Proceedings of the 3rd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/145/2025.21875

YOLOv8-Based Road Sign Detection Algorithm Incorporating CBAM and DWConv

Zhichao Qian

School of Internet of Things Engineering, Jiangnan University, Wuxi, China
1758935769@qq.com

Abstract: Detecting road signs is essential for autonomous driving technology, especially in the identification of small objects.To overcome the difficulties of identifying tiny road signs, In order to increase detection performance, this work proposes an enhanced YOLOv8 method that combines Depthwise Separable Convolution (DWConv) with the Convolutional Block Attention Module (CBAM). Specifically, YOLOv8 serves as the baseline model, which is optimized in the feature extraction, fusion, and detection stages. The CBAM attention mechanism is incorporated into the Neck section, while traditional convolutions are replaced with DWConv, improving the model's focus on tiny information while reducing computational complexity. To improve the model's generalization ability, data augmentation methods like Mosaic and Mixup are incorporated. Mosaic augmentation increases the diversity of training data by stitching different images together, whereas Mixup improves the model's adaptability to various scenes by blending images. Additionally, common augmentation techniques, including cropping, color adjustment, and flipping, are effectively applied to optimize model performance. Experimental results indicate that, compared with YOLOv8n, the improved YOLOv8 algorithm achieves a 2.1 percentage point increase in mean Average Precision (mAP0.5), a 4.9% improvement in mAP50-95, and a 7.2% increase in recall rate. Furthermore, the algorithm significantly reduces the missed detection rate and improves small-object detection performance while lowering runtime by 4.1%. These results demonstrate the practical applicability of the proposed method.

Keywords: CBAM, DWConv, YOLOv8, small-object recognition, feature extraction

1. Introduction

With the rapid development of autonomous driving technology, road sign detection has become a crucial research area. One of the key challenges in this domain is detecting road signs, which are often small objects. This paper focuses on addressing this issue. It is essential to accurately recognize small road signs for the safety and decision-making capabilities of autonomous driving systems. Since road signs are frequently found in complex environments, are small in size, or are partially occluded, improving small-object detection accuracy can significantly enhance a system's ability to recognize distant or unclear signs. This ensures vehicles adhere to traffic regulations and respond promptly, thereby improving the reliability and safety of autonomous driving systems.

Many researchers globally have explored methods for small-object detection. Zhai et al. excluded detection components designed for larger objects and unnecessary layers to minimize model complexity, thereby boosting UAV detection efficiency [1]. Hao et al. implemented a dual-branch

DOI: 10.54254/2755-2721/145/2025.21875

framework with an attention mechanism to refine local feature extraction and developed a bidirectional pyramid network guided by attention to enhance feature distinctiveness [2].Feng et al. optimized the network structure by eliminating the backbone P5 layer, which is used for large-object detection, and instead merged the P4, P3, and P2 layers [3]. Xu et al. replaced the CSPDarknet53 backbone with the more lightweight MobileNetV3, effectively lowering the model's parameter count and computational burden while simultaneously improving its inference efficiency [4]. Li et al. incorporated the RFCBAM into the backbone for down sampling, which improved the efficiency of feature extraction and reduced the sparsity of spatial information typically caused by down sampling [5]. Huang et al. introduced the AFPN to emphasize key layer features after feature fusion while mitigating the impact of non-adjacent layer interactions. Zhang et al. designed an optimized multi-branch Cross Stage Partial (CSP) module along with a dual-path feature fusion framework to enhance feature integration, particularly for small-scale traffic signs. Swastik Saxena et al. adopted Generalized Intersection over Union (GIoU) in place of the conventional IoU as a distance metric. Their refined model incorporated an improved PANet with grouped convolutional layers in the detection neck and introduced an extra feature scale to enhance the recognition of smaller traffic signs. Marco Flores-Calero et al. conducted extensive evaluations using YOLO, confirming its practicality and superiority in traffic sign detection [9], which provided direction for this study. Ruturaj Mahadshetti et al. proposed Sign-YOLO, an attention-based one-stage method integrating YOLOv7 with the squeeze-and-excitation (SE) model and a special attention mechanism to overcome small-object detection challenges [10]. Their approach effectively reduced computational costs while enhancing the robustness of feature extraction.

The Convolutional Block Attention Module improves the model's proficiency in emphasizing important features by merging channel and spatial attention mechanisms, optimizing how features are extracted. Channel attention alters the influence of each feature channel depending on its relevance, while spatial attention assigns importance to specific spatial regions, sharpening the model's focus on vital areas. This method empowers the model to accurately identify and detect small objects, even in complex and cluttered environments. Depthwise Separable Convolution (DWConv), on the other hand, separates depthwise and pointwise convolutions, significantly reducing computational costs while maintaining feature extraction effectiveness. This method allows the model to efficiently process large-scale datasets and improve detection accuracy.

In this study, we integrate DWConv and CBAM into YOLOv8 by incorporating CBAM into the Backbone module and replacing standard convolution kernels with DWConv. Additionally, data augmentation techniques are employed to enhance feature representation. These enhancements ultimately boost the model's recall and precision.

2. Background Knowledge

YOLO is a deep learning-based object detection algorithm that frames the detection process as a regression task, directly mapping an image to the prediction of bounding boxes and class labels. As a one-stage detection method, YOLO's greatest advantage over traditional object detection techniques lies in its speed. It simultaneously predicts all bounding boxes and class labels in one forward pass, greatly enhancing the efficiency of detection.

The core principles of YOLO focus on bounding boxes and confidence scores. Each bounding box is defined by five parameters: x, y, which represent the coordinates of the box's center relative to the grid; w, h, the width and height of the box; and the Confidence Score, which reflects the predicted likelihood that the box contains an object. The confidence score is calculated as follows:

$$\text{Confidence} = \Pr(\text{Object}) \times \text{IoU (Intersection over Union)} \quad (1)$$

DOI: 10.54254/2755-2721/145/2025.21875

Since its initial release in 2016, YOLO has undergone continuous development and has now evolved to YOLOv11. The following sections briefly introduce several versions of the YOLO series.

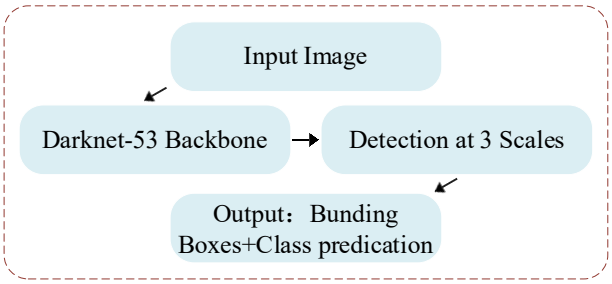

Figure 1: Simplified YOLOv3 Flowchart

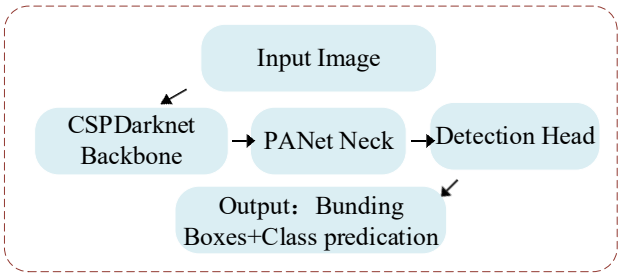

Figure 2: Simplified YOLOv5 Flowchart

As illustrated in Figure 1, YOLOv3 [11] primarily adopts Darknet-53 as its backbone. Darknet-53, a convolutional neural network built upon a residual structure, enables efficient extraction of high-level image features. In the Head Layer, multi-scale prediction is introduced, allowing the network to make predictions at three different scales. This enhances YOLOv3's ability to handle objects of various sizes. Each scale outputs a convolutional layer that generates multiple bounding box predictions (position, confidence, and class probability). The final output consists of bounding boxes along with their predicted class labels.

As shown in Figure 2, YOLOv5 [12] follows a similar detection pipeline but introduces notable differences in the Backbone and Neck layers compared to YOLOv3. In the Backbone, YOLOv5 incorporates CSPDarknet (Cross-Stage Partial Darknet) as its primary architecture. CSPDarknet optimizes feature extraction efficiency using the Cross-Stage Partial structure. In the Neck layer, YOLOv5 introduces PANet (Path Aggregation Network) to enhance feature fusion. PANet significantly improves object detection, particularly for small and densely packed objects. These advancements highlight that the core of the YOLO algorithm lies in the design of its Backbone and Neck layers.

3. Methods Used in This Study

This study primarily utilizes the Convolutional Block Attention Module (CBAM), supported by Depthwise Separable Convolution (DWConv), to improve detection performance in the YOLOv8 algorithm.

DOI: 10.54254/2755-2721/145/2025.21875

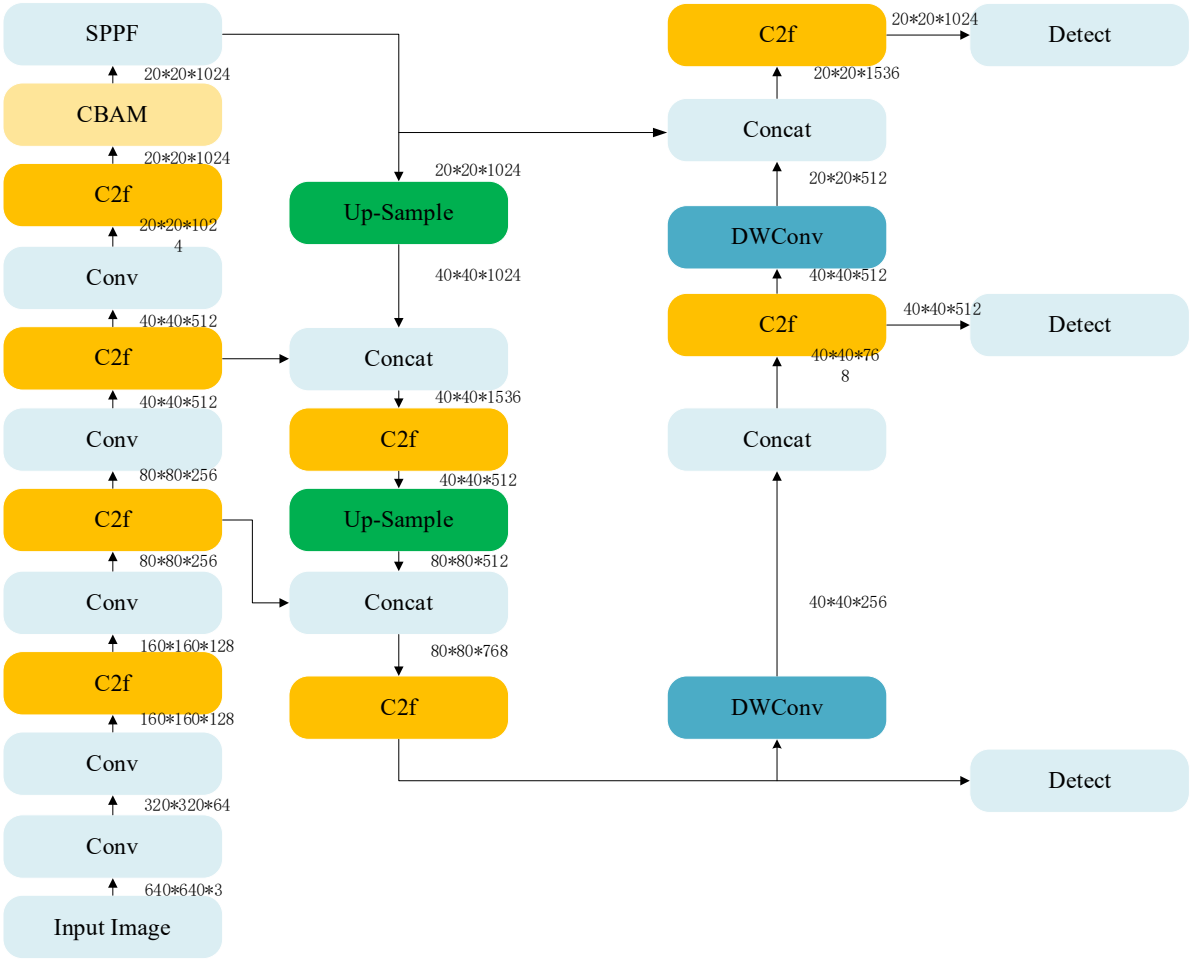

Figure 3: Improved YOLOv8 Method

The diagram above illustrates the improvements made to the YOLOv8 algorithm [13]. Serving as the foundation of this model, the Backbone network, shown in the leftmost column, is essential for feature extraction from the input image. The middle two columns correspond to the Neck, which handles feature fusion and refinement. The final column represents the Head, which serves as the decision-making component of the object detection model, generating the ultimate detection results. In this improved version, CBAM is added after the third C2f module, and the last two standard convolution layers in the Neck are replaced with DWConv.

3.1. CBAM

CBAM (Convolutional Block Attention Module) [14] enhances the network's focus on crucial features through the combination of Channel Attention and Spatial Attention mechanisms.

CBAM Pseudocode

CBAM
Input: input_feature // Input feature map
Output: output_feature // Output feature map
begin
Step 1: Channel Attention Module (CAM)
1: 1.1 Apply Global Average Pooling and Global Max Pooling
2: avg_pool = GlobalAveragePooling(input_feature) // Shape: [batch_size, channels]

DOI: 10.54254/2755-2721/145/2025.21875

```
    max_pool = GlobalMaxPooling(input_feature)          // Shape: [batch_size, channels]
    1.2 Apply Fully Connected (FC) layers
    avg_fc = FullyConnected(avg_pool)   // Shape: [batch_size, channels]
    max_fc = FullyConnected(max_pool)   // Shape: [batch_size, channels]
    1.3 Combine avg_fc and max_fc using Sigmoid activation
    channel_attention = Sigmoid(avg_fc + max_fc)   // Shape: [batch_size, channels]
    1.4 Apply channel attention to input feature map
    channel_weighted = input_feature * channel_attention   // Broadcasting
  Step 2: Spatial Attention Module (SAM)
    2.1 Apply Channel-wise Max and Average Pooling
    max_pool_spatial = MaxPooling(channel_weighted)   // Shape: [batch_size, height, width]
    avg_pool_spatial = AvgPooling(channel_weighted)   // Shape: [batch_size, height, width]
    2.2 Concatenate max_pool_spatial and avg_pool_spatial
    spatial_concat = Concatenate(max_pool_spatial, avg_pool_spatial)   // Shape: [batch_size, height, width, 2]
    2.3 Apply 3x3 Convolution to generate spatial attention map
    spatial_attention = Conv2D(spatial_concat)
    spatial_attention = Sigmoid(spatial_attention)   // Shape: [batch_size, height, width]
    2.4 Apply spatial attention to feature map
    spatial_weighted = channel_weighted * spatial_attention   // Broadcasting
    Final output after both channel and spatial attention
    output_feature = spatial_weighted
  end
```

Pseudocode Explanation

Channel Attention (CAM) creates feature maps by applying Global Average Pooling (GAP) and Global Max Pooling (GMP), which yield two distinct descriptors capturing channel-specific information. These descriptors are then passed through shared fully connected (FC) layers, followed by ReLU and Sigmoid activations to generate the channel attention weights. The resulting weights are applied to the original feature map using element-wise multiplication, enhancing the relevance of more important channels.

Spatial Attention (SAM) improves the localization of spatial features by performing Max Pooling and Average Pooling along the channel axis, generating two distinct spatial maps. These maps are then merged and passed through a 3×3 convolution to form a spatial attention map. The Sigmoid activation function is used to normalize the attention weights within the interval [0,1], which are then applied to the feature map using element-wise multiplication. By integrating these two components, CBAM enhances the model's focus on essential features while reducing the impact of irrelevant background noise, leading to better results in image classification and object detection.

3.2. DWConv

DWConv is a computationally efficient convolution technique that decomposes standard convolution into two separate operations:

(1) Depthwise Convolution (DW):

Applies an independent convolution kernel to each input channel, reducing computational complexity:

DOI: 10.54254/2755-2721/145/2025.21875

$$Y_{i,j}^{(d)} = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} X_{i+m,j+n}^{(d)} \cdot K_{m,n}^{(d)} \quad (2)$$

where: $Y^{(d)}$ is the output of the depthwise convolution, $K^{(d)}$ is the independent convolution kernel for each channel, d denotes the input channel index.

(2) Pointwise Convolution (PW):

Uses 1×1 convolution to combine the depthwise convolution outputs:

$$Y_{i,j}^{(p)} = \sum_{d=0}^{C} Y_{i,j}^{(d)} \cdot W_d^{(p)} \quad (3)$$

where: C is the number of input channels, $W_d^{(p)}$ represents the weights of the pointwise convolution.

By applying DWConv, each input channel is processed separately before the results are mixed in the pointwise convolution stage. This dramatically reduces the number of computations compared to standard convolution while maintaining high feature extraction efficiency.

3.3. Data Augmentation

The dataset used contains a variety of traffic signals, but its overall size is relatively small. To mitigate data scarcity issues, data augmentation techniques such as Mosaic and Mixup were implemented.

Mosaic Augmentation combines multiple images into a single image to expose the model to more diverse scenarios. Each training iteration presents different backgrounds, objects, and spatial layouts. This forces the model to capture a wider range of object distributions and background variations. By merging multiple objects and backgrounds, the model learns to distinguish spatial relationships among different objects. Other augmentation parameters used are listed in Table 1:

Table 1: Data Augmentation Parameters

Parameter	Value	Description
mosaic	1.0	Enables image-mosaic augmentation to increase dataset diversity.
mixup	0.5	Mixes two images in a weighted manner for augmentation.
crop	0.3	Randomly crops 30% of the image for augmentation.
hsv_h	0.015	Randomly adjusts hue values.
hsv_s	0.7	Randomly adjusts saturation levels.
hsv_v	0.4	Randomly adjusts brightness levels.
degrees	0.0	Rotation angle range (0 = no rotation).
translate	0.1	Percentage of horizontal and vertical translation.
scale	0.5	Scaling range, typically between 0.5 and 2.0.
shear	0.0	Shearing transformation (0 = disabled).
flipud	0.5	50% probability of flipping the image vertically.
fliplr	0.5	50% probability of flipping the image horizontally.

4. Experimental Results

The dataset used is the publicly available YOLOv8 Traffic Sign Dataset. Upon reviewing the dataset, it was observed that some small targets lacked annotations. To address this issue, MakeSense was utilized to manually label certain unannotated small road signs. Additionally, a portion of the dataset was supplemented with manually labeled small-object samples to ensure comprehensive training data. This study classifies road signs into four categories: Speed Limit, Crosswalk, Traffic Sign, Stop

DOI: 10.54254/2755-2721/145/2025.21875

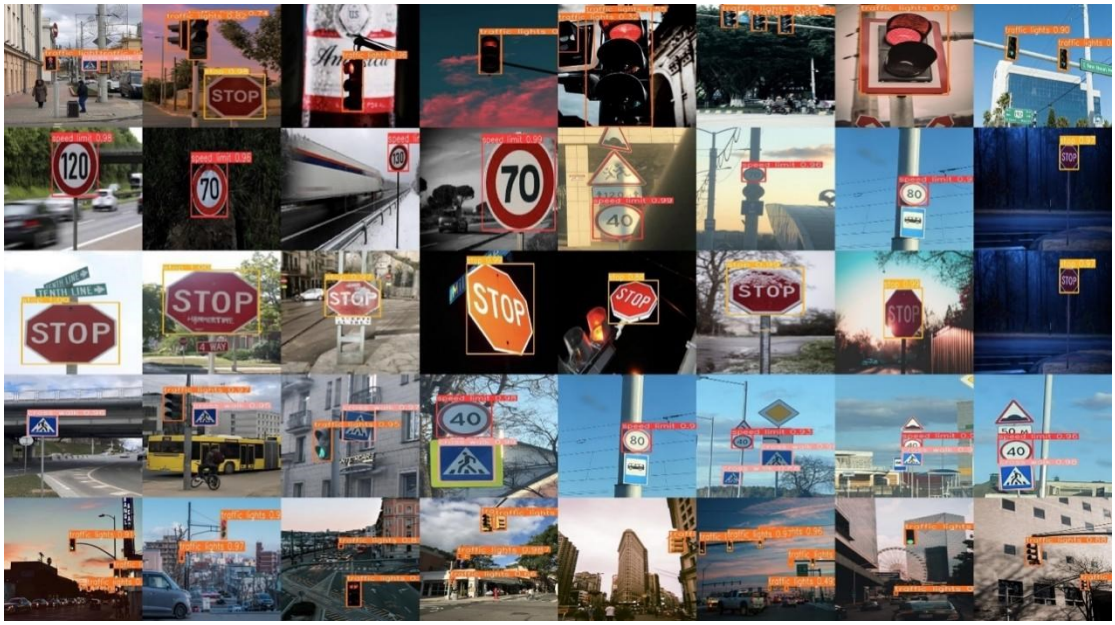

Figure 4: Detection Results

The image above presents the detection results, with the first four rows corresponding to: Traffic lights, Speed limit signs, Stop signs, Crosswalk signs. The last row consists of mixed-class images containing multiple small objects.

Traffic Lights (Row 1): The algorithm accurately detects traffic lights under different weather conditions, providing precise bounding boxes and correct classifications. However, for traffic lights that are turned off, the confidence score is relatively low.

Speed Limit Signs (Row 2): The model classifies these signs with high precision, with confidence scores generally exceeding 0.95. Even in challenging scenarios such as reflective surfaces or nighttime conditions—where even humans struggle to recognize signs—the model achieves accurate predictions.

Stop Signs (Row 3): The algorithm performs well even in snow-covered and glare-affected conditions, making correct detections.

Crosswalk Signs (Row 4): Although the model detects more objects in the images, it still maintains accurate predictions. Small-Object Detection.

Small-Object Detection (Row 5): The model effectively detects road signs that need to be read while the vehicle is in motion. Even objects that appear very small in the camera view are accurately enclosed in bounding boxes with high confidence scores—performing at a level that sometimes surpasses human vision. During training, multiple hyperparameter adjustments were made. It was found that by epoch 20, the model's performance metrics had already reached near saturation.

DOI: 10.54254/2755-2721/145/2025.21875

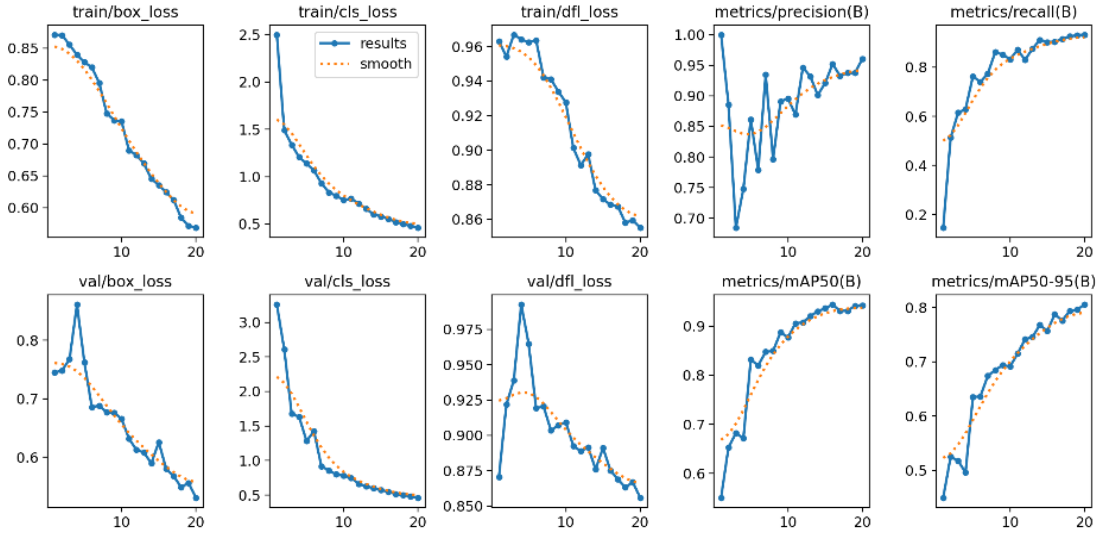

Figure 5: Training Process

The final training results are as follows:
Bounding Box Accuracy: 96%
Overall Recall: 93.2%
mAP50: 94.2%
The improvements over the original YOLOv8n model are as follows:
Bounding Box Accuracy increased by 2.4%
Overall Recall improved by 7.2%
mAP50 increased by 2.1%

These results demonstrate a significant reduction in the missed detection rate while also improving the accuracy of model predictions.

5. Conclusion

Road sign detection in autonomous driving presents challenges due to the small size of road signs and the impact of weather conditions, which can cause signs to appear blurred or unclear. To address these issues, this study incorporates CBAM to adaptively correct important feature representations in feature maps, and DWConv to decrease the number of parameters required for model training, accelerating computation speed. By integrating these two mechanisms, the YOLOv8n architecture was improved, making it better suited for small-object detection in road sign recognition. This enhancement significantly reduces missed detections, a common issue in YOLOv8n for small targets. The enhanced model exhibits better detection performance than other models and achieves high precision while maintaining a lightweight architecture.This method can be practically applied in real-world scenarios.

References

[1] *Zhai X, Huang Z, Li T, et al. YOLO-Drone: an optimized YOLOv8 network for tiny UAV object detection[J]. Electronics, 2023, 12(17): 3664.*

[2] *Yi H, Liu B, Zhao B, et al. Small object detection algorithm based on improved YOLOv8 for remote sensing[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2023.*

[3] *Wang Y, Zhang J, Yang Z, etal. Improving extractive summarization with semantic enhancement through topic-injection based BERT model[J]. Information Processing & Management, 2024, 61(3): 103677.*

DOI: 10.54254/2755-2721/145/2025.21875

[4] Xu W, Cui C, Ji Y, et al. YOLOv8-MPEB small target detection algorithm based on UAV images[J]. Heliyon, 2024, 10(8).
[5] Li Y, Li Q, Pan J, et al. Sod-yolo: Small-object-detection algorithm based on improved yolov8 for uav images[J]. Remote Sensing, 2024, 16(16): 3057.
[6] Huang Z, Li L, Krizek G C, et al. Research on traffic sign detection based on improved YOLOv8[J]. Journal of Computer and Communications, 2023, 11(7): 226-232.
[7] Zhang Y, Liu H, Dong D, et al. DPF-YOLOv8: Dual Path Feature Fusion Network for Traffic Sign Detection in Hazy Weather[J]. Electronics, 2024, 13(20): 4016.
[8] Saxena S, Dey S, Shah M, et al. Traffic sign detection in unconstrained environment using improved YOLOv4[J]. Expert Systems with Applications, 2024, 238: 121836.
[9] Flores-Calero M, Astudillo C A, Guevara D, et al. Traffic sign detection and recognition using YOLO object detection algorithm: A systematic review[J]. Mathematics, 2024, 12(2): 297..
[10] Jin J, Zhang J, Zhang K, etal. 3D multi-object tracking with boosting data association and improved trajectory management mechanism[J]. Signal Processing, 2024, 218 (2024): 109367.
[11] Farhadi A, Redmon J. Yolov3: An incremental improvement[C]//Computer vision and pattern recognition. Berlin/Heidelberg, Germany: Springer, 2018, 1804: 1-6.
[12] Wu W, Liu H, Li L, et al. Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image[J]. PloS one, 2021, 16(10): e0259283.
[13] Wang G, Chen Y, An P, et al. UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios[J]. Sensors, 2023, 23(16): 7190.
[14] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.
[15] Liu G, Hu Y, Chen Z, et al. Lightweight object detection algorithm for robots with improved YOLOv5[J]. Engineering Applications of Artificial Intelligence, 2023, 123: 106217.