOv8 model to make the model lightweight while improving detection accuracy.
- (b) Add an improved small target detection layer to enhance the model's detection accuracy for small targets.
- (c) The YOLOv8 model's BCE loss has been replaced with the EfficientSlide loss. This change smooths the model parameters and improves its stability and robustness.
- (d) Integrating Deformable Attention into the MSGC-YOLO model improves detection efficiency and performance, making it more efficient and accurate when processing complex visual tasks.

The improved model has significantly improved the accuracy of traffic sign recognition under snowy conditions. This method solves the problem of wrong detection and missed detection of traffic signs in snowy environment to a certain extent.

The work arrangement of the following chapters of this article is as follows: Section 2 introduces the relevant work background and research status at home and abroad, and proposes the superiority of the method in this article; Section 3 introduces the main improvement points and improvement methodology of the model in this article; Section 4 The effectiveness of this algorithm is verified through experiments; finally, Section 5 concludes that the algorithm has shown good performance in a variety of complex environments including snow, has high robustness and generalization, and has certain practical application value.

2. Related Work

Traffic sign detection is a crucial task in computer vision, with applications ranging from autonomous driving to intelligent transportation systems. Over the years, significant progress has been made in this field, driven by advancements in deep learning techniques and the availability of large-scale annotated datasets.

Early traffic sign detection systems often relied on handcrafted features and classical machine learning algorithms. For example, Houben et al. [17] proposed a method based on Histogram of Oriented Gradients (HOG) features for traffic sign detection.

In recent years, deep learning has revolutionized traffic sign detection, enabling end-to-end learning from raw pixel data. CNN-based architectures have become the cornerstone of modern traffic sign detection systems. For instance, Sermanet et al. [18] introduced a CNN-based approach for traffic sign detection and recognition, achieving state-of-the-art performance.

Variability in traffic sign appearance and adverse environmental conditions pose significant challenges for traffic sign detection systems. To address these challenges, researchers have proposed various solutions. For example, Dewi and Christine [19] presented a method for robust traffic sign detection using spatial pyramid pooling and multi-scale feature fusion.

Recent advances in deep learning have significantly improved TSR performance, but challenges remain, especially in dealing with variability and adverse conditions. Ongoing research efforts are needed to develop robust, real-time TSR systems that can operate effectively in diverse environments.

For example, in the above-mentioned literature, the HOG feature traffic sign method is quite sensitive to noise. In practical applications, after Block and Cell division, sometimes a Gaussian smoothing is performed to remove noise in each image area. And it itself does not have scale invariance, and its scale invariance is achieved by scaling the size of the detection window image; in order to pursue high classification accuracy, CNN has deepened the depth of the model and increased the complexity, resulting in high memory usage of the model and slow training speed. The power consumption and hardware performance of traffic sign recognition equipment are low, and the speed and accuracy requirements of the model are high, which makes the existing CNN model difficult to apply in practice. Robust traffic sign detection uses spatial pyramid pooling and multi-scale feature fusion. The method is mainly based on the YOLOv3 model, and today's YOLOv8 algorithm is lighter in size and has higher detection accuracy.

This article will use the latest YOLOv8 model for improvement. Most current research is on traffic sign recognition for simple weather such as clear or dark. In order to solve the problem of inaccurate traffic sign detection under snowy conditions, this paper proposes MSGC-YOLO, a model that is lighter, smaller and has higher detection accuracy than today's popular algorithms. MSGC-YOLO is effective in inspecting small target traffic signs and shows good performance.

3. Methodology

This article proposes the MSGC-YOLO fusion model. MSGC-YOLO fuses multi-scale grouped convolutions into the enhanced YOLOv8 backbone, adding additional detection layers for small objects and improving the classification loss function. It makes the model lightweight while improving detection accuracy. It integrates the Deformable Attention mechanism into the MSGC-YOLO model to effectively extract complex information and improve model detection performance. In this section, we will elaborate on the model framework, parameters, and specific implementation solutions.

3.1. YOLOv8n Model and Improvement Method

3.1.1. YOLOv8n Network

The YOLOv8 series model, as the latest YOLO series target detection model, is mainly aimed at tasks such as target detection, image classification and instance segmentation. Its

category includes a total of five models. According to the model size and training speed, it can be divided into YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. Based on the needs of actual tasks, this article selects the fastest and smallest YOLOv8n model as the benchmark model. The YOLOv8n model is shown in Figure 1.

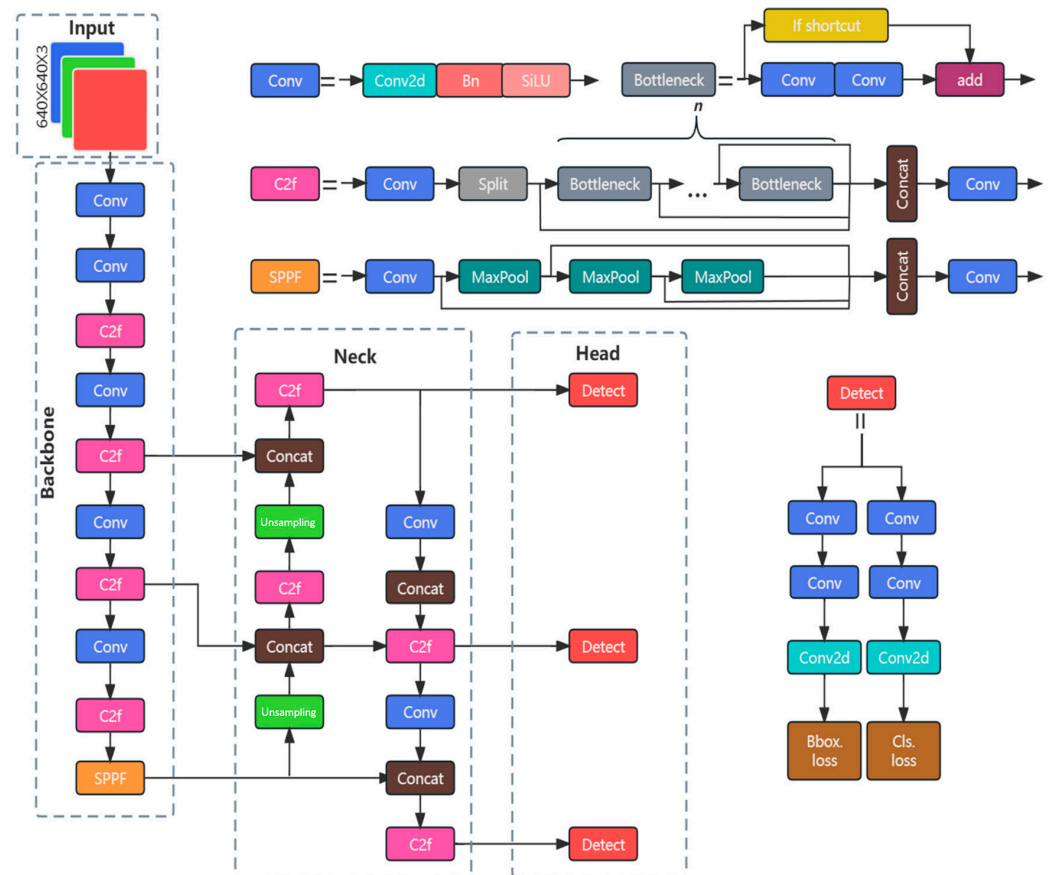


Figure 1. Overall architecture of YOLO.

The YOLOv8n model detection network is mainly divided into the following four parts: input, backbone, neck, and head. The input end uses mosaic data enhancement and introduces the idea of YOLOx to close the last 10 epochs to repair the accuracy loss problem after the early input size is aligned to 640×640 ; uses adaptive anchor box calculation and adaptive grayscale filling to reduce anchors; a number of box predictions to speed up non-maximum suppression NMS. The backbone network and neck part replace the block from C3 to C2F, and capture feature information of different scales through different convolution kernel sizes and strides. This enables the model to better adapt to targets of different shapes and sizes and improves the model's detection capabilities and accuracy. In terms of head, the classification and detection heads are separated from anchor-based to anchor-free to form a decoupled head structure. Loss calculation includes classification loss function and regression loss function. The classification loss uses the BCE loss function, and the regression loss consists of two parts, namely CIOU and DFL function.

3.1.2. Model Improvement Strategies

Aiming at the problem that the YOLOv8 network model has low detection accuracy in snow conditions and the number of model parameters is large, this article introduces a MSGC-YOLOv8 traffic sign detection model. The model structure is shown in Figure 2.

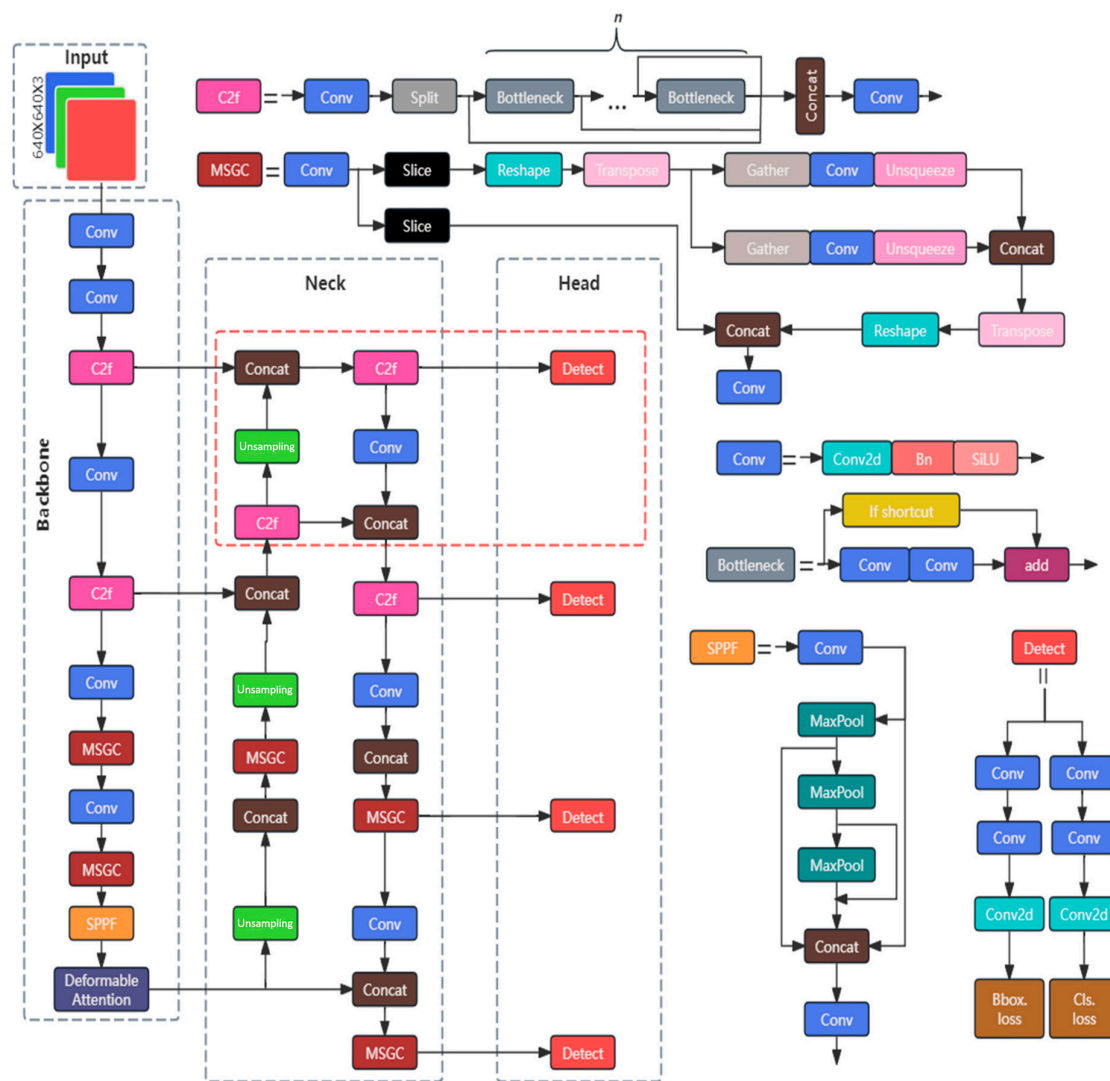


Figure 2. Overall architecture of MSGC-YOLO.

There are four main improvement parts. In the feature extraction backbone network and neck layer, part of the C2f module is replaced by the MSGC module, which reduces the number of parameters and slightly improves the accuracy. To address the problem of inaccurate small target detection, a small target detection layer is added. This makes it better for small target detection. Replace the classification loss function BCE loss with EfficientSlide loss to better solve the problem of sample imbalance. The Deformable Attention mechanism is added after the SPPF layer to greatly improve the detection accuracy of small target traffic signs.

3.2. Multi-Scale Group Convolution

In the YOLOv8 backbone, the MSGC replaces some C2f modules. MSGC is smaller than convolution and can obtain multi-scale features. Improve detection accuracy and training speed while reducing model size. The MSGC structure is shown in Figure 3.

Multi-Scale Grouping Conv is mainly implemented through the idea of group convolution [20]. Compared with Conv with the same number of channels, MSGC has a lower number of input and output parameters and calculations. Multi-scale information can be extracted while reducing the number of parameters and calculations. The Ghost module [21] found that in a set of feature maps output by a certain layer of visual Resnet-50, some channel contents are similar. This kind of information is called redundant information. For this type of information, the Ghost module suggests using simple linear operations

to generate ghosts. And the size of the feature map remains the same as the size of the original feature map. The redundant information is shown in Figure 4.

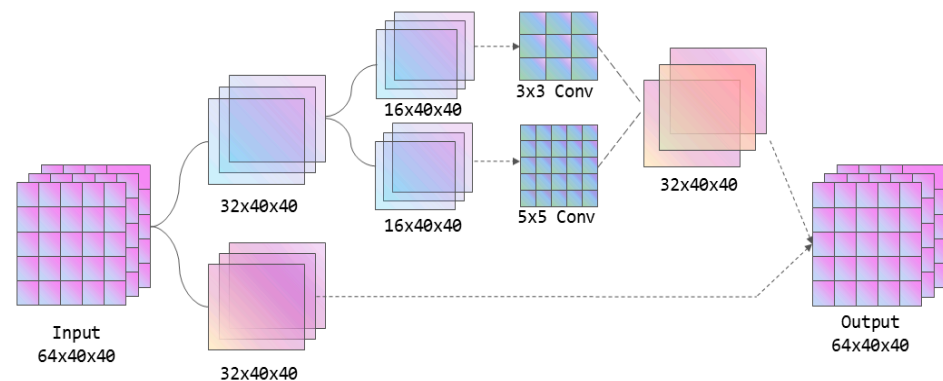


Figure 3. Multi-Scale Group Convolution.

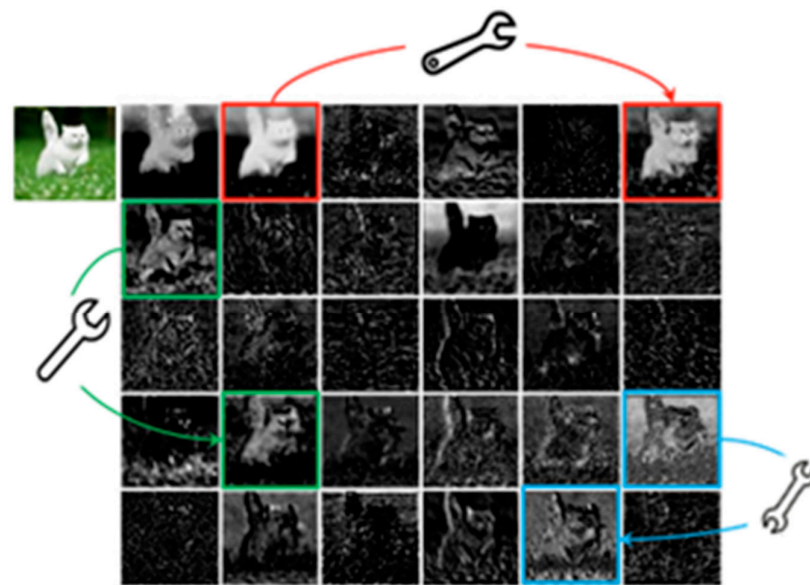


Figure 4. Redundant information.

Inspired by the idea of depth-wise separable convolution [22], we proposed the MSGC module in order to avoid too much redundant information. MSGC divides a Conv into two parts, one half is directly connected, and the other part is split again. And perform a 3×3 convolution on the two-quarters obtained and a 5×5 convolution on the other part. Splice the features obtained separately. However, since these features are completed in a single feature channel, the information between each feature channel is independent of each other. Therefore, this article adopts the method in MobileNetV2 [23], using Pointwise Conv [24] to increase or decrease the dimension to fuse the features of each channel.

Due to the smaller number of parameters required in the MSGC module than in the C2f module and the existence of multi-scale information, replacing part of the C2f module in the network with the MSGC module can reduce the number of model parameters and slightly improve the accuracy, making the model more lightweight.

3.3. Small Target Detection Layer

One of the main challenges when performing traffic sign recognition is that most targets are small-sized samples. Due to the small size of these small targets, and the relatively large downsampling multiple of YOLOv8, it is difficult for deep feature maps to effectively capture the key feature information of small targets.

In order to solve the above problems, this paper proposes a method of adding a small target detection layer. The scale of this layer is set to 160×160 , and a feature fusion neck layer and corresponding detection head are introduced to enhance the feature information extraction capability of small targets. First, the fifth 80×80 scale feature layer of the backbone network and the neck layer upsampling features are further stacked upward. Through the C2f module and upsampling processing, they are stacked with the third shallow position feature layer in the backbone network to obtain a 160×160 scale fusion feature containing small target feature information. Next, additional decoupling heads are added to the obtained feature information to extract target position and category information, respectively, learn through different network branches, and finally perform feature fusion. Improve the detection accuracy and range of smaller traffic signs and increase accuracy.

3.4. EfficientSlide Loss

The original YOLOv8n network model uses the BCE classification loss function, which is mainly suitable for binary classification tasks. And in the case of multi-label classification tasks. It often happens that the number base of easy samples is large and the number base of difficult samples is sparse.

To address this problem, YOLO-Face2 [25] introduces a sample weighting function (Slide) in detection. The difference between simple samples and difficult samples is based on the Intersection-over-Union (IoU) size of the prediction box and the ground truth box. However, samples near the boundaries tend to suffer larger losses due to unclear classification. In this case Slideloss tries to assign higher weights to difficult samples. First, the samples are divided into positive samples and negative samples by parameter μ . Then, the samples at the boundaries are emphasized through the weighting function Slide. Slideloss is shown in Figure 5.

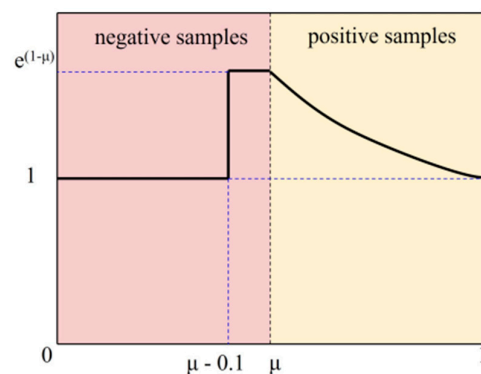


Figure 5. Slide loss.

Although this method can solve the problem of category imbalance to a certain extent, due to the lack of moving average thinking, when the model begins to converge in the later stages of model training, parameter values usually jump. A monotonic focusing mechanism for cross-entropy proposed in Focal loss [26] can effectively reduce the impact of simple examples on the loss value. This allows the model to focus on difficult examples and improve classification performance. For this reason, this article introduces the Exponential Moving Average (EMA) mechanism and proposes the EfficientSlide loss function to solve the problem of sample imbalance while weakening the impact of long-term data. By constructing the dynamic monotonic focus coefficient L_{Slide}^{γ} of Slide Loss, the coefficient introduces the idea of sliding average into Slide Loss, which can be used to smooth the model parameters and enhance the stability and robustness of the model. Its definition can be expressed as:

$$EMA : V_t = \beta \cdot V_{t-1} + (1 - \beta) \cdot \theta_t \quad (1)$$

V_t represents the average value of the previous t items ($V_0 = 0$), and β is the weighted weight value (generally set to 0.9–0.999).

$$\text{EfficientSlide loss} = \begin{cases} L_{\text{Slide}}^\gamma & x < \mu - 0.1 \\ L_{\text{Slide}}^\gamma e^{1-\mu} & \mu < x < \mu + 0.1 \\ L_{\text{Slide}}^\gamma e^{1-x} & x \geq \mu \end{cases} \quad (2)$$

We compared the improved EfficientSlide loss with the BCE loss and Slide loss of the original model. The experimental results are shown in Table 1. The data fully prove the effectiveness of the EfficientSlide loss improved in this article.

Table 1. Loss function performance comparison.

Methods	mAP@0.5/%	mAP@0.5:0.95/%
BCE loss	63.8	48.0
Slide loss	64.1	47.9
EfficientSlide loss	65.0	48.9

3.5. Deformable Attention Transformer

Deformable Attention Transformer (DAT) [27] contains a Deformable Attention mechanism that allows the model to dynamically adjust attention weights based on input content. The design of this model is inspired by the combination of Deformable Convolutional Networks (DCN) [28] and attention mechanisms to utilize deformation operations to dynamically adjust the shape and size of attention to better adapt to the structure of the input data.

The traditional Transformer [29] uses a standard self-attention mechanism, which processes all pixels in the image, resulting in a large amount of calculation. DAT introduces a Deformable Attention mechanism, which only focuses on a small number of key areas in the image. This approach can significantly reduce the computational effort while maintaining good performance. In the Deformable Attention mechanism, DAT dynamically selects sampling points instead of fixedly processing the entire image. This dynamic selection mechanism allows the model to focus more intensively on those areas that are most important to the current task. The design of DAT allows it to adapt to different image sizes and contents, making it work effectively in a variety of vision tasks, such as image classification and object detection. Deformable Attention is shown in Figure 6.

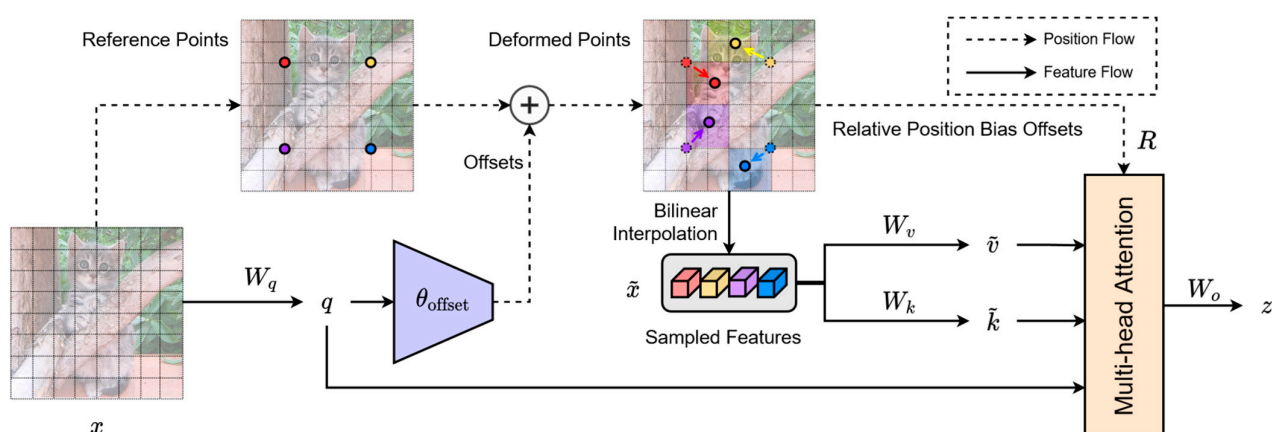


Figure 6. Deformable Attention Model.

The figure above shows the information flow of Deformable Attention. On the left part, a set of reference points are evenly placed on the feature map, and the offsets of these points are learned by the query through the offset network. Then, as shown on the right,

the deformed keys and values are projected from the sampled features based on the deformation points. The relative position deviation is also calculated through the deformation points, which enhances the multi-head attention for outputting transformed features. For clarity, only 4 reference points are shown in the figure, but in actual implementation, there are actually many more points.

Figure 7 shows the detailed structure of the offset generation network. The sizes of the input and output feature maps of each layer are marked (this Offset network needs to be controlled in the network code to add or not add it). DAT generates a variety of reference points distributed on the image through the above method, thereby improving the efficiency of detection.

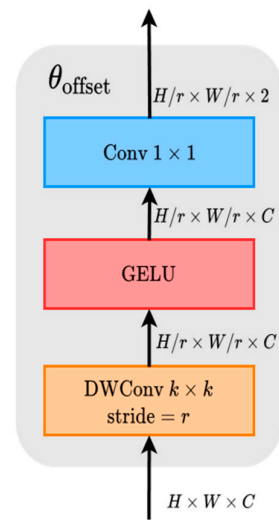


Figure 7. Offset network.

This article combines the Deformable Attention mechanism into the SPPF layer in the MSGC-YOLO model, adopts the dwc mode, and fixes the input size to 640×640 . Focus your limited attention on key information to save resources and obtain the most effective information quickly. This is used to enhance model detection efficiency and improve model detection accuracy.

4. Experiments

4.1. Dataset

The TT100K [30] dataset is a traffic sign dataset jointly produced by the joint laboratory of Tencent and Tsinghua University. However, due to uneven category instances in the original dataset, the dataset needs to be cleaned. This article only retains the instances with more than 100 categories. The remade dataset has a total of 43 categories and 8524 images. Since most of the existing traffic sign datasets are shot in simple environments, it is difficult to accurately identify traffic signs under harsh climate conditions. Therefore, in response to this situation, the tt100k dataset was data enhanced through python's third-party library *imgaug*. The original image was added with noise, changed saturation, brightness and other operations to simulate the situation where the camera is blocked by snowflakes in snowy conditions. We will divide the final dataset into the test set, training set and validation set in a ratio of 2:7:1. The enhanced effect is shown in Figure 8.



Figure 8. Data augmentation effect. (a) Original image; (b) enhanced image.

4.2. Experimental Environment

Our experiments were conducted using Python 3.8 and the PyTorch11.0 framework. The development platform is a 64-bit Linux system, and the processor is a 16 vCPU Intel(R) Xeon(R) Platinum 8352 V CPU @ 2.10 GHz. In order to improve training efficiency, NVIDIA GeForce RTX 4090 GPU with CUDA 11.3 and CuDNN 10.0 is used for graphics acceleration and accelerated through Baidu AutoDL cloud server resources. Furthermore, stochastic gradient descent is used to control loss reduction, the batch size is set to 16; close mosaics set to 10; workers set to 8, optimizer choose SGD and we use 150 training epochs.

4.3. Evaluation Criterion

We evaluate model performance and accuracy using standard precision (P), recall (R), average precision (AP), mean average precision (mAP), parameter count (Params), and speed (fps). Higher P and R values mean higher detection accuracy. mAP is an overall measure of model performance and reflects the effectiveness of training. Compared to P and R , mAP provides a more comprehensive evaluation of algorithm performance. In this experiment, we used $mAP@0.5$ and $mAP@0.5:0.95$ to comprehensively evaluate the model performance.

$$P = \frac{T_P}{T_P + F_P} = \frac{T_P}{all detections} \quad (3)$$

$$R = \frac{T_P}{T_P + F_N} = \frac{T_P}{all ground truths} \quad (4)$$

$$AP = \int_0^1 P_i(R_i) dR_i \quad (5)$$

$$mAP = \frac{1}{n} \sum_{j=1}^n AP_j \quad (6)$$

4.4. Ablation Experiment

In order to verify the effectiveness of the algorithm improvement module in this article, YOLOv8n is selected as the benchmark model. Use indicators such as precision, recall, $mAP@0.5$, $mAP@0.5:0.95$, Parameters and FPS for evaluation. Ablation experiments with different permutations and combinations of multiple modules were conducted. The experimental results are shown in Table 2.

Table 2. Ablation experiment.

MSGC Block	Small Layer	Efficient Slide	Deformable Attention	mAP@0.5/%	mAP@0.5:0.95/%	Parameters/M	FPS/(f.s ^{−1})
×	×	×	×	63.8	48.0	3.15	172.3
✓	×	×	×	64.5	48.5	2.86	173.4
×	✓	×	×	68.3	51.9	3.35	151.9
×	×	✓	×	65.0	48.9	3.15	172.3
×	×	×	✓	67.9	51.3	3.42	160.3
✓	✓	×	×	68.7	52.4	3.06	152.3
✓	×	✓	×	65.3	49.8	2.86	173.4
×	✓	×	✓	74.9	56.1	4.80	140.4
×	✓	✓	×	68.9	52.6	3.35	151.9
✓	✓	✓	×	69.2	52.8	3.06	152.3
✓	✓	✓	✓	75.1	56.7	4.51	142.2

As shown in Table 2, after adding the MSGC module, YOLOv8n can reduce the number of parameters and increase mAP0.5:0.95 by 0.7% and 0.5%, respectively. After adding the small target detection layer, although FPS are somewhat reduced, the accuracy can be greatly improved. mAP0.5:0.95 is increased by 4.5% and 3.9%, respectively; after improving the loss function, while maintaining the original model parameters and FPS unchanged, mAP0.5:0.95 were improved, respectively, 1.2% and 0.9%; finally add Deformable Attention mechanism, mAP was greatly improved, mAP0.5:0.95 increased by 4.1% and 3.3%, respectively. The final improved MSGC-YOLO, although the number of parameters and FPS has been lost, mAP0.5:0.95 has increased by 17.7% and 18.1% compared to the original model. Although some speed is sacrificed, a large accuracy gain is gained. Since the YOLOv8n original model has low detection accuracy, it is very important to improve the detection accuracy. This makes the detection of traffic signs in snow more likely to be applied to actual scenarios.

The visualization results of MSGC-YOLO and original YOLOv8n are shown in Figure 9. Although adding the small object detection layer and Deformable Attention slightly increases the number of model parameters, they significantly improve the recognition accuracy. In summary, combining these four improvements greatly improves detection accuracy, making traffic sign detection in snowy environments more effective.

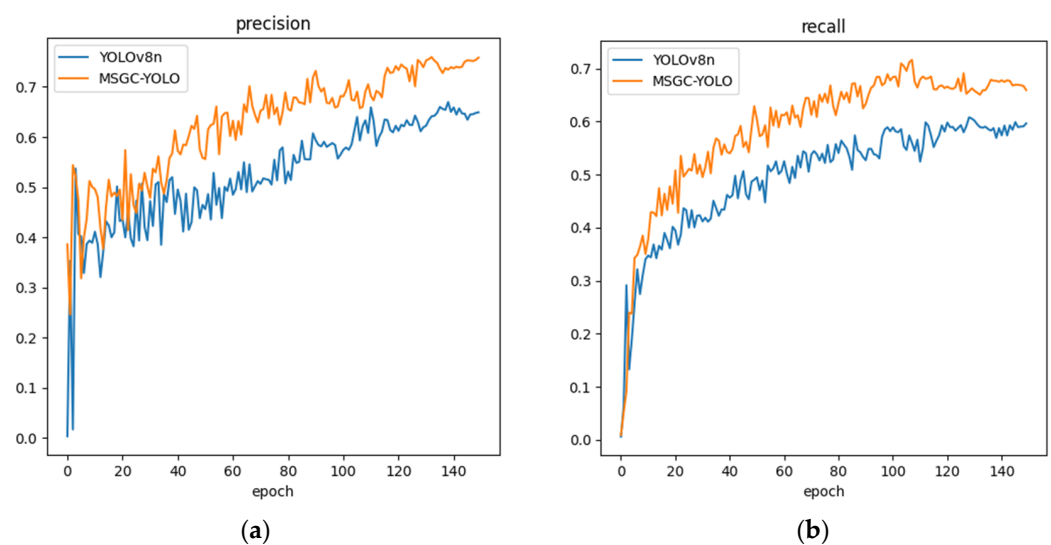


Figure 9. Cont.

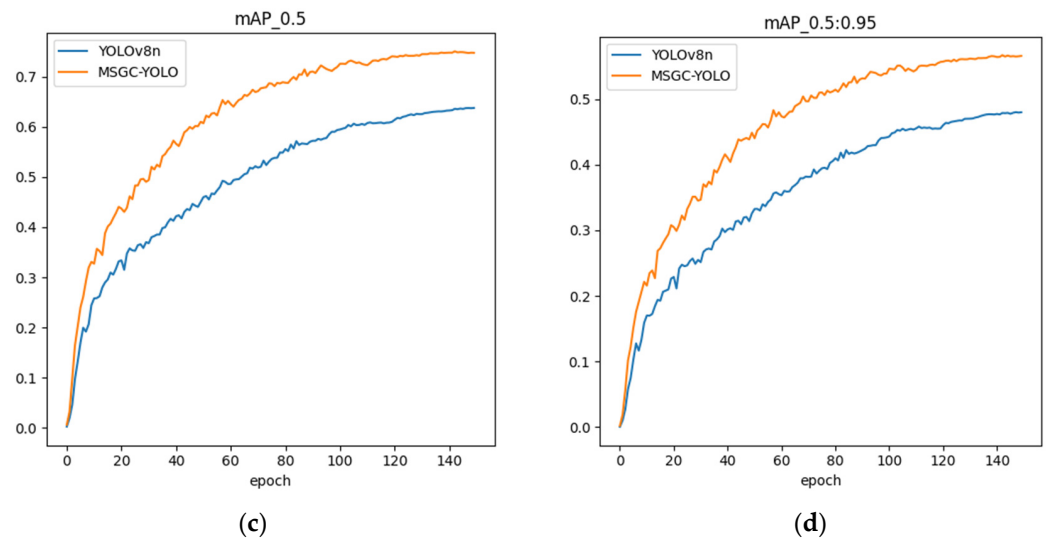


Figure 9. Visualize results. (a) Precision; (b) recall; (c) mAP0.5; (d) mAP0.95.

4.5. Comparison with Other Classic Algorithmst

In order to explore the superiority of the improved model MSGC-YOLOv8 in this article compared to the current popular traffic sign detection models, we compared it with the current popular traffic sign detection models. (i.e., YOLOv7-tiny, YOLOX_s, YOLOv8s, UniRepLkNet-YOLOv8, EfficientFormerV2-YOLOv8, Fasternet-YOLOv8). The results are listed in Table 3.

Table 3. Comparison of performance of different models.

Methods	mAP@0.5/%	mAP@0.5:0.95/%	Parameters/M	FPS/(f.s ^{−1})
YOLOv7-tiny	49.3	38.9	6.04	162.3
YOLOX_s	58.2	40.3	8.98	118.9
YOLOv8s	76.3	58.3	11.16	140.9
UniRepLkNet-YOLOv8	58.8	43.7	6.22	71.1
EfficientFormerV2-YOLOv8	65.7	49.7	5.25	98.0
Fasternet-YOLOv8	62.4	45.6	4.32	156.9
MSGC-YOLO	75.1	56.7	4.51	142.2

According to Table 3, among a series of YOLO algorithms, although YOLOv7-tiny has a higher FPS, the accuracy loss is serious. The mAP0.5:0.9 of MSGC-YOLO has increased by 52.3% and 45.7%, respectively. Compared with UniRepLkNet-YOLOv8 and EfficientFormerV2-YOLOv8, the model in this article has higher detection accuracy and FPS when the number of parameters is lower. Compared with Fasternet-YOLOv8, the model in this article achieves obvious mAP advantages with a small difference in the number of parameters. Finally, compared with the YOLOv8s model, this article only reduced mAP0.5:0.95 by 1.5% and 2.7% while reducing the number of parameters by 59.6%. The above data fully illustrate the superiority of the MSGC-YOLO detection algorithm.

4.6. Model Performance vs. Complexity Trade-Off

When analyzing the balance between model complexity and performance, we need to consider several factors. First, the complexity of the model includes aspects such as the number of parameters, computational complexity, and storage requirements. More complex models typically have more parameters and computational requirements, which can result in degraded performance or increased inference times in resource-constrained environments. Performance, on the other hand, covers aspects such as model accuracy, speed, and resource utilization. Ideally, we want the model to be as simple as possible while maintaining high performance so that it is more efficient in real-world deployments.

This article compares today's popular models with the MSGC-YOLO model, using FPS and mAP0.5 as evaluation indicators. The visualization results are shown in Figure 10.

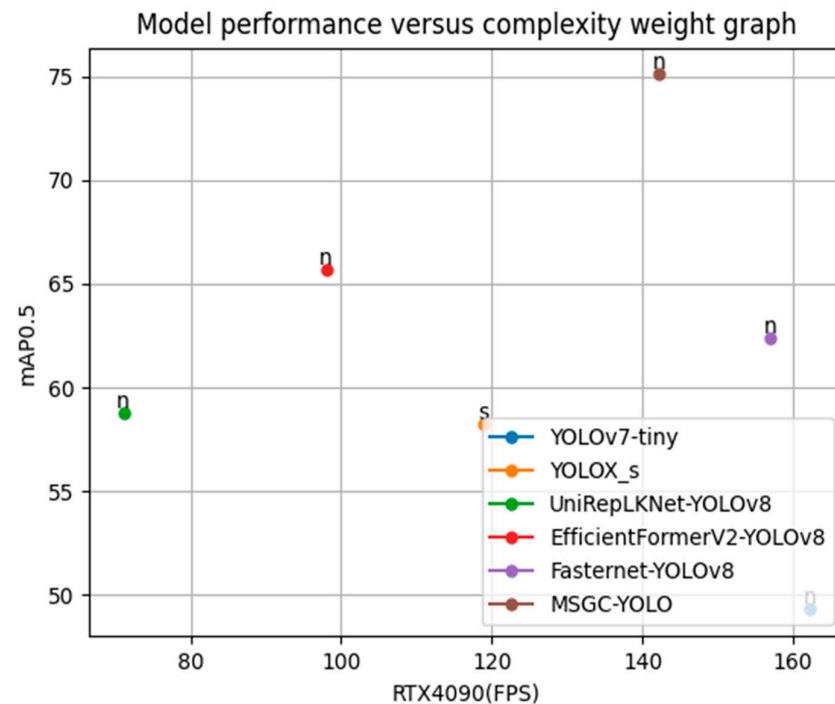


Figure 10. Model performance versus complexity weight graph.

As can be seen from Figure 10, the algorithm in this article can achieve a better balance between performance and complexity than other algorithms.

4.7. Detection Effect Comparison

Use YOLOv8n and our model to identify traffic signs in snowy conditions and compare the detection results with images. The results are shown in Figure 11. As can be seen from Figure I, the algorithm in this paper has higher detection accuracy when detecting a single target. As can be seen from Figure II, the original YOLOv8n incorrectly identifies p10 and p23, while MSGC-YOLO can correctly identify the flag information. As can be seen from Figure III, when there are multiple targets, the number of targets detected by MSGC-YOLO is relatively comprehensive, and it can detect targets even when the original model cannot detect them. As can be seen from Figure IV, MSGC-YOLO can also detect traffic sign information when facing small targets at long distances, but the original model cannot. The above data fully prove that the algorithm proposed in this article can improve the problems such as inaccurate expression of target features and difficulty in identifying small targets detected under severe weather interference factors.



Figure 11. Detection effect comparison. (a) MSGC-YOLO; (b) YOLOv8n.

4.8. Validation of Model Effectiveness in Other Environments

Due to the serious lack of traffic sign datasets in snowy weather, this paper adopts data enhancement method to simulate snowy weather for training. This method can enhance the robustness of the model and improve its performance in noisy environments. Image recognition is most inseparable from pixel recognition. In a real environment, no matter what kind of weather conditions, the most important factor that actually affects image recognition is the change in pixels. As a result, the detection information cannot be matched well. In this section, the trained snow traffic sign model will be used to detect traffic signs in a variety of different real weather environments to verify the generalization of the model. Since the selection of the original dataset may make it impossible to verify the validity of the model, this article selects the CCTSDB [31] dataset to conduct inference verification of the model under different weather environments. The results are shown in Figure 12.

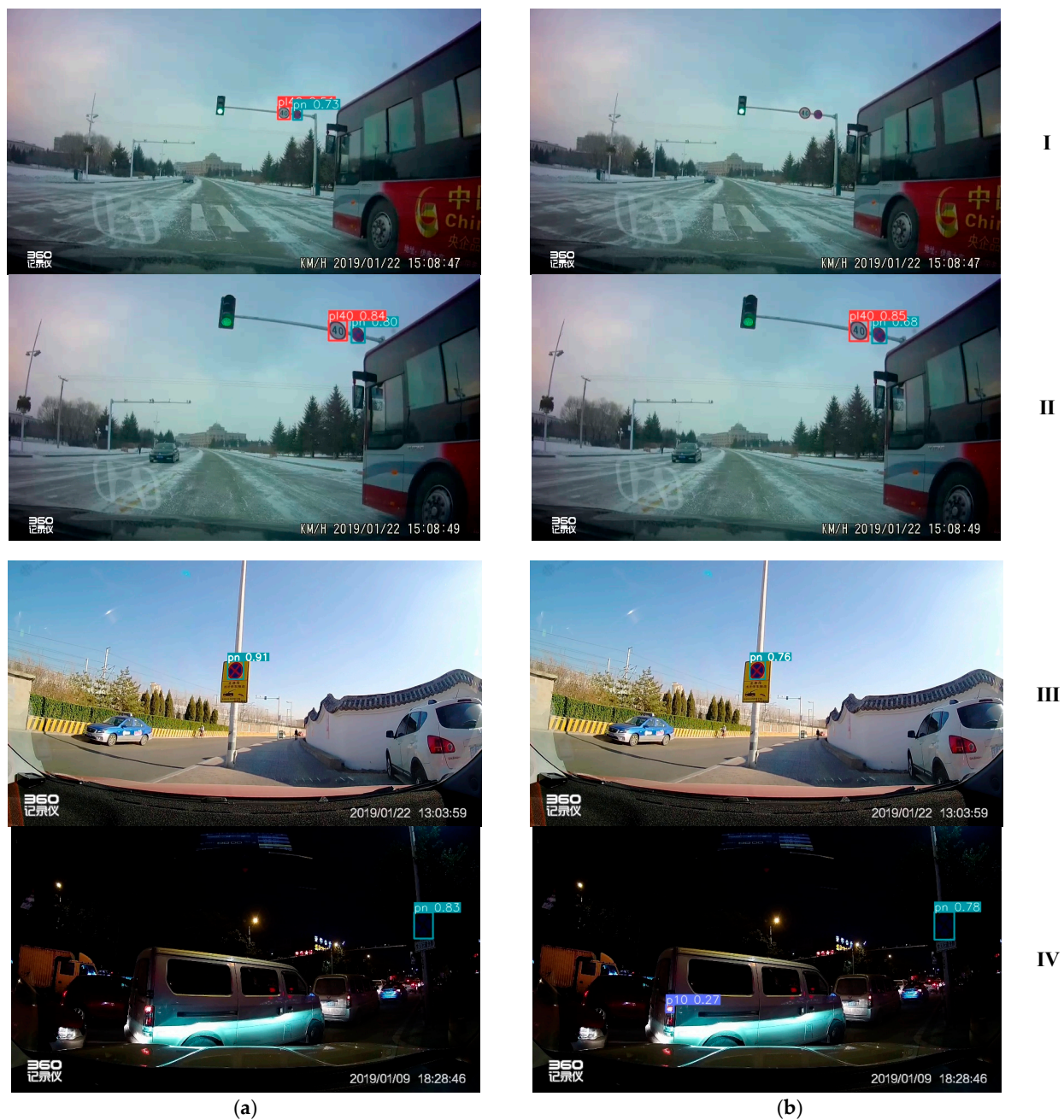


Figure 12. Comparison of detection effects in different environments. (a) MSGC-YOLO; (b) YOLOv8n.

According to Figure 12, it can be seen that in the snowy conditions of Figure I, the model of this article can detect the traffic signs very well, while the original model cannot detect the traffic signs at all; in Figure II, although the original model also detects the traffic signs logo. However, when faced with a variety of traffic signs and certain occlusions, the model in this paper has obviously higher detection accuracy; in Figure III, under clear weather conditions, MSGC-YOLO also achieves higher detection accuracy; finally, in Figure IV conditions at night. Under the circumstances, the original model mistakenly detected the lights of other cars as traffic signs, while the model in this paper not only has higher accuracy but also has no misdetection.

From the above data, it can be concluded that the detection effect of this model is not only improved in snowy days compared with the original model, it also has good generalization and robustness in the face of various complex weather environments and can provide better detection results.

4.9. Embedded System Deployment Feasibility

The MSGC-YOLO algorithm is used as a traffic sign recognition model and will be deployed on embedded devices in practical applications. Therefore, the authors discuss deployment feasibility in terms of model size and performance.

First, when deploying deep learning models on embedded devices, power consumption and heat issues need to be considered. More complex models may require more computing resources, resulting in increased power consumption or heating issues in the device. Therefore, when selecting models and deployment solutions, a balance between model complexity and device resources needs to be weighed to ensure the stability and reliability of embedded devices. Since the algorithm in this article has a lower number of parameters and a higher FPS than YOLOv5s, this article uses YOLOv5s as the benchmark model.

We chose NVIDIA's jetson series for discussion. Today's most popular embedded system offers scalable software, modern AI stacks, flexible microservices and APIs, production-ready ROS packages, and application-specific AI workflows at your fingertips. The new generation Orin series it launched has stronger performance, faster speed, and greater computing power. The new generation Orin model was benchmarked and the results are shown in Figure 13.

Model	Jetson Orin Nano 4GB	Jetson Orin Nano 8GB	Jetson Orin NX 8GB	Jetson Orin NX 16GB	Jetson AGX Orin 32GB	Jetson AGX Orin 64GB
Inception_V4	182	361	593	769	1337.8	1702.6
VGG19	174	361	442	532	937	1471
Super_resolution	102	203	280	386	610	882
UNET-segmentation	76	148	183	217	387	584
Pose Estimation	280	546	665	800	1424	2048
Yolov3-tiny	371	731	1156	1440	2611	3179
Resnet50	621	1158	1725	2183	3717	4834
SSD-Mobilenet	1094	2156	2893	3457	6415	7671
SSD_Resnet34_1200x1200	18	34	52	72	120	163
Yolov5m	69	131	162	193	342	519
Yolov5s	158	301	379	449	785	1135

Figure 13. Jetson Family Benchmarks.

As can be seen from Figure 13, in the new generation Orin series, even the lightest 4 GB nano model has an FPS of 158 when tested using YOLOv5s, while MSGC-YOLO has

a lower number of parameters and a higher accuracy than YOLOv5s. FPS. Therefore, the model in this article will perfectly adapt to the deployment of embedded devices and have better stability and reliability.

5. Conclusions

Aiming at the situation of inaccurate detection and recognition of traffic signs in heavy snow weather, incorrect detection and missed detection, etc., this article proposes a superior and lightweight MSGC-YOLOv8 network model. Introducing the newly designed MSGC module based on the group convolution idea, which slightly improves the accuracy while reducing the number of model parameters and model size, adds a small target detection layer to strengthen the feature extraction of small targets for the problem of difficult recognition of traffic signs for small targets in the distance; the ability to improve the detection accuracy of small targets; uses EfficientSlide loss based on the idea of sliding average to improve the problem of uneven number of difficult sample categories in the deep learning process. Adding an improved Deformable Attention mechanism enables the model to focus more on key areas in the image by adaptively adjusting the attention weight. Compared with the original network model, the MSGC-YOLOv8n network model has improved by 17.7% and 18.1% in mAP@0.5 and mAP@0.95, respectively; at the same time, compared with today's popular models, it has fewer parameters, faster FPS, and higher precision.

In the future, we will continue to improve the network model based on the model in this article, make the model lightweight, study traffic sign detection adapted to various environments, and transplant the model to embedded devices for verification to improve its practical application value.

Author Contributions: Writing—original draft preparation, B.C.; writing—review and editing, B.C. and X.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study are openly available in Tsinghua-Tencent 100K Annotations 2021 (with more classification) and CCTSDB.

Acknowledgments: We would like to thank the anonymous reviewers for their helpful remarks.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Purwar, S.; Chaudhry, R. A Comprehensive Study on Traffic Sign Detection in ITS. In Proceedings of the 2023 International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 11–12 May 2023; pp. 173–179.
2. Flores-Calero, M.; Astudillo, C.A.; Guevara, D.; Maza, J.; Lita, B.S.; Defaz, B.; Ante, J.S.; Zabala-Blanco, D.; Armingol Moreno, J.M. Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review. *Mathematics* **2024**, *12*, 297. [[CrossRef](#)]
3. Benallal, M.; Meunier, J. Real-Time Color Segmentation of Road Signs. In Proceedings of the CCECE 2003—Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology (Cat. No.03CH37436), Montreal, QC, Canada, 4–7 May 2003; IEEE: Piscataway, NJ, USA, 2003; Volume 3, pp. 1823–1826.
4. Yildiz, G.; Dizdaroglu, B. Traffic Sign Detection via Color And Shape-Based Approach. In Proceedings of the 2019 1st International Informatics and Software Engineering Conference (UBMYK), Ankara, Turkey, 6–7 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
5. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 580–587.
6. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
9. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.

10. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
11. Jocher, G. YOLOv5 by Ultralytics 2020. Available online: <https://github.com/ultralytics/yolov5> (accessed on 23 November 2023).
12. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
13. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
14. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
15. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection Using Convolutional Networks. *arXiv* **2013**. [CrossRef]
16. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLO 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 5 March 2024).
17. Houben, S.; Stallkamp, J.; Salmen, J.; Schlipsing, M.; Igel, C. Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark. In Proceedings of the The 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013; pp. 1–8.
18. Sermanet, P.; LeCun, Y. Traffic Sign Recognition with Multi-Scale Convolutional Networks. In Proceedings of the 2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; pp. 2809–2813.
19. 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. Humaniz. Comput.* **2023**, *14*, 8135–8152. [CrossRef]
20. Ioannou, Y.; Robertson, D.; Cipolla, R.; Criminisi, A. Deep Roots: Improving CNN Efficiency with Hierarchical Filter Groups. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5977–5986.
21. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features From Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1577–1586.
22. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
23. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 4510–4520.
24. Hua, B.-S.; Tran, M.-K.; Yeung, S.-K. Pointwise Convolutional Neural Networks. *arXiv* **2018**, arXiv:1712.05245.
25. Yu, Z.; Huang, H.; Chen, W.; Su, Y.; Liu, Y.; Wang, X. YOLO-FaceV2: A Scale and Occlusion Aware Face Detector. *arXiv* **2022**, arXiv:2208.02019.
26. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef] [PubMed]
27. Xia, Z.; Pan, X.; Song, S.; Li, L.E.; Huang, G. Vision Transformer with Deformable Attention. *arXiv* **2022**, arXiv:2201.00520.
28. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. *arXiv* **2017**, arXiv:1703.06211.
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
30. Zhu, Z.; Liang, D.; Zhang, S.; Huang, X.; Li, B.; Hu, S. Traffic-Sign Detection and Classification in the Wild. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2110–2118.
31. Zhang, J.; Zou, X.; Kuang, L.-D.; Wang, J.; Sherratt, R.S.; Yu, X. CCTSDB 2021: A More Comprehensive Traffic Sign Detection Benchmark. *Hum.-Centric Comput. Inf. Sci.* **2022**, *12*, 289–306. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.