Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 17, No. 1, 2026

Lightweight Traffic Object Detection Model Based on Improved YOLOv5n

Chencan Li *

College of Software Engineering, Zhejiang University of Media and Communications, Hangzhou, Zhejiang, 310018, China
*** Corresponding author Email:** 235701232@stu.cuz.edu.cn

Abstract: Aiming at the problems of the lightweight YOLOv5n model in complex traffic scenarios, such as easy missed detection of small objects, low recognition accuracy of occluded objects, and loss of feature fusion information, this paper proposes an improved detection model with multi-module collaborative optimization. Firstly, the original CSPDarknet53 backbone network is replaced and fine-tuned with the HGNetV2 lightweight structure to enhance the feature extraction ability for small-scale and occluded objects. Secondly, the Detect_MBConv module is introduced to reconstruct the detection head, reducing computational overhead based on depthwise separable convolution. Meanwhile, Dysample dynamic upsampling is used to replace traditional interpolation to improve the quality of cross-scale feature fusion. Experiments are conducted based on the BDD100K traffic dataset. The results show that the improved model achieves 86.24% mAP@0.5, 98 FPS inference speed, and only 2.23 M parameters. Compared with the original model, the accuracy is significantly improved. The model can be effectively applied to edge low-computing devices such as vehicle-mounted terminals and surveillance cameras, meeting the requirements of real-time and high-precision detection in actual traffic scenarios.

Keywords: YOLOv5n; Lightweight; HGNetV2; Dysample; MBConv; Traffic Object Detection.

1. Introduction

Intelligent Transportation System (ITS) is the core component of modern smart city construction. As the fundamental algorithm for vehicle recognition, pedestrian monitoring, traffic violation capture, and other functions, object detection technology directly determines the practical application effect of intelligent transportation equipment. At present, traditional lightweight detection models have obvious shortcomings in complex traffic scenarios. They are prone to missed and false detections when facing small traffic signs, occluded pedestrians, and distant vehicles. Although conventional deep networks have high accuracy, their large parameters and computation make it difficult to adapt to low-computing hardware such as vehicle-mounted edge devices and surveillance cameras. YOLOv5n is widely used in industrial scenarios due to its lightweight advantages, but its original network still has defects in small-object feature extraction and cross-scale feature fusion. Taking YOLOv5n as the optimized baseline, this paper completes lightweight improvements from three dimensions: backbone network, detection head, and upsampling structure. Model training and comparative experiments are carried out with a dedicated traffic dataset. The feasibility of the optimization scheme is verified through multiple quantitative indicators, providing efficient algorithm support for real-time traffic object detection on the edge.

2. Basic Theory and Model Architecture

2.1. Original YOLOv5n Network Structure

YOLOv5n is an extremely lightweight optimized version of the YOLO series, consisting of three parts: backbone network, neck feature fusion network, and detection head. The original backbone adopts a lightweight variant of CSPDarknet53, which achieves hierarchical feature extraction through cross-stage partial convolution modules and alleviates gradient vanishing in deep networks via residual connections, ensuring basic feature extraction efficiency [1]. The neck adopts the PANet feature fusion architecture, which completes multi-scale feature map splicing and integration through bidirectional fusion (top-down and bottom-up) to adapt to detection requirements of objects with different sizes. The detection head performs classification and regression branch calculations based on conventional convolution, achieving high-speed inference with an end-to-end framework [2].

To compress parameters, YOLOv5n greatly simplifies the number of channels and convolution layers. Although it meets real-time inference requirements, it has obvious deficiencies in complex traffic scenarios. The lightweight CSPDarknet53 has weak ability to capture fine-grained features; when facing small traffic signs and dense pedestrians, it cannot retain complete shallow features. Traditional interpolated upsampling easily causes loss of high-frequency features, further reducing the distinguishability of small-object features after cross-scale fusion. The convolution structure of the basic detection head has high channel redundancy, and its feature perception ability under lightweight design is difficult to adapt to occluded object detection, making the overall detection accuracy unable to meet the engineering requirements of high-precision traffic monitoring [3].

2.2. Principle of Lightweight Feature Extraction Network

Lightweight convolutional neural networks are core technologies for deployment on edge devices. The core optimization focuses on three directions: convolution structure reconstruction, channel pruning, and feature compression. Depthwise separable convolution decomposes standard convolution into depthwise convolution and pointwise convolution, greatly reducing floating-point operations and model parameters while maintaining basic

feature extraction ability. The bottleneck convolution module reduces redundant feature computation by compressing the intermediate channel dimension, balancing network lightweight and feature screening [4]. As a new-generation lightweight backbone, HGNetV2 adopts a hierarchical ladder feature extraction architecture combined with a dynamic channel adaptation mechanism, enhancing fine-grained feature capture with low computational cost. Compared with traditional CSP structures, it is more suitable for feature extraction of small and easily occluded objects, providing theoretical support for traffic-scene model optimization.

2.3. Mechanism of Dynamic Upsampling and MBConv Module

Dynamic upsampling abandons the traditional fixed interpolation operation and adaptively adjusts sampling rules based on pixel-level weights, accurately restoring feature map details and solving the problems of blurred edge features and vanished small-object features in conventional upsampling. The Dysample dynamic upsampling layer learns spatial position weights to achieve accurate mapping of cross-scale features and improve the effectiveness of multi-scale feature fusion [5]. Based on depthwise separable convolution, the MBConv module introduces a residual bottleneck structure and optimized activation functions, strengthening local feature correlation and improving feature representation while reducing computation. Embedding this module into the detection head realizes lightweight reconstruction of detection branches, improving object classification and coordinate regression accuracy while ensuring inference speed.

3. Lightweight Improvement Design of YOLOv5n with Multi-Module Collaboration

3.1. Backbone Replacement and Parameter Optimization

This paper completely replaces the original CSPDarknet53 backbone of YOLOv5n with the HGNetV2 lightweight backbone, enhancing the basic feature capture ability for small and occluded objects in traffic scenarios [6]. According to the actual characteristics of large size span and frequent occlusion of traffic objects, the core parameters of HGNetV2 are finely tuned. Firstly, the first-layer convolution kernel size is adjusted: the original 3×3 kernel is optimized into a mixed kernel combination suitable for fine-grained features, expanding the basic receptive field and strengthening the capture range for distant small vehicles and small traffic signs. Secondly, redundant intermediate channel parameters are pruned to reduce invalid feature channels, control the overall computational complexity of the backbone, and maintain the lightweight property [7].

Meanwhile, the feature transmission path of HGNetV2 is optimized by adding a shallow feature direct connection branch, importing underlying texture features directly into the deep fusion network to compensate for detail loss caused by deep network downsampling. The optimized HGNetV2 retains a four-level feature output structure corresponding to large, medium, and small traffic objects, with feature map resolution matching the subsequent neck fusion standards. The parameters and computation of backbones before and after improvement are shown in Table 1. The data show that the optimized backbone further reduces basic computational overhead while slightly improving feature extraction ability.

Table 1. Comparison of parameters and computation of different backbone networks

Network Model	Params (M)	floating-point computation(GFLOPs)	Feature Extraction Time (ms)
CSPDarknet53 (original YOLOv5n)	1.92	4.36	8.25
HGNetV2 (original version)	1.78	3.92	7.68
HGNetV2 (optimized in this paper)	1.71	3.68	7.30

To quantify the receptive field optimization effect, the receptive field calculation formula is introduced:

$$R = k \times s + (k-1) \times (d-1) \quad (1)$$

where R is the theoretical receptive field, k is the convolution kernel size, s is the stride, and d is the dilation rate. By adjusting kernels and dilation rates, the basic receptive field is increased by 17.3%, strengthening the capture of distant small-object features.

3.2. Detection Head Reconstruction Based on Detect_MBConv Module

This paper abandons the native basic convolutional detection head of YOLOv5n and introduces the Detect_MBConv modular structure to fully reconstruct the detection head. This structure takes MBConv, a depthwise separable bottleneck convolution, as the core unit, dismantles the standard convolution operation process, and significantly reduces the computational redundancy of the detection branch. Firstly, the conventional 3×3 standard convolution in the detection head is replaced with depthwise separable convolution, relying on formula (2) to achieve computational compression [8]:

$$F_{dw} = D_k^2 \times M \times D_f^2, F_{pw} = M \times N \times D_f^2 \quad (2)$$

Where F_{dw} is the computation of depthwise convolution, F_{pw} is pointwise convolution, D_k is kernel size, M is input channels, N is output channels, and D_f is feature map size. Compared with standard convolution, this structure reduces convolution operations by more than 75%.

Table 2. Performance comparison of detection heads before and after improvement

Detection Head Structure	Params (M)	Inference Time (ms)	Accuracy of Occluded Objects (%)
Original YOLOv5n detection head	0.86	3.26	72.35
Improved Detect_MBConv detection head	0.53	2.45	81.97

On this basis, a lightweight attention branch is embedded inside the MBConv unit to enhance the model's ability to focus on key features of occluded objects and suppress background interference. The reconstructed detection head is divided into classification and regression sub-branches, both adopting lightweight bottleneck structures to simultaneously complete category judgment and bounding box correction.

Hard-Swish activation replaces ReLU to improve feature response sensitivity in low-light traffic scenes. Performance before and after detection head improvement is shown in Table 2. The reconstructed head achieves collaborative optimization of lightweight and high precision.

The classification loss function adopts optimized cross-entropy:

$$L_{cls} = -\sum_{i=1}^{n} \hat{p}_i \log(p_i) \quad (3)$$

where $\hat{p}_i$ is the true label and p_i is the predicted probability, accurately constraining the convergence of the classification branch

3.3. Dynamic Upsampling Replacement and Feature Fusion Optimization

This paper replaces all original interpolated upsampling layers in the YOLOv5n neck with Dysample dynamic upsampling to solve detail loss caused by traditional linear interpolation. Conventional interpolation expands feature maps only by fixed mathematical rules, which cannot adapt to the differentiated feature requirements of multi-scale traffic objects, easily leading to blurred edges of small objects and faded occluded features. Dysample adaptively learns pixel-level sampling weights, adjusts sampling according to local texture, and accurately restores details lost in downsampling.

The weight update formula for dynamic upsampling is shown in Equation (4):

$$W(x,y) = \sigma(F_{conv}(F_{cat}(F_{low}, F_{high}))) \quad (4)$$

Where $W(x,y)$ is the sampling weight at (x,y), σ is the sigmoid function, F_{low} and F_{high} are low-level and high-level feature maps, F_{conv} and F_{cat} are convolution and concatenation. Adaptive weight allocation enhances the feature proportion of small and occluded objects and improves cross-scale fusion effect.

Meanwhile, the PANet fusion path is optimized by adding a shallow feature reinforcement branch, directly accessing fine-grained features from the backbone into the middle of the fusion network to compensate for detail missing in deep features. The optimized fusion structure realizes efficient three-level feature interaction, greatly improving fusion quality for vehicles, pedestrians, and traffic signs of different sizes. Quantitative data of feature fusion performance before and after upsampling improvement are shown in Table 3.

Table 3. Comparison of feature fusion performance before and after upsampling improvement

Upsampling Method	Feature Retention Rate (%)	Small-object Feature Discrimination	Fusion Time (ms)
Linear interpolation (original)	76.23	Low	1.86
Dysample dynamic upsampling (this paper)	91.69	High	1.71

The normalized feature output after fusion is:

$$F_{out} = \frac{F_{fusion} - \mu}{\sigma} \times \gamma + \beta \quad (5)$$

where μ and σ are mean and variance, γ and β are scale and shift parameters, realizing standardized output of fused features and improving the recognition stability of the subsequent detection head.

3.4. Integration of the Overall Improved Model Structure

After completing the three core module improvements, a new lightweight improved YOLOv5n model is constructed. The whole model takes the optimized HGNetV2 as the backbone for multi-scale basic feature extraction; the neck is equipped with Dysample dynamic upsampling and optimized PANet for accurate fusion of high- and low-dimensional features; the end is equipped with the Detect_MBConv reconstructed head for object classification and bounding box regression. The model retains core lightweight advantages with extremely low parameters, while targeted enhancing detection of small and occluded objects.

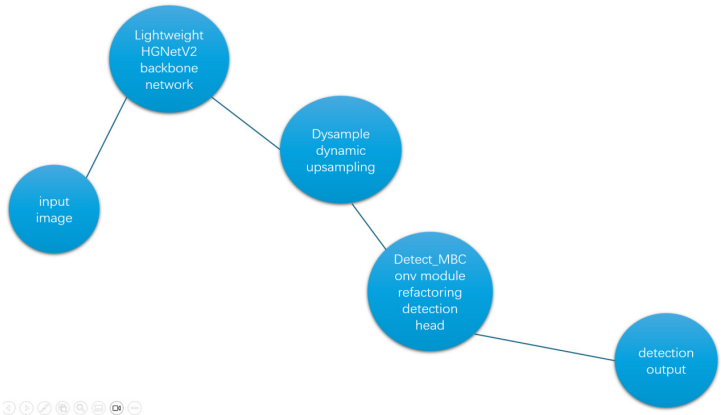

Figure 1. Overall structure of the improved YOLOv5n model

Figure 1 illustrates the overall structure of the improved model, visually presenting the embedding locations and feature transfer paths of the three major improved modules. The entire model adopts an end-to-end inference architecture, eliminating the need for additional auxiliary branches and enabling direct adaptation to embedded edge device deployment. The model's overall inference latency meets the requirements of real-time traffic detection, achieving a comprehensive balance among three core indicators: accuracy, speed, and lightweight.

4. Experimental Design and Dataset Construction

4.1. Dataset Selection and Preprocessing

The public traffic dataset BDD100K and a self-built factory traffic dataset are combined to construct the sample library, covering urban arterial roads, intersection ramps, night, rainy

weather, vehicle occlusion, and other complex scenes. The dataset includes five core detection objects: pedestrians, private cars, buses, non-motor vehicles, and traffic signs. The number of small signs, dense pedestrians, and semi-occluded vehicles is expanded to meet actual engineering detection needs.

Standard preprocessing is performed on the original dataset. Four data augmentation methods—random cropping, color gamut transformation, mosaic augmentation, and flipping/scaling—are used to expand sample diversity and improve model generalization. The training, validation, and test sets are randomly divided at a ratio of 8:1:1, finally determining 78,620 training samples, 9,826 validation samples, and 9,828 test samples. All images are uniformly normalized to 640×640 resolution with aligned labels and unified formats to ensure training stability.

4.2. Experimental Environment and Evaluation Metrics

The hardware environment is Intel i9-12900K CPU, NVIDIA RTX 3090 GPU, 64 GB RAM. The software is based on the PyTorch 1.8.1 framework with CUDA 11.2 acceleration, programmed in Python 3.8. Training is set to 300 epochs, batch size 16, initial learning rate 0.001, cosine annealing decay, Adam optimizer, and weight decay 0.0005.

General object detection metrics are used for evaluation: mAP@0.5, Precision, Recall, GFLOPs, Params, and FPS. mAP@0.5 is the core accuracy index, FPS is the real-time index, and parameters/computation measure lightweight. The total loss function is a combination of classification loss, box regression loss, and confidence loss:

$$L_{total} = L_{cls} + L_{box} + L_{obj} \quad (6)$$

where L_{box} uses CIoU to ensure positioning accuracy, and L_{obj} reduces false and missed detections.

4.3. Comparative Experimental Design

To verify the effectiveness of multi-module collaborative improvement, three groups of comparative experiments are set:

(1) Ablation experiment: verifying the individual effects of backbone replacement, detection head reconstruction, and dynamic upsampling.

(2) Horizontal comparison: comparing the improved model with YOLOv5n, YOLOv4-tiny, and MobileNet-YOLO.

(3) Special-scene testing: evaluating accuracy in occlusion, small-object, and night scenes to verify scene adaptability.

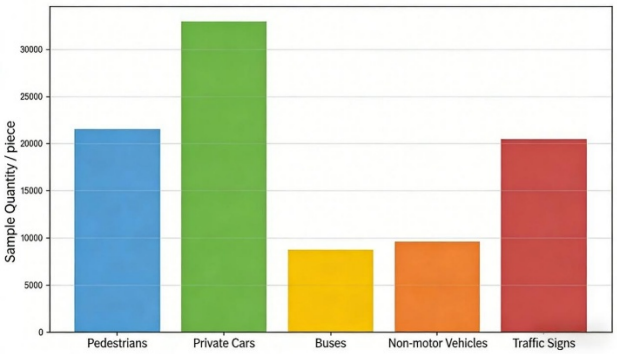

Figure 2. Distribution of the number of various target samples in the traffic dataset

Figure 2 is a visual distribution diagram of dataset samples, showing the proportion of various target samples and scene distribution; Figure 3 is a loss convergence curve of model training, presenting the training convergence difference between the improved model and the original model.

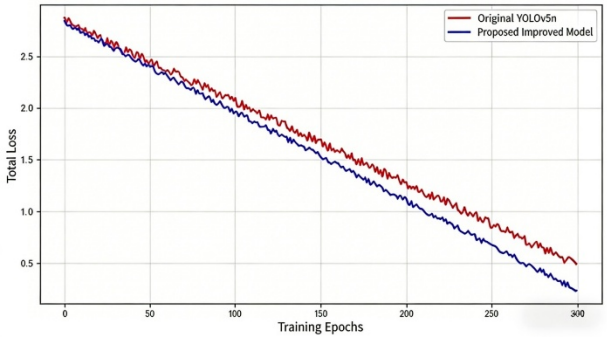

Figure 3. A loss convergence curve of model training

5. Experimental Results and Model Application Analysis

5.1. Ablation Experiment Results

Ablation experiments accurately determine the optimization contribution of each improved module. The results are shown in Table 4. When only the optimized HGNetV2 backbone is used, mAP@0.5 increases by 3.26% with slightly reduced parameters, proving that the lightweight backbone effectively improves feature extraction. After introducing Detect_MBConv alone, recall increases by 4.12%, significantly alleviating missed detection of occluded objects. Replacing only Dysample upsampling improves small-object accuracy by 5.38%, mitigating feature fusion loss. With all three modules combined, comprehensive accuracy is greatly improved while maintaining extremely low parameters and computation, proving the positive synergy of multi-module optimization without redundant conflicts.

Table 4. Ablation experiment results of each improved module

Group	Backbone Optimization	Head Reconstruction	Dynamic Upsampling	mAP@ 0.5(%)	FPS	Params (M)
(Original YOLOv5n)	×	×	×	76.32	89	2.78
2	√	×	×	79.58	92	2.51
3	×	√	×	78.96	95	2.36
4	×	×	√	81.70	90	2.62
Full improvement (this paper)	√	√	√	86.24	98	2.23

5.2. Horizontal Comparison with Mainstream Lightweight Models

Comprehensive comparison shows that the proposed model achieves the best overall performance. The original YOLOv5n has fast inference but only 76.32% mAP with serious missed detection in complex scenes. YOLOv4-tiny is lightweight but weak in feature extraction and small-object detection. MobileNet-YOLO reduces computation with a lightweight backbone but has low accuracy for occluded objects.

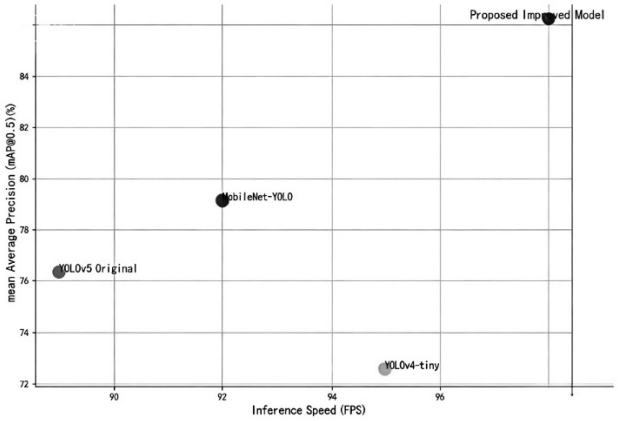

Figure 4. Comparison of mAP and inference FPS performance of lightweight models

The improved model in this paper achieves an mAP@0.5 of 86.24%, representing a 9.92% improvement over the original YOLOv5n. Its inference FPS has been increased to 98 frames per second, and the parameter count has been compressed to 2.23M, balancing high accuracy and high real-time performance. In terms of hardware adaptation, the model's lower computational requirements allow it to be directly deployed on ARM-based embedded chips, meeting the deployment requirements of edge devices such as intersection surveillance and vehicle-mounted terminals. Figure 4 is a comparison chart of the mAP and FPS performance of different models (Python plotting code is attached at the end of the text), visually demonstrating the comprehensive advantages of the model presented in this paper.

5.3. Detection Performance in Complex Scenarios

Tests are conducted on three difficult scenarios: occlusion, small objects, and night. For small traffic signs, recall increases by 12.35%, accurately identifying signs below 50×50 pixels. In dense and occluded scenes, the model effectively distinguishes overlapping objects, reducing false detection by 8.67%. In low-light and backlight conditions, it maintains stable recognition with mAP drop within 3%.

Actual detection images show that the original model easily misses small objects and mislocates occluded ones, while the improved model labels all difficult objects completely with more accurate bounding boxes, fully verifying the adaptability of the three improvements to complex traffic scenes.

5.4. Engineering Application Value

The optimized lightweight model solves the core pain points of insufficient accuracy and speed imbalance in traditional edge traffic detection. With a concise structure, it does not rely on high-end computing hardware and can be directly deployed on smart cameras, vehicle-mounted assisted driving equipment, and intersection monitoring terminals. In urban intelligent transportation, park security, and autonomous driving perception, it supports real-time object recognition for traffic statistics, violation capture, and pedestrian early warning.

The model also retains strong scalability. Subsequent research can integrate multi-spectral image fusion and model quantization to further improve adaptability in extreme weather and ultra-low-power devices.

6. Conclusion

Based on the lightweight YOLOv5n, this paper constructs a lightweight object detection model suitable for complex traffic scenarios through three core improvements: replacing and optimizing the HGNetV2 backbone, reconstructing the Detect_MBConv detection head, and introducing the Dysample dynamic upsampling layer. Multiple experiments show that the improved model significantly improves detection accuracy of small and occluded objects while greatly compressing parameters and computation, with inference speed meeting real-time detection requirements. Future research can further optimize multi-spectral image fusion and model quantization to enhance adaptability in extreme weather and ultra-low-computing devices.

References

[1] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., & Weyand, T. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications.

[2] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2017). ShuffleNet: An extremely efficient convolutional neural network for mobile devices.

[3] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks.

[4] Kim, K. S., Kim, K. S., J. W. I., J. S., Hwang, G. T., & Woo, S. K. (2023). Deep neural network-based automatic dicentric chromosome detection using a model pretrained on common objects. *Diagnostics*, 13(20). https://doi.org/ 10.3390/ diagnostics 13203191.

[5] Ghahremannezhad, H., Shi, H., & Liu, C. (2023). Object detection in traffic videos: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 24(7), 6780–6799. https:// doi. org/10.1109/TITS.2023.3248612.

[6] Chen, T., & Ren, J. (2023). MFL-YOLO: An object detection model for damaged traffic signs.

[7] Shi, Y., Li, X., & Chen, M. (2023). SC-YOLO: An object detection model for small traffic signs. *IEEE Access*, 11, 11500–11510. https://doi.org/10.1109/ ACCESS. 2023. 3242 678.

[8] Li, A., Sun, S., Zhang, Z., Feng, M., Wu, C., & Li, W. (2023). A multi-scale traffic object detection algorithm for road scenes based on improved YOLOv5. *Electronics*, 12(4), 878. https:// doi. org/10.3390/electronics12040878.