Traffic Sign Detection Algorithm Based on Improved YOLOv8s

Xiaoming Zhang, Ying Tian

Abstract—**Aiming at the problems of low accuracy, false detection, missed detection, and low real-time detection of current traffic sign detection, this paper proposes an improved traffic sign detection algorithm based on the YOLOv8s algorithm. Firstly, this paper proposes a double-layer semi-composite backbone network structure (DSCB), which uses the auxiliary backbone network to extract features, and then transmits the extracted features to the backbone network to enhance the ability of the backbone network to extract target features. At the same time, the deformable convolution is integrated into the DC2f structure of the auxiliary backbone network to enhance the generalization performance of the network. Secondly, the coordinate attention mechanism is used after the SPPF layer. The coordinate attention mechanism can better retain the coordinate position information of small targets, reduce the miss rate of the model, and increase detection accuracy. Finally, this paper introduces a new CAB module to learn and aggregate the output of each layer of the feature pyramid for global spatial context to enhance the feature representation ability further. The experimental results show that the improved algorithm achieves 90.51% detection accuracy, 82.00% recall rate, 89.51% mAP@0.5 on the TT100K dataset, and the FPS reaches 106. Compared with the original algorithm model, the detection accuracy is increased by 2.27%, and the recall rate is increased by 2.48%. mAP@0.5 is increased by 2.01%, and FPS is increased by 1. The improved traffic sign detection algorithm meets the requirements in detection accuracy and real-time detection.**

Index Terms—**traffic sign detection, YOLOv8s, deformable convolution, coordinate attention mechanism, feature pyramid**

I. Introduction

In recent years, due to the rapid development of Intelligent Transportation Systems (ITS), traffic sign detection technology has been playing an increasingly vital role in intelligent traffic safety. With the continuous increase in traffic volume and the growing prominence of road traffic safety issues, traffic sign detection accuracy and real-time capability have become critically important. As a result, traffic sign detection technology has become one of the focal points of research in traffic safety [1-6].

The primary objective of traffic sign detection technology is to accurately identify traffic signs on road surfaces within complex road traffic environments. Traditional methods of traffic sign detection typically rely on image processing techniques such as template matching and feature extraction. However, these methods have numerous areas for improvement, such as sensitivity to lighting conditions, weather, and viewing angles. They also require significant manual intervention and parameter adjustments, resulting in poor robustness. Consequently, new algorithms for traffic sign detection continue to emerge, aiming to address these limitations.

As time has progressed, deep learning techniques have gradually found widespread applications in computer vision, effectively enhancing the accuracy and robustness of traffic sign detection. Among these techniques, object detection algorithms based on Convolutional Neural Networks (CNNs) [7], such as Faster R-CNN [8], SSD [9], FCOS [10], and YOLO [11], have emerged as mainstream approaches for traffic sign detection. In particular, the YOLO (You Only Look Once) algorithm has gained significant attention and application in recent years due to its high speed and accuracy characteristics. However, directly applying these methods to traffic sign detection often yields less satisfactory results in practical scenarios. This is because onboard mobile terminals exhibit low detection accuracy for objects of varying scales, and achieving real-time performance while meeting detection requirements remains challenging.

YOLOv8, an algorithm open-sourced by Ultralytics in January 2023, builds upon the successful foundation of previous YOLO series iteration. It introduces new features and improvements aimed at further enhancing performance and flexibility. Among its various models, YOLOv8s effectively balances detection precision and real-time capability. This advantage positions YOLOv8s favorably for tasks demanding both high accuracy and real-time performance, such as traffic sign detection. Therefore, this paper proposes an enhanced traffic sign detection algorithm based on the YOLOv8s architecture. As the latest iteration in the YOLO series, the YOLOv8 algorithm exhibits significant improvements in speed and accuracy compared to its predecessors. Despite having fewer parameters, it maintains enhanced precision, effectively reconciling the trade-off between real-time responsiveness and detection accuracy. Moreover, the algorithm boasts high versatility and scalability, accommodating many complex traffic environments and signage. The primary contributions of this paper are as follows:

First and foremost, this paper introduces a novel dual-layer semi-composite backbone network structure built upon the foundation of YOLOv8s backbone. The proposed backbone network structure comprises a primary backbone and an auxiliary backbone. Recognizing that including an

Manuscript received August 22,2023; revised November 16,2023.

This work was funded by the foundation of Liaoning Educational committee under the Grant No. LJKZ0310.

Xiaoming Zhang is a postgraduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China. (e-mail: 2468988768@qq.com).

Ying Tian is a Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China. (corresponding author to provide phone: +8613898015263; e-mail: astianying@126.com).

auxiliary backbone would inevitably increase the parameter count and potentially degrade real-time performance, we opt to utilize only half of the primary backbone's structure for the auxiliary backbone. Addressing the noteworthy contribution of the C2f module within YOLOv8s, this paper replaces standard convolutions with deformable convolutions within the module, enhancing its network generalization performance and facilitating the detection of occluded objects. For specific implementation details, please refer to Chapter Three.

Furthermore, this paper introduces a novel Context Aggregation Block (CAB). The central concept of this module is rooted in the idea that even after the aggregation of feature maps across different levels by the backbone network, the feature pyramid still retains spatial local information. The CAB module learns the global spatial context of each level's output in the feature pyramid to enhance features more effectively. This approach enables the fusion of local and global features in a manner that reduces information ambiguity, thereby achieving a balanced integration.

Continuing to address the issue of losing precise location coordinate information for small objects as the network deepens, this paper introduces a Coordinate Attention Mechanism. This attention mechanism encodes the precise positional information of targets through a sequence of convolutional operations along the image's height and width directions. This process yields feature maps imbued with width and height dimensions attention weights.

Finally, a comprehensive summary and analysis of the research findings will be presented, followed by exploring the practical application value and prospects of the optimized YOLOv8s algorithm in real-world traffic sign detection. The significance of this study lies in its optimization and enhancement of the YOLOv8s algorithm, leading to improved accuracy and robustness in traffic sign detection. This advancement contributes to the technical support required for intelligent transportation systems' secure and efficient operation. Moreover, the methods and insights gained from this research can also serve as a reference and inspiration for addressing object detection challenges in other domains. The remaining structure of this paper is outlined as follows: Section Two introduces traditional traffic sign detection algorithms, and deep learning-based traffic sign detection algorithms and provides an overview of the YOLOv8s algorithm. Section Three elaborates on the proposed methodology, detailing the development of a real-time and efficient traffic sign detection approach. Experimental results and analysis are presented in Section Four. Lastly, Section Five offers concluding remarks.

II. Related Work

As a significant branch of object detection, traffic sign detection has emerged as a research hotspot in computer vision in recent years. Research in traffic sign detection can be divided into traditional and deep learning-based methods. Traditional techniques for traffic sign detection can be categorized into color-based, shape-based, and machine-learning-based methods. Color and shape-based detection techniques manually extract features from images based on specific color and shape characteristics. Algorithms such as Histograms of Oriented Gradient (HOG) and Scale Invariant Feature Transform (SIFT) are utilized to crop and extract traffic signs from images, detecting them through template matching mechanisms. Reference [12] proposed a traffic sign detection approach based on HOG features and Support Vector Machines (SVM). This method initially segments traffic sign images using color thresholds to eliminate substantial interference. Then, the Max Stable Extremal Region algorithm is employed to detect connected regions. Reference [13] introduced a color-based segmentation model for traffic sign detection. It transforms RGB color ranges into HIS and detects red, yellow, blue, and green colors for traffic sign identification. The Region of Interest (ROI) with extracted features using Histograms of Oriented Gradient (HOG) or Pyramid HOG (PHOG) is subsequently classified using Support Vector Machines, ultimately leading to traffic sign detection. However, color and shape-based methods are susceptible to limitations, particularly the influence of lighting, weather conditions, and other environmental factors.

With the advancement of deep learning, object detection algorithms have matured over time, categorizing mainstream object detection methods into two main approaches: one-stage and two-stage detection. Two-stage object detection algorithms follow a process where the network first proposes regions of interest, followed by object detection carried out by classification and localization networks. While these algorithms tend to achieve higher detection accuracy, their real-time performance is often compromised. Representative algorithms in this category include SPP-Net, Faster R-CNN, and R-FCN. In contrast, one-stage object detection algorithms accomplish end-to-end detection without the need to pre-extract regions of interest. These algorithms directly yield object classification probabilities and bounding box coordinates during detection. While typically offering lower detection accuracy than two-stage methods, one-stage algorithms excel in real-time performance. Noteworthy examples within this category include SSD (Single Shot Multibox Detector), YOLOv3, YOLOv5, and YOLOv7 [14].

In recent years, numerous researchers have delved into deep learning-based traffic sign detection. Reference [15] introduces an improved feature pyramid model called AF-FPN, which employs an Adaptive Attention Module (AAM) and Feature Enhancement Module (FEM) to mitigate information loss during feature map generation, thereby enhancing the expressiveness of the feature pyramid. Reference [16], addressing low detection accuracy and data collection issues in traffic sign detection, proposes an enhanced Sparse R-CNN by integrating the Coordinate Attention Mechanism and ResNeSt, constructing a feature pyramid to rectify the backbone and improve detection precision. Reference [17] tackles the lack of high-level spatial information for minor traffic signs. It introduces the Parallel Deformable Convolution Module (PDCM), maintaining the integrity of abstract information through symmetric branches to enhance feature extraction capabilities. These improvements aim to enhance detection accuracy while maintaining real-time performance. However, challenges such as missed detections, false

positives, and further enhancement of multi-scale feature fusion persist, particularly for small-sized traffic signs. Addressing the earlier challenges in traffic sign detection, this paper opts to study the latest model in the YOLO series, YOLOv8, as the foundational algorithm. YOLOv8 offers five versions: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x, categorized based on model size. Among these, YOLOv8s is a lightweight model, but there is room for further improvement in detection accuracy. Hence, this paper focuses on enhancing the YOLOv8s version to boost detection precision.

Like YOLOv5, YOLOv8 comprises three main components: the backbone network, the neck network, and the detection head. The backbone network continues to employ the Conceptually Simple and Practical (CSP) architecture. However, unlike YOLOv5, YOLOv8 replaces the C3 module found in YOLOv5 with the C2f module, drawing inspiration from the ELAN concept in YOLOv7. This adaptation enables YOLOv8 to achieve richer gradient flow information while maintaining a lightweight design.

YOLOv8 utilizes a PAN-FPN (Path et al. with Feature Pyramid Network) structure for the neck network. The backbone network performs downsampling operations, followed by two cross-layer fusion connections between the upsampled and downsampled branches. This enhances the fusion and utilization of feature information across various scales.

Notably, YOLOv8 introduces significant improvements in the detection head. It employs a decoupled head structure, separating the classification and detection heads while discarding the objectness branch. Only the classification and regression branches are retained, representing a departure from previous designs. By integrating these modifications, YOLOv8 aims to achieve enhanced detection performance while maintaining lightweight characteristics, offering advancements over its predecessors.

Regarding the sample matching strategy, YOLOv8 departs from the Anchor-Based approach and instead adopts an Anchor-Free methodology. This shift is motivated by the fact that traditional Anchor-Based methods can lead to significant computational overhead during training and require extensive manual tuning of hyperparameters. In contrast, the Anchor-Free approach simplifies the determination of positive and negative samples, achieving and surpassing the accuracy of Anchor-Based methods while offering faster processing speeds. YOLOv8 employs the Task-Optimal Odd Sampling (TOOD) strategy for allocating positive and negative samples. This approach aligns the task and distribution of positive samples based on weighted scores for classification and regression, as illustrated by the following equation:

$$t = s^{\alpha} \times u^{\beta} \tag{1}$$

In the YOLOv8 algorithm, s represents the model's classification score, and u signifies the Intersection over Union (IOU) between the predicted box and the ground truth box. The YOLOv8 algorithm incorporates both a classification branch and a regression branch. In the context of the classification branch, the Binary Cross-Entropy (BCE) loss is employed, and it is formulated as follows:

$$Loss_n = -w\left[y_n \log x_n + (1 - y_n) \log (1 - x_n) \right] \tag{2}$$

Where w represents weights, y_n denotes ground truth values, and x_n signifies algorithm-predicted values, the regression branch employs the Distribution Focal Loss (DFL) function and the Complete Intersection over Union (CIOU) loss function. The primary objective of the DFL loss function is to model the positional distribution of bounding boxes as a Gaussian distribution. This modeling facilitates the network's rapid attention toward positions closely aligned with the target location. The formulation of the DFL loss function is presented as follows:

$$\begin{aligned} DFL(S_n, S_{n+1}) = -((y_{n+1} - y)\log(S_n) \\ + (y - y_n)\log(S_{n+1})) \end{aligned} \tag{3}$$

$$S_n = \frac{y_{n+1} - y}{y_{n+1} - y_n}, S_{n+1} = \frac{y - y_n}{y_{n+1} - y_n} \tag{4}$$

The CIOU (Complete Intersection over Union) introduces an aspect ratio term to the DIOU (Distance-IoU) metric, with its specific definition as follows:

$$CIOU_{Loss} = 1 - IOU + \frac{d^2}{c^2} + \frac{v^2}{(1 - IOU) + v} \tag{5}$$

Where d represents the distance between the centers of the predicted and ground truth bounding boxes, and c corresponds to the diagonal distance of the minimum enclosing rectangle. v stands for the similarity factor based on aspect ratio, and its definition is provided below:

$$v = \frac{4}{\pi^2}(\arctan\frac{W_b}{H_b} - \arctan\frac{W_p}{H_P})^2 \tag{6}$$

Where W_b is the width of the true box, H_b is the height of the true box, W_p is the width of the predicted box, and H_p is the height of the predicted box.

III. Improved Model

In Chapter 3, this paper presents a detailed exposition of the modules introduced. Section 3.1 outlines the overall structure of the proposed model. Section 3.2 delves into the newly introduced dual-layer semi-composite backbone network structure and its variants. Subsequently, Section 3.3 elaborates on the Context Aggregation Block (CAB) introduced in this study. Lastly, Section 3.4 provides insight into the introduced Coordinate Attention Mechanism.

A. Improved YOLOv8s Algorithm Model

This section presents the enhanced structure of the YOLOv8s algorithm proposed in this paper, as depicted in Fig. 1. Firstly, an improvement is made to the backbone network of YOLOv8s by introducing a dual-layer semi-composite backbone network structure. This structure comprises a leading backbone network and an auxiliary backbone network. Both components perform feature extraction on the input image, with the auxiliary backbone network transmitting the extracted features to the leading backbone network for subsequent detection tasks. Furthermore, a Coordinate Attention mechanism (CA) is

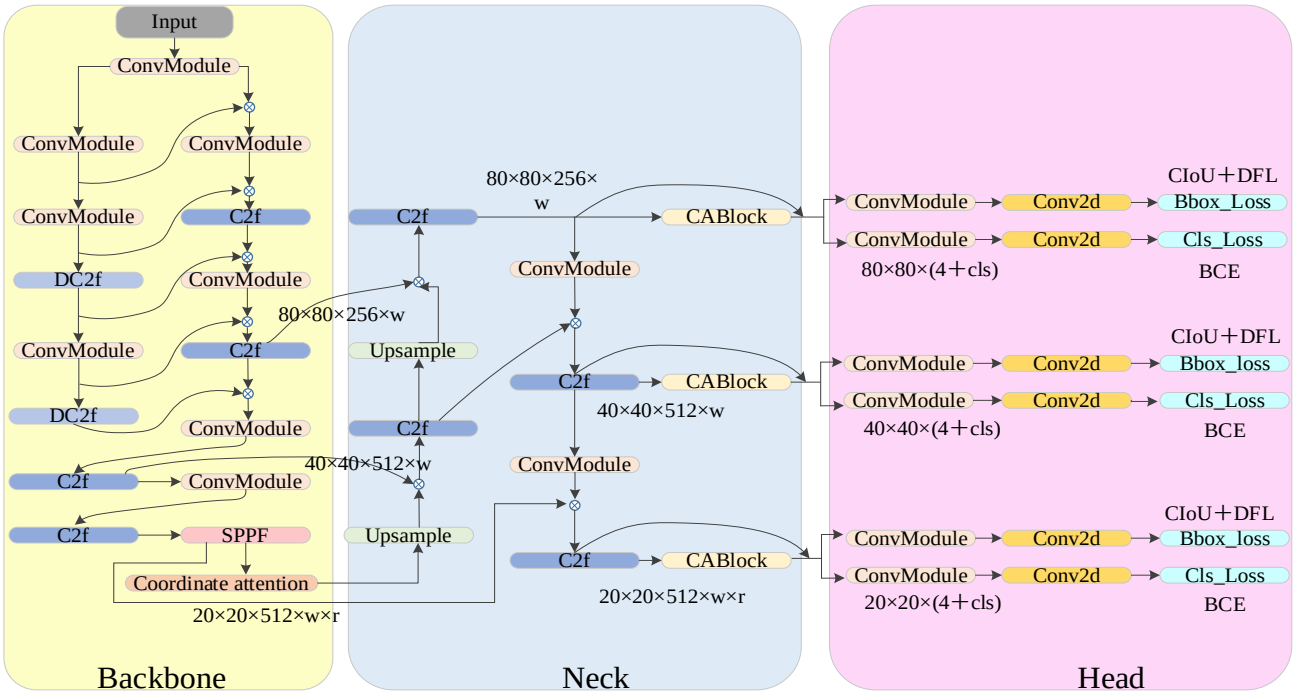


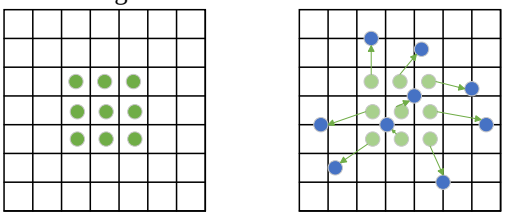
Fig. 1. The improved YOLOv8s structure diagram

introduced after the Spatial Pyramid Pooling Fusion (SPPF) layer. This is motivated by the ability of the Coordinate Attention mechanism to mitigate the issue of losing positional information for small objects in object detection. Moreover, the Coordinate Attention mechanism is designed as a lightweight attention mechanism, minimizing the additional computational burden on the network model. Lastly, a Context Aggregation Block (CAB) is introduced before each detection head. This module enhances features by learning the global spatial context from the outputs of each layer in the feature pyramid. This strategy effectively fuses local and global features while reducing information confusion.

B. Dual-Layer Semi-Composite Backbone Network Structure

The strength of convolutional neural networks (CNNs) in feature extraction is closely intertwined with the structure of the backbone network. Therefore, a robust backbone network often yields promising results for various research tasks. As long as the backbone network extracts sufficient features and the loss of information remains within an acceptable range, the network's performance can be significantly enhanced. YOLOv8s utilizes the CSPDarkNet structure as its backbone network. This paper proposes a dual-layer semi-composite backbone network structure based on the CSPDarkNet architecture. The specific implementation details are outlined as follows.

The Dual-Layer Semi-Composite Backbone (DSCB) network consists of a leading backbone network structure and an auxiliary backbone network. The auxiliary backbone network is composed of three convolutional layers and two DC2f modules. The difference between DC2f and C2f lies in introducing deformable convolutions in the C2f module. Deformable convolutions introduce position offsets within the receptive field, which are learnable. This enables the extracted features during the convolution process to better align with the actual shape of objects, regardless of their deformation. The convolutional region consistently covers the surroundings of the object. The deformable convolution is illustrated in Fig. 2.



(a) (b)

Fig. 2. Schematic of deformable convolutions

For standard convolution, the computation can be generally expressed as follows: sample a set of pixels from the input feature map $R=\{(-1,-1),(-1,0)...,(1,1)\}$, using convolution to calculate the sampling results and obtain the results after convolution. As shown in Equation 7. Where: p_n denotes the position in R and w denotes the convolution weight.

$$y(p_0)=\sum_{p_n\in R} w(p_n)\bullet x(p_0+p_n) \tag{7}$$

For deformable convolution, it does not directly change the shape of the convolution kernel, but modifies the sampling result, to indirectly achieve the effect of changing the shape of the convolution kernel. In deformable convolution, Δp_n is used to augment a point p_n on the feature map, where $\{\Delta p_n \mid n=1,2,...,N\}$. The deformable convolution is calculated as follows.

$$y(p_0)=\sum_{p_n\in R} w(p_n)\bullet x(p_0+p_n+\Delta p_n) \tag{8}$$

The auxiliary backbone network first extracts features using two convolutional modules. The extracted features are then fed into the DC2f module, where another convolutional operation occurs. Afterward, the extracted features are passed through the final DC2f module, completing the entire

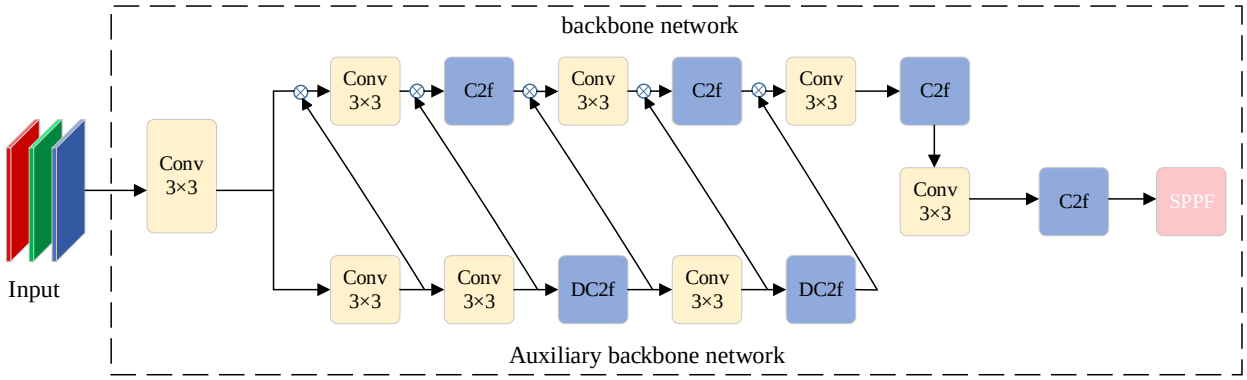


Fig. 3. First Feature Fusion Structure in Double-Layer Semi-Composite Backbone Network (DSCB1)

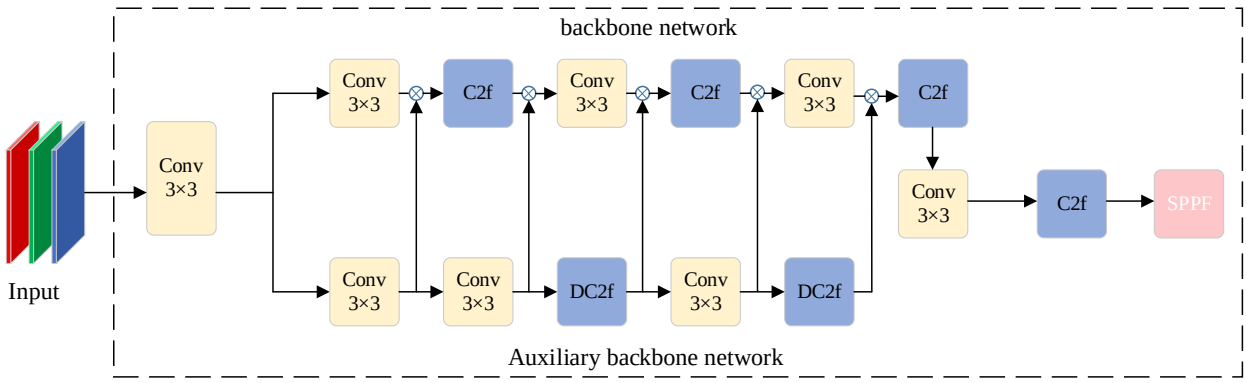


Fig. 4. Second Feature Fusion Structure in Double-layer Semi-composite Backbone Network (DSCB2)

operation of the auxiliary backbone network. Structural differences exist between the auxiliary backbone network and the leading backbone network. Since adding an auxiliary backbone network would increase the parameter count and computational complexity, the auxiliary backbone network is designed to have only half the structure of the leading backbone network. This design ensures no significant increase in parameters or computational complexity. The proposed dual-layer semi-composite backbone network structure is illustrated in Fig. 3.

Once the auxiliary backbone network structure is designed, this paper proposes two methods for feature fusion with the leading backbone network. The first method involves feature fusion among corresponding layers through high-level connections, while the second method employs feature fusion among corresponding layers through peer-level connections. Among these, the first feature fusion method is illustrated in Fig. 3. In this method, the output of the first convolutional layer of the auxiliary backbone network is passed to the leading backbone network before its first convolutional layer. The main backbone network incorporates the features extracted by the auxiliary backbone network from the first layer before its initial convolution, thus obtaining features of varying scales during the feature extraction process. Like the preceding step, the auxiliary backbone network continues its subsequent feature extraction tasks and submits the features it extracts to the leading backbone network at the same layer before the corresponding feature extraction operation takes place. This procedure enhances the feature extraction capability of the leading backbone network. The second feature fusion method involves peer-level connections among corresponding layers, as depicted in Fig. 4. The outputs of each layer from both the auxiliary backbone network and the leading backbone network are aligned. Within the same layers, the features extracted by the auxiliary backbone network are fused with those of the leading backbone network. The input of the subsequent layer of the leading backbone network will carry features extracted by the auxiliary backbone network and those obtained by the upper-level main backbone network. The outputs of each subsequent layer of the auxiliary backbone network will be fused with the aligned outputs of the main backbone network, thereby augmenting the overall feature extraction capability of the entire network.

C. Context Aggregation Block

The leading backbone network is responsible for extracting crucial features from input images, while the feature pyramid structure of the neck network aggregates feature mappings of varying levels. However, the feature pyramid structure still incorporates certain spatial local information. Therefore, this paper introduces contextual aggregation blocks to merge global contextual information for each layer, further enhancing the capability of representing features.

Following the neck network, this paper introduces contextual aggregation blocks, with each layer employing a residual structure. The detailed design of this module is illustrated in Fig. 5. Within each block, per-pixel spatial context is aggregated using Equation 9. Here, P_i and Q_i represent the input and output feature mappings of the i-th layer in the feature pyramid, each containing N_i pixels. Indices $j, m \in \{1, N_i\}$ denote the pixel positions, while w_k and w_v stand for linear transformation matrices used for

projecting the feature mappings. In practice, a 1x1 matrix is employed for feature mapping operations.

$$Q_i^j = P_i^j + a_i^j \bullet \sum_{j=1}^{N_i} \left[\frac{\exp(w_k P_i^j)}{\sum_{m=1}^{N_i} \exp(w_k P_i^m)} \bullet w_v P_i^j \right] \quad (9)$$

As depicted in Fig. 5, the feature pyramid structure within the neck network aggregates features from different levels of the input image. After this aggregation process, three branches are generated to feed into the detection head for the final objective of target detection.

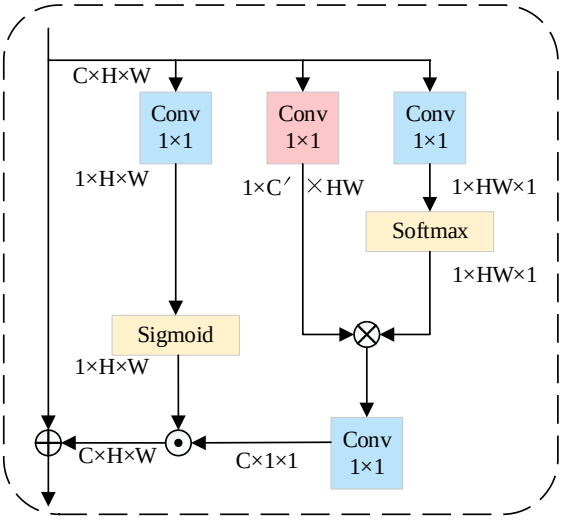


Fig. 5. context aggregation block

Due to each branch carrying a certain amount of spatial contextual information, we introduce contextual aggregation blocks after each branch. For each branch, this paper incorporates residual connections to ensure the integrity of feature information. The feature maps produced by each branch initially undergo a 1x1 convolutional feature mapping operation within the contextual aggregation block. Following the 1x1 convolution, a Sigmoid operation is applied for the first branch. The second branch begins with a 1x1 convolutional feature mapping operation and then fuses its output with the third branch. The third branch executes a 1x1 convolutional feature mapping operation and subsequently feeds into the Softmax function. This output is merged with the output of the second branch. After the feature fusion of the two branches, the output is passed through another 1x1 convolutional block. This output is then element-wise multiplied (Hadamard product) with the output of the first branch. The result of the Hadamard product is subjected to a residual connection operation with the initial input. This sequence outlines the execution process of the contextual aggregation block described in the subsequent sections.

D.Coordinate Attention Mechanism

The attention mechanism originates from the phenomenon of attention in the human visual system, wherein humans automatically focus on a specific area while ignoring others when observing objects. Currently popular attention mechanisms include the Channel Attention Mechanism (SE), the CBAM Attention Mechanism which combines channel attention with spatial attention, and the Spatial Attention mechanism (SA) that involves channel shuffling. The integration of these attention mechanisms, to varying degrees, into object detection models often leads to improved network performance. However, when it comes to small objects, these mechanisms tend to overlook positional information. The Coordinate Attention Mechanism (CA) is an improved version built upon the Channel Attention Mechanism (SE). It incorporates positional information to capture spatial structure, thus making it a lightweight attention approach with lower module complexity compared to SE and CBAM. By integrating positional information with channel information, it enhances the feature representation of mobile networks. The structure of the Coordinate Attention Mechanism is depicted in Fig. 6.

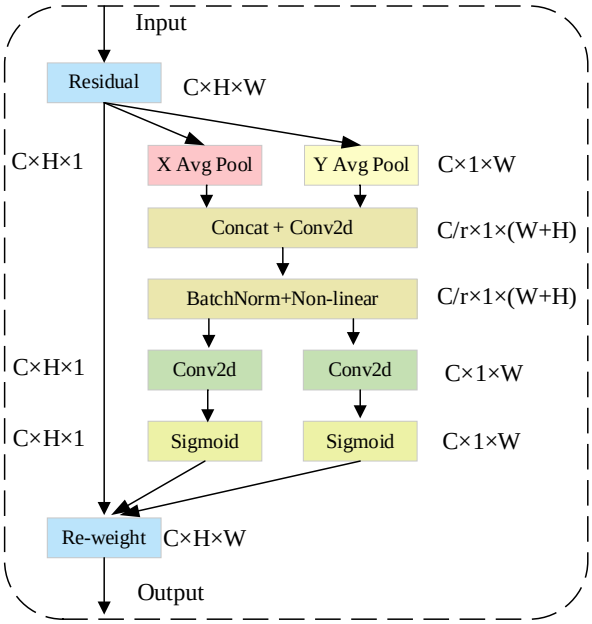


Fig. 6. coordinate attention structure map

The Coordinate Attention Mechanism achieves precise encoding of positional information for channel relationships and long-range dependencies through Coordinate Information Embedding and Coordinate Attention Generation. During the Coordinate Information Embedding process, global pooling methods are typically employed to encode global spatial information for channel attention. However, such methods often compress global spatial information into channel descriptions, making it challenging to preserve the positional information of targets. To address this, the Coordinate Information Embedding process decomposes the global pooling, as shown in Equation 10. Here, Z_c represents the output associated with the c-th channel, and the input x comes directly from a convolutional layer with a fixed kernel size. This decomposition helps retain positional information, allowing the Coordinate Attention Mechanism to encode channel relationships and long-range dependencies more accurately.

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j) \quad (10)$$

The coordinate information embedding process specifically, given an input x, each channel is first encoded along the horizontal and vertical coordinate directions, respectively, using a pooling kernel of size $(h,1)$ or $(1,w)$.

Therefore, the output of channel c with height h can be expressed as follows:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i) \tag{11}$$

Similarly, the output of the c channel of width w can be expressed as follows:

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le j < H} x_c(j, w) \tag{12}$$

The transformations aggregate feature information along two spatial directions, producing directionally aware feature maps. These transformations enable the attention module to capture long-range dependencies along one spatial direction while preserving precise positional information along the other spatial direction. This capability aids the network in accurately localizing the regions of interest within the input, contributing to improved object localization.

Continuing with the Coordinate Attention Generation process, the next step involves concatenating the feature maps from the two directions, corresponding to the width and height of the global receptive field. These concatenated feature maps are then passed through a shared convolutional module with a 1x1 kernel. The dimensionality of the feature maps is reduced to the original value C / r . Subsequently, the batch-normalized feature map F_1 is fed into the Sigmoid activation function to obtain the feature map f in the form of $1 \times (W + H) \times C / r$, as shown in the following equation:

$$f = \delta(F_1([z^h, z^w])) \tag{13}$$

Continuing, the feature map f is further processed by performing 1x1 convolution separately along the original height and width dimensions. This results in two feature maps, F_h and F_w , both having the same number of channels as the original feature map. After passing through the Sigmoid activation function, attention weights g^h and g^w are obtained for the height and width directions, respectively. The following formulas describe this process:

$$g^h = \sigma(F_h(f^h)) \tag{14}$$

$$g^w = \sigma(F_w(f^w)) \tag{15}$$

After the computations outlined above, attention weights g^h for the height direction and g^w for the width direction will be obtained. Finally, these attention weights are used to perform element-wise multiplication on the original feature map, resulting in the final feature map that carries attention weights in both the width and height directions. The formula for this process is as follows:

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j) \tag{16}$$

IV. Experiments and Analysis

A. Introduction to Database

To validate the effectiveness of the proposed improvements to the YOLOv8s algorithm, this study utilizes the Chinese Traffic Sign dataset, TT100K, as its experimental dataset. TT100K is a dataset curated and released by a collaborative effort between Tsinghua University and Tencent's Joint Laboratory. The dataset is sourced from six high-resolution wide-angle digital cameras and comprises panoramic images captured in various cities across China. The lighting conditions and weather at the capture locations vary significantly. The TT100K dataset encompasses a comprehensive range of traffic sign classes featuring 221 distinct categories. However, the dataset exhibits class imbalance due to variations in capturing difficulty. The dataset was cleaned to address this, and 45 traffic sign classes with instance counts greater than 100 were selected. The training set comprises 7,198 images, while the testing set comprises 1,850 images.

B. Experimental Environment and Parameter Configuration

The original images in the TT100K dataset have a size of 2048x2048. Due to the high resolution and large dimensions of these images, the YOLOv8s algorithm resizes the input images uniformly to 640x640 before training. Additionally, data augmentation is performed using the Mosaic technique. The experimental setup and parameter configuration are presented in Table I.

TABLE I
Experimental Setup

Environment	Configuration
Operation platform	Windows10
CUDA	v11.3
Batch size	32
Initial learning rate	0.001
Momentum	0.937
Pytorch	v11.3
GPU	RTX4070ti

C. Evaluation Index

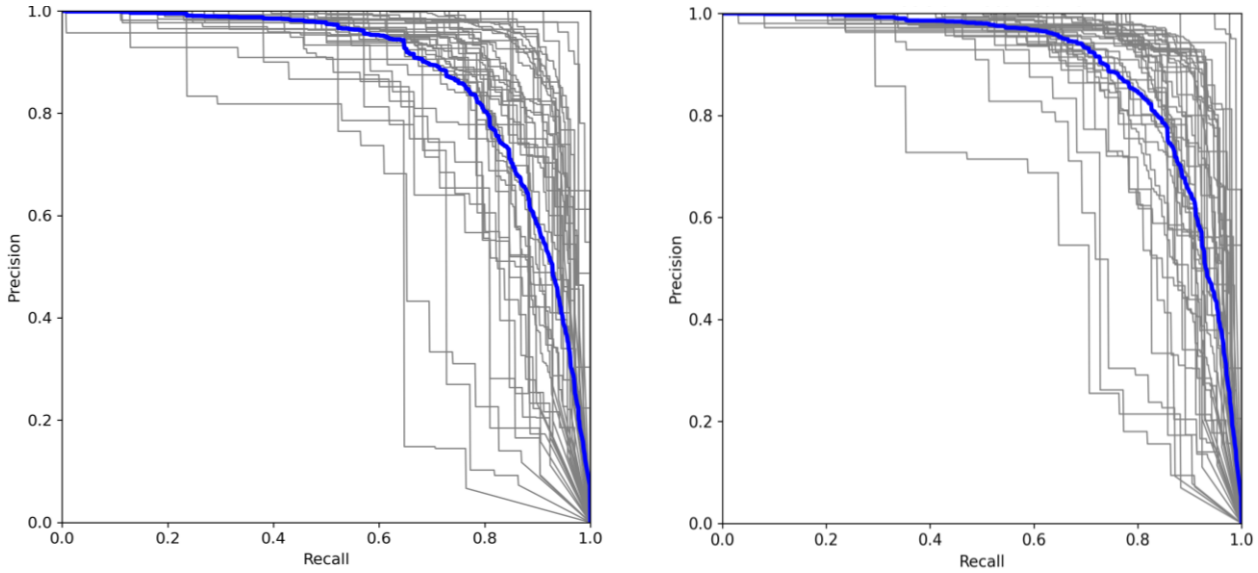
To visually demonstrate the effectiveness of the improvements made to the YOLOv8s algorithm in this paper, several evaluation metrics are chosen to assess the algorithm's performance. These metrics include precision, recall, mean average precision (mAP), frames per second (FPS), and the Precision-Recall curve. Precision refers to the probability of correct detections among all positive samples, providing a direct measure of the model's false positive rate. The recall represents the probability of correctly detecting all positive samples among the total positive samples, offering insight into the model's false negative rate. Mean Average Precision (mAP) is the most crucial performance metric, encapsulating the model's detection performance across all categories. The calculation expression for mAP is as follows:

$$mAP = \frac{1}{n} \sum_{j=1}^{n} AP(j) \tag{17}$$

Where n =45, represents the 45 categories within the dataset employed in this paper. AP refers to the average precision for a specific category within the dataset. This configuration allows for a comprehensive evaluation of the algorithm's performance across the diverse range of 45 classes present in the dataset.

D. Experimental Results and Analysis

In the Precision-Recall (P-R) curve, P represents precision, and R represents recall. As a convention, recall is typically assigned to the horizontal axis, while precision is placed on the vertical axis. The P-R curve illustrates the relationship between precision and recall, showcasing how



(a) YOLOv8s-P-R curve before improvement

(b) YOLOv8s-P-R curve after improvement

Fig. 7. P-R curve comparison of YOLOv8s algorithm on TT100K dataset before and after improvement

they vary concerning each other. This relationship is depicted in Fig. 7, which presents the Precision-Recall curve plotted after conducting experiments on the TT100K dataset [18], as discussed in this paper.

In the experiment results on the TT100K dataset, Fig. 7(a) shows the P-R curve of the original YOLO8s algorithm, while Fig.7(b) shows the P-R curve of the improved YOLO8s algorithm. The blue curve represents the mAP@0.5 for all categories. In Fig. 7(a), the mAP@0.5 of the blue curve is 87.50%, while in Fig 7(b), it is 89.51%. A larger area under the curve indicates better model performance. The area under the blue curve in Fig. 7(a) is smaller than that in Fig. 7(b), which demonstrates the effectiveness of the improvements made to the YOLOv8s algorithm in this paper.

E. Ablation Study

In this paper, three improvements were made to the YOLOv8s algorithm. To demonstrate the positive effects of each module on the original algorithm, the following ablation experiments were conducted:

(1) Only the proposed dual-layer semi-composite backbone network was used in place of the backbone network in YOLOv8s.

(2) Only the coordinate attention mechanism was added after the SPPF layer of the original algorithm.

(3) Only the context aggregation block was added to the Neck part of the original algorithm.

(4) The dual-layer semi-composite backbone network, coordinate attention mechanism, and context aggregation block were applied to the YOLOv8s algorithm altogether, validating the effectiveness of each module and the overall effectiveness of applying all modules to the original network.

The experimental results are presented in Table II. The ablation experiment data in Table II demonstrates the effectiveness of the improvements made to the original algorithm in this paper.

The proposed Dual-layer Semi-Composite Backbone (DSCB) structure improves mAP@0.5 by 1.5% while only adding 0.7M parameters, increasing 9.3G FLOPs, and boosting FPS by 4. Integrating the Coordinate Attention (CA) mechanism into the original algorithm increases mAP@0.5 by 0.92%, adds 10.2M parameters, increases 53.5G FLOPs, and decreases FPS by 6. Incorporating the Context Aggregation Block (CAB) into the original algorithm increases mAP@0.5 by 0.51%, adds 0.7M parameters, increases 0.9G FLOPs, and decreases FPS by 1. From the experimental data alone, it can be observed that adding the CA attention mechanism significantly increases model complexity, leading to a decrease in FPS and a slight increase in mAP.

TABLE II
ABLATION EXPERIMENT OF YOLOV8S ALGORITHM ON TT100K DATASET

Models	DSCB	CA	CAB	P (%)	R (%)	mAP@0.5 (%)	#Params(M)	FLOPs(G)	FPS
YOLOv8s	×	×	×	88.24	79.52	87.50	11.1	28.5	105
YOLOv8s+ DSCB	√	×	×	89.38	80.60	89.00	11.8	37.8	109
YOLOv8s+ CA	×	√	×	86.01	81.60	88.42	22.3	81.8	99
YOLOv8s+ CAB	×	×	√	88.17	81.44	88.01	11.8	29.4	104
Ours	√	√	√	90.51	82.00	89.51	12.5	38.7	106

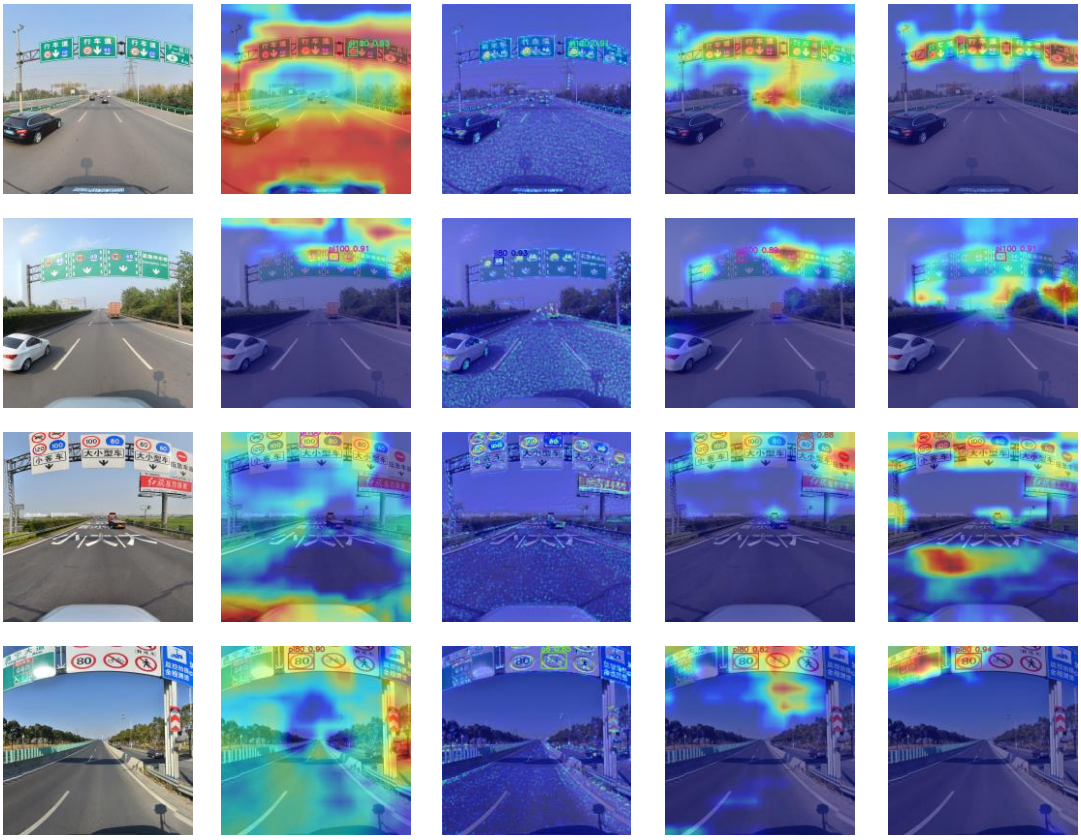


Fig. 8. Comparison of heatmaps of different improved modules

However, when the three proposed improvement modules are combined and applied to the original algorithm, the adverse effects caused by the CA attention mechanism are mitigated, thanks to the design of the DSCB structure. The DSCB structure has fewer parameters and lower complexity in the model's shallow stages compared to the original model's detection stage.

Ultimately, this paper achieves a 2.01% improvement in mAP@0.5 over the original algorithm. Although there is a slight increase in parameters and FLOPs, it does not affect the real-time detection performance of the model. On the contrary, the FPS is ultimately increased by 1. The proposed improvements not only enhance the detection accuracy of the model but also ensure real-time detection performance, striking a balance between detection accuracy and real-time capability.

As shown in Fig. 8, this paper incorporates different improvement modules into the baseline model. Four images were randomly selected, and the effects of integrating the original algorithm with each module were visualized using heatmaps. It can be observed that without integrating any module, the baseline model extracts features more broadly without a specific focus on traffic signs. However, when the DSCB module is integrated, the model focuses on traffic signs. The model further emphasizes the traffic signs by integrating the CA attention mechanism. Similarly, incorporating the CAB module produces results superior to the baseline model.

F. Performance Comparison of The Two DSCB Modules

In this paper, a comparative experiment was conducted on the two feature fusion methods within the DSCB module, as shown in Table III. The performance comparison between the two DSCB structures is presented. DSCB1 shows a slightly lower detection accuracy compared to DSCB2. However, considering the overall performance in terms of recall rate, mAP@0.5, parameter count, and FPS, DSCB1 outperforms DSCB2. Based on the experimental results mentioned above, although the DSCB2 structure shows improvement compared to the original algorithm, this paper selects the DSCB1 structure, which demonstrates better performance, for further experiments. We can see from Table III that the performance of DSCB1 structure is significantly better than that of DSCB2 structure.

TABLE III
PERFORMANCE COMPARISON OF TWO DIFFERENT DOUBLE-LAYER SEMI-COMPOSITE BACKBONE NETWORKS

Network Structure	P(%)	R(%)	mAP@0.5(%)	#Params(M)	FLOPs(G)	FPS
DSCB1	89.38	80.60	89.00	11.8	37.8	109
DSCB2	90.40	78.41	88.62	12.2	36.4	100

TABLE IV
PERFORMANCE COMPARISON WITH MAINSTREAM OBJECT DETECTION ALGORITHMS

Model Name	P(%)	R(%)	mAP(%)	# Parameters (M)	FLOPs(G)	FPS
YOLOv3	73.36	72.14	73.79	61.6	77.7	50
YOLOv5s	82.21	77.31	83.35	7.1	16.1	136
Faster R-CNN	75.54	75.20	74.58	41.2	91.1	24
YOLOv4-tiny	76.62	75.43	76.80	5.9	6.9	131
TRD-YOLO	86.30	81.19	86.50	12.6	26.0	73
YOLOv7-tiny	71.12	73.23	72.82	6.2	13.8	142
YOLOv8s	88.24	79.52	87.50	11.1	28.5	105
Ours	**90.51**	**82.00**	**89.51**	**12.5**	**38.7**	**106**

G.Compared with the Performance of Advanced Object Detection Algorithms

To demonstrate the advantages of the proposed method in traffic sign detection algorithms, this paper validated the proposed algorithm on the TT100K dataset. In the evaluation, a comparison was made with YOLOv3 [19], YOLOv4-tiny [20], YOLOv5s, Fast-RCNN, TRD-YOLO [21], YOLOX-s [22], and YOLOv7-tiny algorithms. The performance of the models was evaluated based on detection accuracy, recall rate, model parameter count, floating-point operations per second (FLOPs), mAP@0.5, and frames per second (FPS). The specific results can be found in Table IV.

From the data in Table IV, it can be observed that the proposed model outperforms other classical algorithms in terms of precision, recall rate, and mAP@0.5 evaluation metrics. Although the proposed model has a higher parameter count than YOLOv8s by 1.4M and an increase of 10.2G in FLOPs compared to YOLOv8s, the proposed model achieves one more FPS in terms of real-time performance evaluation compared to YOLOv8s. These experimental results indicate that even with a slight increase in model parameters and computational complexity, the real-time detection capability of the model may not necessarily be affected and may even improve slightly. Comparing with other mainstream algorithms, although the proposed model has lower FPS than YOLOv5s, YOLOv4-tiny, and YOLOv7-tiny, its detection accuracy is superior to these mentioned algorithms.

H.Detection on Random Images

In Fig. 9, the detection results of YOLOv8s and the improved version of YOLOv8s on the TT100K dataset are presented. The top three images in Fig. 9 represent the detection results of the YOLOv8s algorithm, while the bottom three images represent the detection results of the improved version of the YOLOv8s algorithm. From the comparative images, we can observe that the improved version of the YOLOv8s algorithm demonstrates superior detection performance compared to the original YOLOv8s algorithm.

Fig. 9. Comparison of the detection effect of YOLOv8s and the improved YOLOv8s algorithm

V. Conclusion

In this paper, we propose an improved traffic sign detection algorithm based on the YOLOv8s algorithm, which outperforms current mainstream object detection algorithms. The key improvements in this paper are as follows: We introduce a new Dual-layer Semi-Composite Backbone (DSCB) structure that enhances the capability of the backbone network to extract target features. This structure also reduces the complexity of the model's backbone network to some extent, resulting in improved detection performance and real-time capability. After the SPPF layer of YOLOv8s, we incorporate a Coordinate Attention mechanism to compensate for the loss of coordinate position information in feature extraction, particularly for small objects. This attention mechanism helps the model better focus on and localize traffic signs. We propose a new Context Aggregation Block (CAB) module to enhance feature representation. This module leverages global spatial context learning and aggregation at each level of the feature pyramid, further improving the detection performance and robustness of the model. Through these improvements, our traffic sign detection algorithm in this paper surpasses the performance of some current mainstream object detection algorithms. Additionally, it maintains good real-time performance while improving detection accuracy.

References

[1] N. Malarvizhi, A. K. Jupudi, M. Velpuri, and T. V. K. Dheeraj, "Autonomous Traffic Sign Detection and Recognition in Real Time," in *3rd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications, ICMISC 2022, March 28, 2022 - March 29, 2022*, Hyderabad, India, 2023, vol. 540: Springer Science and Business Media Deutschland GmbH, in Lecture Notes in Networks and Systems, pp. 415-423, doi: 10.1007/978-981-19-6088-8_36. [Online]. Available: http://dx.doi.org/10.1007/978-981-19-6088-8_36

[2] U. S. Rahman and Maruf, "Traffic Sign Detection and Recognition Using Deep Learning Approach," in *1st International Conference on Machine Intelligence and Emerging Technologies, MIET 2022, September 23, 2022 - September 25, 2022*, Noakhali, Bangladesh, 2023, vol. 490 LNICST: Springer Science and Business Media Deutschland GmbH, in Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, pp. 331-343, doi: 10.1007/978-3-031-34619-4_27. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-34619-4_27

[3] H. Zhang and J. Zhao, "Traffic Sign Detection and Recognition Based on Deep Learning," *Engineering Letters*, vol. 30, no. 2, pp. 666-673, 2022.

[4] N. Bhatt, P. Laldas, and V. B. Lobo, "A Real-Time Traffic Sign Detection and Recognition System on Hybrid Dataset using CNN," in *7th International Conference on Communication and Electronics Systems, ICCES 2022, June 22, 2022 - June 24, 2022*, Coimbatore, India, 2022: Institute of Electrical and Electronics Engineers Inc., in 7th International Conference on Communication and Electronics Systems, ICCES 2022 - Proceedings, pp. 1354-1358, doi: 10.1109/ICCES54183.2022.9835954. [Online]. Available: http://dx.doi.org/10.1109/ICCES54183.2022.9835954

[5] J. Cao, P. Li, H. Zhang, and G. Su, "An Improved YOLOv4 Lightweight Traffic Sign Detection Algorithm," *IAENG International Journal of Computer Science*, vol. 50, no. 3, pp.825-831, 2023.

[6] Z. Meihong, "Vehicle Detection Method of Automatic Driving based on Deep Learning," *IAENG International Journal of Computer Science*, vol. 50, no. 1, pp.86-93, 2023.

[7] J. L. Crowley, "Convolutional Neural Networks," in *18th European Advanced Course on Artificial Intelligence, ACAI 2021, October 11, 2021 - October 15, 2021*, Berlin, Germany, 2023, vol. 13500 LNAI: Springer Science and Business Media Deutschland GmbH, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 67-80, doi: 10.1007/978-3-031-24349-3_5. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-24349-3_5

[8] R. Girshick, "Fast R-CNN," in *15th IEEE International Conference on Computer Vision, ICCV 2015, December 11, 2015 - December 18, 2015*, Santiago, Chile, 2015, vol. 2015 International Conference on Computer Vision, ICCV 2015: Institute of Electrical and Electronics Engineers Inc., in Proceedings of the IEEE International Conference on Computer Vision, pp. 1440-1448, doi: 10.1109/ICCV.2015.169. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2015.169

[9] W. Liu *et al.*, "SSD: Single shot multibox detector," in *14th European Conference on Computer Vision, ECCV 2016, October 8, 2016 - October 16, 2016*, Amsterdam, Netherlands, 2016, vol. 9905 LNCS: Springer Verlag, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 21-37, doi: 10.1007/978-3-319-46448-0_2. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46448-0_2

[10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318-327, 2020, doi: 10.1109/TPAMI.2018.2858826.

[11] A. Nazir and M. A. Wani, "You Only Look Once - Object Detection Models: A Review," in *17th INDIACom; 2023 10th International Conference on Computing for Sustainable Global Development, INDIACom 2023, March 15, 2023 - March 17, 2023*, New Delhi, India, 2023: Institute of Electrical and Electronics Engineers Inc., in Proceedings of the 17th INDIACom; 2023 10th International Conference on Computing for Sustainable Global Development, INDIACom 2023, pp. 1088-1095.

[12] C. Yao, F. Wu, H.-J. Chen, X.-L. Hao, and Y. Shen, "Traffic sign recognition using HOG-SVM and grid search," in *2014 12th IEEE International Conference on Signal Processing, ICSP 2014, October 19, 2014 - October 23, 2014*, Hangzhou, China, 2014, vol. 2015-January: Institute of Electrical and Electronics Engineers Inc., in International Conference on Signal Processing Proceedings, ICSP, October ed., pp. 962-965, doi: 10.1109/ICOSP.2014.7015147. [Online]. Available: http://dx.doi.org/10.1109/ICOSP.2014.7015147

[13] C. Liu, F. Chang, and Z. Chen, "Traffic sign detection based on regions of interest and HOG-MBLBP features," *Dianzi Yu Xinxi Xuebao/Journal of Electronics and Information Technology*, vol. 38, no. 5, pp. 1092-1098, 2016, doi: 10.11999/JEIT150918.

[14] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv, 2022.

[15] J. Wang, Y. Chen, Z. Dong, and M. Gao, "Improved YOLOv5 network for real-time multi-scale traffic sign detection," *Neural Computing and Applications*, vol. 35, no. 10, pp. 7853-7865, 2023, doi: 10.1007/s00521-022-08077-5.

[16] T. Liang, H. Bao, W. Pan, and F. Pan, "Traffic Sign Detection via Improved Sparse R-CNN for Autonomous Vehicles," *Journal of Advanced Transportation*, vol. 2022, 2022, doi: 10.1155/2022/3825532.

[17] J. Hu, Z. Wang, M. Chang, L. Xie, W. Xu, and N. J. S. Chen, "PSG-Yolov5: A Paradigm for Traffic Sign Detection and Recognition Algorithm Based on Deep Learning," vol. 14, no. 11, p. 2262, 2022.

[18] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-Sign Detection and Classification in the Wild," in *29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, June 26, 2016 - July 1, 2016*, Las Vegas, NV, United states, 2016, vol. 2016-December: IEEE Computer Society, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2110-2118, doi: 10.1109/CVPR.2016.232. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2016.232

[19] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv, 2018.

[20] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv, 2020.

[21] J. Chu, C. Zhang, M. Yan, H. Zhang, and T. Ge, "TRD-YOLO: A Real-Time, High-Performance Small Traffic Sign Detection Algorithm," *Sensors*, vol. 23, no. 8, 2023, doi: 10.3390/s23083871.

[22] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," arXiv, 2021.