Improving Image Object Recognition Accuracy in V2X Environments Using the CBAM-MobileNet Model

Fan Luo[1], Xun Qiu[2]
[1]Secretarial office of Lincang Transportation Bureau, Lincang 677099, Yunnan, China
[2]Durability team in department of components and materials testing and evaluation, CAERI, Chongqing 401122, China
Corresponding author: Fan Luo (e-mail: luofan0623@126.com)

ABSTRACT Addressing the issue of low object recognition accuracy in real-time intelligent vehicle-to-everything (V2X) environments, where sparse information and severe occlusion of small objects often degrade perception performance, this paper proposes an engineered lightweight detection framework based on YOLOv8s. Reliable object perception in V2X systems is essential for intelligent wireless communication networks and electromagnetic sensing environments, where accurate interpretation of visual information supports cooperative perception and real-time decision-making. The proposed framework replaces the original backbone with a lightweight MobileNetV3-CBAM model, employing depthwise separable convolutions and the CBAM attention mechanism to improve computational efficiency while preserving fine-grained features. The neck adopts weighted bidirectional feature fusion to effectively integrate shallow high-resolution information with deep semantic representations through learnable upsampling and downsampling weights. Furthermore, the detection head is optimized using SIOU loss to enhance localization sensitivity and focal loss to alleviate category imbalance. Experiments conducted on the DAIR-V2X-C dataset demonstrate that the proposed method achieves an mAP@0.5 of 95.5% and an mAP @[0.5:0.95] of 73.4%. For small objects, the mAP@0.5 reaches 88.1%, while under an extreme occlusion level of 0 .2, the model still maintains an mAP@0.5 of 80.4%. Meanwhile, the average inference latency remains as low as 13.5 ms, demonstrating its suitability for real-time edge deployment. The proposed framework provides an efficient solution for lightweight perception in V2X systems and offers valuable technical references for intelligent electromagnetic sensing and wireless propagation-aware visual perception applications.

INDEX TERMS v2x environment, object recognition, CBAM-MobileNet model, electromagnetic sensing, wireless perception

I. INTRODUCTION

WITH the development of autonomous driving and assisted driving systems, vision-based perception plays a key role in intelligent connected vehicles[1-3]. However, V2X (Vehicle-to-Everything) environments are highly dynamic and complex. Vehicle speeds, diverse viewpoints, frequent occlusions, and environmental interference such as lighting and weather often result in sparse or partially lost target information in images. Furthermore, V2X systems often rely on onboard or roadside edge devices, which are limited in computing power, storage, and bandwidth. Improving the recognition accuracy of small and occluded objects while maintaining low latency has become an urgent engineering challenge and research hotspot.

While the default CSP-Darknet backbone in YOLOv8s performs well in general object detection, its high computational and memory overhead, along with some downsampling design, leads to the loss of important high-resolution details in edge devices and sparse/strongly occluded scenarios, reducing its sensitivity to small targets and locally visible features. This paper replaces the backbone with MobileNetV3 and embeds CBAM. MobileNetV3 significantly reduces FLOPs while preserving low-level resolution information through depthwise separable convolutions and inverse residual structures, ensuring rich local features even with limited computing power. CBAM adaptively amplifies and locates weakly salient regions in both channel and spatial dimensions, helping to recover sparse signals and enhance local cues of occluded targets. The combination of these two technologies theoretically balances the real-time requirements of edge deployment with improved model discrimination capabilities for small, partially visible targets in V2X scenarios.

This paper aims to address the degradation of small object recognition accuracy caused by sparse information and severe occlusion within the constraints of real-time V2X

perception, and proposes an engineered real-time edge deployment solution. Using YOLOv8s (You Only Look Once version 8 small) as a baseline, this paper systematically replaces and restructures its key modules. MobileNetV3 (Mobile Network version 3) replaces the original backbone to preserve high-resolution details in lower layers while reducing computational overhead. CBAM is inserted into several layers of the backbone to adaptively amplify residual target signals. Weighted bidirectional feature fusion is applied at the neck to enable learnable up-and-down propagation of multi-scale information. This weighted bidirectional fusion dynamically adjusts the relative contributions of shallow and deep features, allowing high-resolution edge information and deep semantic information to complement each other during fusion. This method preserves fine-grained features while suppressing semantic noise, improving responsiveness to small and partially visible objects. The detection head maintains a decoupled structure and replaces CIOU with SIOU to strengthen constraints on center offset and aspect ratio errors. SIOU applies more detailed penalties for center offset and shape differences during regression, making the predicted boxes more stable and closer to the true object outline when partially occluded or with only partially visible features, reducing errors caused by occlusion and improving recall.The average inference latency is 13.5 ms. Through targeted module replacement, attention enhancement, fusion, and loss reconstruction, this paper improves the detection accuracy and precision of small and occluded objects in V2X scenarios while meeting the resource and latency constraints of real-time edge deployment.

II. RELATED WORK

In the V2X environment, there have been many studies on image object recognition that aim to improve the recognition accuracy of vehicles and surrounding targets. Wu J proposed a visible light and infrared image fusion method based on Bayesian reasoning. Visible light targets were annotated by salient region detection, spectral residual, and EdgeBox algorithm, and vehicle recognition was achieved by combining infrared features. The accuracy was 77% in unobstructed scenes and 74% in partial occlusion, indicating that multimodal fusion can improve performance, but it is still not ideal for small objects and complex occlusions [4]. Nawaz A H proposed a V2X-OccluFusion framework. Through self-supervised masked BEV (bird's-eye view) feature reconstruction and lightweight state space fusion, multi-agent sharing of occlusion information was achieved. The performance was improved by 25.5% under severe occlusion, and real-time deployment was supported, but the complexity of multimodal fusion was relatively high [5]. To enhance the accuracy of object recognition under severe occlusion conditions, researchers have proposed a variety of methods to deal with occlusion, multi-scale and complex environment problems in autonomous driving and intelligent roadside perception. Ruan J reviewed the occluded object detection in autonomous driving in the past five years, dividing the research into vehicle, pedestrian, and traffic sign detection, and pointed out that the existing methods are still limited in terms of real-time performance, adaptability to complex scenes and small object recognition capabilities [6]. The above method improves the accuracy of occluded object detection through multi-scale feature fusion, attention mechanism, and multimodal data fusion, but it still has problems such as complex model and limited real-time performance, and insufficient performance in small object recognition and extreme occlusion under real-time guarantee, providing a research direction for lightweight, high-precision object recognition models that can adapt to complex V2X scenarios.

III. IMAGE OBJECT RECOGNITION FRAMEWORK IN V2X ENVIRONMENT

ORIGINAL YOLOV8S MODEL

In this paper, YOLOv8s is used as a baseline detector, whose core includes a lightweight backbone network, a feature fusion neck, and a decoupled detection head [7]. The original backbone of YOLOv8s adopts the CSP-Darknet (Cross Stage Partial-Darknet) architecture, whose input image undergoes a series of convolution and residual blocks to form a multi-scale feature map [8]. The convolution operation is shown in Formula (1).

$$Y_{i,j,c} = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} \sum_{c'=0}^{C_{\text{in}}-1} W_{m,n,c',c} \cdot X_{i+m,j+n,c'} + b_c \quad (1)$$

In Formula (1), k represents the convolution kernel size, and b_c represents the bias term.

In feature fusion, YOLOv8s uses PAN-FPN (Path Aggregation Network-Feature Pyramid Network), and its fusion process is defined as shown in Formula (2) [9,10]. After direct fusion, enhanced features are formed.

$$\begin{aligned} F_l^{\text{up}} &= \text{UpSample}(F_{l+1}) + F_l, \quad l = 1, 2, \\ F_l^{\text{down}} &= \text{DownSample}(F_l^{\text{up}}) + F_l, \quad l = 2, 3. \end{aligned} \quad (2)$$

In Formula (2), UpSample represents the nearest neighbor or bilinear upsampling operation, and DownSample represents the 2×2 maximum pooling.

The detection head of YOLOv8s adopts a decoupling strategy to separate the classification task and the regression task, and outputs tensors including the class probability matrix, bounding box prediction, and confidence.

CBAM-MOBILENETV3 MODEL

MobileNetV3

In this paper, the MobileNet series is often used as a lightweight network to speed up inference [11,12]. This paper uses MobileNetV3 as the core lightweight feature extraction network to replace the YOLOv8s backbone. For the input image, the initial feature map is generated by standard convolution, and then enters the depth-separable convolution module.

Depthwise convolution operation is shown in Formula (3).

$$Y_{i,j,c}^{\mathrm{dw}} = \sum_{m=0}^{k-1}\sum_{n=0}^{k-1} \hat{W}_{m,n,c}^{\mathrm{dw}} \cdot X_{i+m,j+n,c} + b_c^{\mathrm{dw}} \quad (3)$$

In Formula (3), c represents the channel index, and $\hat{W}_{m,n,c}^{\mathrm{dw}}$ represents the independent convolution kernel weight for each channel.

Pointwise convolution (1×1 convolution) is shown in Formula (4) [13,14].

$$Y_{i,j,c'}^{\mathrm{pw}} = \sum_{c=0}^{C_{\mathrm{in}}-1} \hat{W}_{c,c'}^{\mathrm{pw}} \cdot Y_{i,j,c}^{\mathrm{dw}} + b_{c'}^{\mathrm{pw}} \quad (4)$$

In Formula (4), $Y_{i,j,c'}^{\mathrm{pw}}$ represents the output of pointwise convolution. MobileNetV3 uses Hard-Swish instead of ReLU (Rectified Linear Unit) in nonlinear activation to speed up inference. Hard-Swish is shown in Formula (5).

$$\begin{aligned} \mathrm{HSwish}(x) &= x \cdot \frac{\mathrm{ReLU6}(x+3)}{6}, \\ \mathrm{ReLU6}(x) &= \min(\max(0,x),6). \end{aligned} \quad (5)$$

In Formula (5), x represents the convolution output.

To enhance the channel feature selection capability, a Squeeze-and-Excitation (SE) module is added after each residual module of MobileNetV3. In the Squeeze stage, global average pooling is used to generate channel descriptors, as shown in Formula (6).

$$z_{c'} = \frac{1}{H \times W} \sum_{i=1}^{H}\sum_{j=1}^{W} Y_{i,j,c'}^{\mathrm{pw}} \quad (6)$$

In Formula (6), $z_{c'}$ represents the channel descriptor.

In the Excitation stage, the channel weights are mapped through two layers of full connections, as shown in Formula (7).

$$s = \sigma\left(\hat{W}_2 \cdot \delta\left(\hat{W}_1 \cdot z\right)\right) \quad (7)$$

In Formula (7), δ represents ReLU activation, and σ represents Sigmoid function. $\hat{W}_1$ and $\hat{W}_2$ represent the corresponding weights, and s represents the channel weight.

The output channel weights are used to reweight the channel features, as shown in Formula (8).

$$\tilde{Y}_{i,j,c'}^{\mathrm{pw}} = s_{c'} \cdot Y_{i,j,c'}^{\mathrm{pw}} \quad (8)$$

MobileNetV3 adopts an inverted residual structure, as shown in Formula (9). First, the channel C_{in} is expanded through a 1×1 convolution, then passes through the Depthwise + HSwish + SE modules, and is finally compressed back to the channel C_{out} through a 1×1 convolution and skip-connected with the input.

$$X_{\mathrm{out}} = X_{\mathrm{in}} + \tilde{Y}, \quad \text{if } C_{\mathrm{in}} = C_{\mathrm{out}} \quad (9)$$

In Formula (9), X_{in} and X_{out} represent the feature map input to the residual block and the final output, respectively.

In YOLOv8s fusion, MobileNetV3 selects three scale features as output $F_l = \tilde{Y}^{(l)}$, $l = 1,2,3$, corresponding to low-level high resolution, enhancing small object details, mid-level semantics and detail balance, and high-level low resolution, enhancing global semantics and occlusion reasoning.

CBAM

In this paper, to enhance the spatial and channel selection capabilities of MobileNetV3 feature extraction and improve the detection accuracy of small objects and occluded objects, CBAM is used as a feature enhancement module [15,16].

This study embeds CBAM modules at the output nodes of the three feature pyramids in the MobileNetV3 backbone network, located after the last inverted residual block of the output 1/4, 1/8, and 1/16 resolution feature maps, respectively. This multi-level attention mechanism design enables the model to adaptively enhance key features at different scales, especially for small targets and occluded regions, while avoiding excessive computational overhead in early layers, ensuring a balance between lightweight characteristics and detection accuracy.

1) Channel attention: The global average pooling operation is shown in Formula (10) [17].

$$f_c^{\mathrm{avg}} = \frac{1}{H \times W} \sum_{i=1}^{H}\sum_{j=1}^{W} F_{i,j,c} \quad (10)$$

In Formula (10), H represents the feature map height; W represents the width; f_c^{avg} represents the global pooled feature.

The global maximum pooling operation is shown in Formula (11).

$$f_c^{\max} = \max_{i=1,\ldots,H,\ j=1,\ldots,W} F_{i,j,c} \quad (11)$$

In Formula (11), $f_c^{\max}$ represents the maximum pooled feature.

After global and maximum pooling, a shared MLP (Multilayer Perceptron) mapping is used, as shown in Formula (12).

$$M_c = \sigma\left(\hat{W}_4 \cdot \delta\left(\hat{W}_3 \cdot f^{\mathrm{avg}}\right) + \hat{W}_4 \cdot \delta\left(\hat{W}_3 \cdot f^{\max}\right)\right) \quad (12)$$

In Formula (12), $\hat{W}_3$ and $\hat{W}_4$ represent the fully connected weight matrix, and M_c represents the channel attention weight.

The channel features are weighted using M_c, as shown in Formula (13).

$$F_c' = M_c \odot F \quad (13)$$

In Formula (13), $\odot$ represents the channel dimension element-wise multiplication, and F_c' represents the features after channel attention weighting.

2) Spatial attention: After channel attention enhancement, the spatial attention weight is further calculated. The channel compression operation is shown in Formula (14) [18,19].

$$F_s^{\text{avg}}(i,j) = \frac{1}{C}\sum_{c=1}^{C} F'_{i,j,c}, \\ F_s^{\max}(i,j) = \max_{c=1,\ldots,C} F'_{i,j,c}. \tag{14}$$

In Formula (14), $F_s^{\text{avg}}(i,j)$ and $F_s^{\max}(i,j)$ represent the features after channel compression operation. The convolution mapping generates spatial weights, as shown in Formula (15).

$$M_s = \sigma\left(f^{7\times7}\left([F_s^{\text{avg}}, F_s^{\max}]\right)\right) \tag{15}$$

In Formula (15), M_s represents the spatial attention weight, and $f^{7\times7}$ represents the convolution layer with a convolution kernel size of 7×7.

The final spatially weighted features are shown in Formula (16).

$$F'' = M_s \odot F' \tag{16}$$

In Formula (16), F'' represents the spatially weighted features.

The CBAM visualization results are shown in Figure 1.

Figure 1 shows the comparison of the visualization effect of the original image and the one after the application of the CBAM attention mechanism. (a) is the input original image, which has not been processed by feature extraction and attention enhancement, and only presents the basic content of the scene as a whole; (b) is the CBAM attention feature map, where multiple highlighted areas can be clearly seen, corresponding to cars and people. These areas correspond to the target positions that the model pays more attention to during the feature extraction process. The comparison shows that CBAM effectively highlights the key target areas and suppresses background interference, providing a clearer and more discriminative feature expression for subsequent object recognition.

YOLOV8S NECK AND DETECTION HEAD ADJUSTMENT

YOLOv8s Neck Adjustment

To improve the V2X environment, this paper makes targeted adjustments to the YOLOv8s neck and detection head. The YOLOv8s native neck adopts the PAN-FPN architecture. This study applies weighted bidirectional feature fusion based on it. The adaptive upsampling fusion and downsampling fusion are shown in Formula (17).

$$F_l^{\text{up}} = \alpha_l \cdot \text{UpSample}(F_{l+1}) + (1-\alpha_l)\cdot F_l, \\ F_l^{\text{down}} = \beta_l \cdot \text{DownSample}(F_{l-1}^{\text{up}}) + (1-\beta_l)\cdot F_l^{\text{up}}. \tag{17}$$

In Formula (17), α_l and β_l represent the learnable upsampling and down-sampling weights, respectively; UpSample is bilinear interpolation; DownSample is 2×2 maximum pooling.

The fusion output is shown in Formula (18).

$$F_l^{\text{neck}} = \gamma_l \cdot F_l^{\text{up}} + (1-\gamma_l)\cdot F_l^{\text{down}}, \quad \gamma_l \in [0,1] \tag{18}$$

In Formula (18), γ_l represents the comprehensive fusion weight, and Flneck represents the fusion output.

YOLOv8s Detection Head Adjustment

In the detection head, the bounding box regression uses SIOU to replace CIOU. SIOU takes into account IoU, center offset, and aspect ratio, and is more suitable for small object offset sensitivity and occluded object positioning. The detection head receives three scale features F_1^{neck}, F_2^{neck}, and F_3^{neck} of the neck. The detection head of YOLOv8 adopts an anchor-free decoupled design. Predictions at different scales correspond to feature maps with different strides, which implicitly favor objects of different sizes through dynamic label assignment rather than predefined anchor boxes. The detection head uses 1×1 convolution for dimensionality reduction and lightweight decoupled convolution to reduce inference FLOPs (Floating Point Operations). All parameters are exported and optimized using TensorRT (Tensor Runtime) to meet the real-time deployment requirements of V2X environments.

The image object recognition framework is shown in Figure 2.

In Figure 2, the input image first extracts multi-scale features through the CBAM-MobileNetV3 backbone network, in which each MobileNetV3 block cooperates with the CBAM attention module to enhance the feature response of the key area. The multi-scale feature map output by the backbone is sent to the YOLOv8s neck fusion network, which generates a fused multi-scale feature map through upsampling, downsampling, and weighted feature fusion, and then enters the three branches of the decoupled detection head: classification, bounding box regression, and confidence prediction. The final detection head outputs the category probability, bounding box, and confidence.

LOSS FUNCTION DESIGN AND MODEL TRAINING OPTIMIZATION

To improve the detection accuracy of small objects and occluded objects in the V2X environment while taking into account the low latency requirements of edge devices, this paper optimizes the loss function of the YOLOv8s detection framework and combines it with the training strategy to achieve efficient convergence.

LOSS FUNCTION DESIGN

The total loss function consists of three parts: bounding box regression loss, classification loss, and confidence loss, as shown in Formula (19).

$$L_{\text{total}} = \lambda_1 L_{\text{SIoU}} + \lambda_2 L_{\text{cls}} + \lambda_3 L_{\text{obj}} \tag{19}$$

FIGURE 1. CBAM visualization results. Figure 1(a) Original image; Figure 1(b) CBAM attention feature map

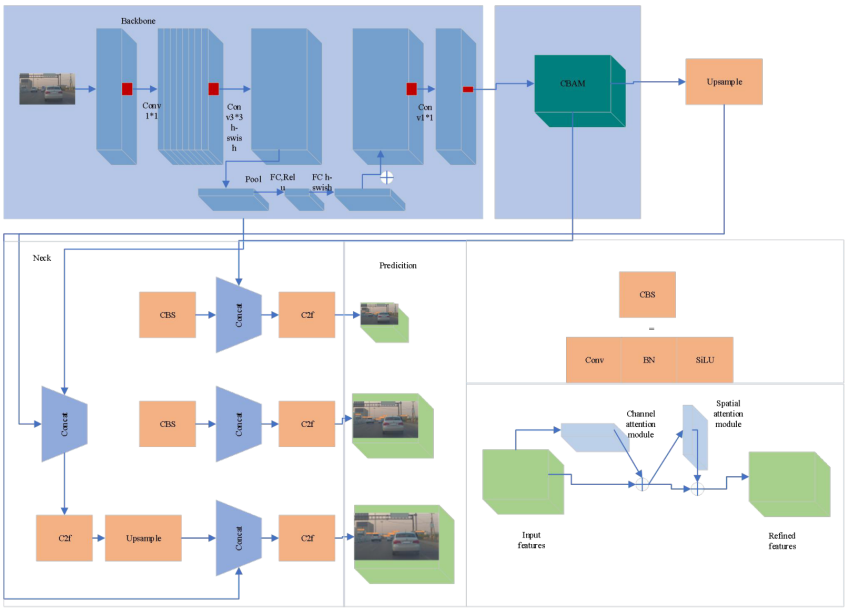

FIGURE 2. Image object recognition framework

In Formula (19), λ_1, λ_2, and λ_3 represent the loss weights of each branch. L_{SIoU} represents the bounding box regression loss; L_{cls} represents the classification loss; L_{obj} represents the confidence loss. Compared to the traditional CIOU, SIOU imposes finer-grained penalties on center offset, aspect ratio, and orientation differences during the regression phase. This allows the predicted bounding box to converge more quickly to a solution consistent with the true target shape and location when the target is only partially visible or has sparse boundary information. This regression constraint complements the high-resolution local features and spatial attention generated by the MobileNetV3-CBAM backbone: CBAM amplifies the local response of occluded/sparse regions, while MobileNetV3 preserves fine-grained edge information, providing SIOU with more reliable geometric cues. Conversely, SIOU's strict geometric penalty makes regression based on these weakly saliency features more robust, reducing center offset and aspect ratio drift. The synergy of these two approaches improves the localization accuracy of small and partially visible targets while reducing false detections and localization errors, maintaining lightweight design and real-time performance.

The formula for the bounding box regression loss is:

$$L_{\mathrm{SIoU}} = 1 - \mathrm{IoU}(B_{\mathrm{pred}}, B_{\mathrm{gt}}) + \frac{\rho^2(\vec{\iota}_{\mathrm{pred}}, \vec{\iota}_{\mathrm{gt}})}{\iota^2} + \varsigma \cdot \upsilon + \pi \cdot \theta \tag{20}$$

In Formula (20), $\varsigma = \frac{\upsilon}{1-\mathrm{IoU}+\upsilon}$, and π represents the direction weight, with a value of 0.5. $\rho^2(\vec{\iota}_{\mathrm{pred}}, \vec{\iota}_{\mathrm{gt}})$ represents the center offset distance; ι represents the minimum diagonal length of the enclosed rectangle between the two boxes; υ represents the aspect ratio penalty; θ represents the direction penalty.

The classification loss uses Focal Loss to balance the positive and negative sample imbalance problem, as shown in Formula (21).

$$L_{\mathrm{cls}} = -\rho_1 \sum_{k=1}^{K} (1 - p_k)^{\rho_2} y_k \log(p_k) \tag{21}$$

In Formula (21), y_k represents the true category label, and p_k represents the predicted probability. ρ_1 and ρ_2 are 0.25 and 2.0, respectively, to adjust the sample attention.

The confidence uses BCE (Binary Cross Entropy), as shown in Formula (22).

$$L_{\text{obj}} = -\sum_{i=1}^{N} [y_i \log(o_i) + (1 - y_i) \log(1 - o_i)] \quad (22)$$

In Formula (22), y_i indicates the target label, and o_i indicates the prediction confidence.

MODEL TRAINING OPTIMIZATION

For learning rate scheduling, the cosine annealing strategy is adopted, and the convolution weight is constrained by L2 regularization, as shown in Formula (23).

$$\begin{aligned} \eta_t &= \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min}) \left(1 + \cos \frac{t\pi}{T}\right), \\ L_{\text{reg}} &= \lambda_{\text{reg}} \sum_i \left\| \hat{W}_i \right\|_2^2. \end{aligned} \quad (23)$$

In Formula (23), η_t indicates the learning rate; T indicates the total number of iterations; L_{reg} indicates the regularization loss; λ_{reg} indicates the regularization weight.

The experiment is based on the cosine annealing strategy and combined with the AdamW optimizer to improve the convergence speed while taking into account regularization [20,21].

IV. EVALUATION AND DATA PLAN

EXPERIMENTAL DATA AND PREPROCESSING

The experimental data for this paper is derived from the DAIR-V2X-C dataset, collected from 10 kilometers of urban roads, 10 kilometers of highways, and 28 intersections in Beijing's high-level autonomous driving demonstration zone, including both 2D and 3D annotations. The data covers a wide range of scenarios, including sunny/rainy/foggy conditions, daytime/nighttime conditions, and urban/highway scenarios. The dataset consists of 40,481 frames of image data and 40,481 frames of point cloud data. In the preprocessing stage, the images and point clouds are first spatiotemporally aligned and calibrated. The images are then resized to 640×640 and normalized (using mean (0.485, 0.456, 0.406) and variance (0.229, 0.224, 0.225)) to reduce interference from illumination and sensor noise. The training, validation, and test sets are split into a 70:20:10 ratio. The test set contains 1,100 images of cars, 600 of trucks, 420 of vans, 180 of buses, 850 of pedestrians, 380 of bicycles, 180 of tricycles, 220 of motorcycles, 68 of carts, and 50 of traffic cones. To enhance the robustness of small object detection under occlusion conditions, this paper designs multiple data augmentation strategies: 1 random scale scaling, which scales the input image within the range of [0.5, 1.5] and maintains the corresponding scale of the annotation box to enhance the multi-scale adaptability of the model; 2 random cropping and translation, which crops or translates the image while ensuring the integrity of the target to simulate partial occlusion and perspective shift; 3 color perturbation, which randomly adjusts the brightness, contrast, and saturation within the range of ±20% to enhance adaptability to lighting and weather changes; 4 mosaic stitching enhancement, which randomly stitches four images into a single image and adjusts the position of its annotation box, significantly increasing the frequency of small objects in complex backgrounds.

EVALUATION METRICS

The formula for precision is:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (24)$$

In Formula (24), TP (True Positive) represents the number of correctly detected targets, and FP (False Positive) represents the number of false positives.

The formula for recall is:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (25)$$

In Formula (25), FN (False Negative) represents the number of false negatives.

F1-score is the harmonic mean of precision and recall, as shown in Formula (26).

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (26)$$

The formula for mAP is:

$$\begin{aligned} AP &= \int_0^1 P(R)\, dR, \\ mAP &= \frac{1}{\psi} \sum_{\psi=1}^{\psi} AP_{\psi}. \end{aligned} \quad (27)$$

In Formula (27), ψ represents the total number of categories.

EXPERIMENTAL PROCESS DESIGN

The experiment is designed according to a reproducible control process to verify the improvement effect of YoloV8s (MobileNetV3-CBAM) on small objects and occluded objects in the V2X environment. All comparison models are trained under the same data partition (training/validation/testing) and a unified preprocessing/enhancement pipeline to ensure that the input distribution and label format are consistent. For each model, mAP@0.5 (%), mAP@[0.5:0.95] (%), precision (%), recall (%), F1 (%), and deployment metrics (FPS, model size, FLOPs, energy consumption, etc.) are calculated on the test set. Scene sensitivity is assessed by analyzing these metrics based on object scale (small/medium/large), occlusion level, and weather/time of day (sunny/rainy/foggy, daytime/nighttime, urban roads, highways). To quantify the uncertainty of the mAP index, this study used the bootstrap method (image-level resampling) to resample the test set 1000 times and calculated the 95% confidence intervals of mAP@0.5 and mAP@[0.5:0.95] for each model.

To avoid introducing additional bias due to differences in training strategies, this paper adopts a "fair but not overly uniform" principle for the training configuration of each comparison model, while ensuring consistency in data partitioning and preprocessing. For each model, the optimizer type, learning rate scheduling strategy, weight decay, and loss function settings recommended in its official code are prioritized, and only minimal adjustments necessary for the input resolution and detection task are made to ensure that the model performance is at a reasonable and reproducible level. During the inference phase, all models are evaluated on the same hardware platform (same GPU, CUDA, and inference framework version), with a uniform batch size of 1 to eliminate the impact of differences in parallelism on latency and energy consumption. Furthermore, the training and deployment phases are clearly distinguished, and only general and publicly available inference optimizations (such as operator fusion and FP16 inference) are used, without introducing specific acceleration strategies for a single model. This ensures fairness and comparability in the accuracy and deployment efficiency comparisons of different methods.

The comparison models include: SSD-Lite, YOLOv5 ((You Only Look Once version 5) (EfficientNet-Lite-CBAM), YOLOv7-tiny, YOLOv8 (MBConv-C3FB-BCFPN-CBAM), YOLOv10 (BiFPN-DETR), and YoloV8s (MobileNetV3-CBAM) in this paper. The experiments also include module-level ablation experiments. Gaussian noise is added to verify module robustness, generalization performance across different datasets, and performance with different attention mechanisms and lightweight feature extraction modules. Hyperparameters are shown in Table 1.

TABLE 1. Hyperparameters

Parameters	Value	Parameters	Value
Training epochs	100	CBAM compression ratio	4
Training batch size	16	CBAM spatial convolution kernel	7×7
Gradient accumulation	2	Bounding box regression loss weight	5
Random seed	42	Classification branch weight	1
Input resolution	640 × 640	Confidence branch weight	1
Initial learning rate	1*10-3	Confidence threshold	0.25
Minimum learning rate η	1*10-6	Non-maximum suppression overlap threshold	0.45
L2 regularization coefficient	5.00E-04	Evaluation period	1 epoch
Optimizer parameters	(0.9, 0.999)	Early stopping patience value	10

V. OBJECT RECOGNITION RESULTS IN V2X ENVIRONMENTS

VISUALIZATION OF OBJECT RECOGNITION DETECTION RESULTS

The visualization of object recognition detection results is shown in Figure 3.

Figure 3 demonstrates the detection performance of the YOLOv8s (MobileNetV3-CBAM) model for multiple object classes in a V2X environment. The visualization demonstrates that the model accurately recognizes multiple object classes, including cars, trucks, pedestrians, and bicycles. Even when some objects are occluded or at a distance, the detection bounding boxes maintain good coverage of the target area. Furthermore, for trucks and prominent foreground objects, the model-generated 2D detection bounding boxes closely match the ground-truth locations, demonstrating that the backbone network and CBAM attention mechanism effectively enhance feature representation and improve detection accuracy for small and occluded objects. The detection bounding boxes are clearly colored and accurately labeled, with confidence scores generally above 0.95, demonstrating the model's robustness and stability across multiple object classes, complex backgrounds, and varying lighting conditions.

CONFUSION MATRIX AND OVERALL DETECTION PERFORMANCE

To explore the accuracy of each category, the confusion matrix is analyzed, as illustrated in Figure 4.

In Figure 4, YoloV8s (MobileNetV3-CBAM) performs best in the confusion matrix, achieving diagonal accuracy of 98.0% for Car, 96.0% for Truck, 96.9% for Van, 95.0% for Bus, 98.9% for Pedestrian, 96.8% for Bicycle, 95.0% for Tricycle, 96.8% for Motorbike, 92.6% for Trolley, and 92.0% for Cone. The distribution of off-diagonal errors shows generally low misclassification rates (most off-diagonal entries are $\leq$2%). For example, only 0.5% of Cars are misclassified as Trucks, and 0.4% of Pedestrians are misclassified as Cars. The underlying reason for these differences is that SSD-Lite loses fine-grained information due to its basic single-level features and lightweighting, making it difficult to distinguish small objects (trolleys, cones) and occluded objects. In contrast, MobileNetV3 preserves high-resolution features in low layers while maintaining information flow through an inverted residual structure. Combined with its multi-scale output, this provides more usable responses for small objects during neck fusion. Class-specific confusion (such as between tricycles/motorcycles or cars/vans) is also affected by visual similarity, occlusion, and sample imbalance. The proposed method significantly mitigates these factors through attention enhancement and multi-scale training, reducing off-diagonal error to below 2% for most classes.

The overall detection performance results are shown in Figure 5.

FIGURE 3. Visualization of object recognition detection results. Figure 3(a) Daytime multi-target detection results for urban road 1. Figure 3(b) Daytime multi-target detection results for urban road 2. Figure 3(c) Daytime multi-target detection results for urban road 3. Figure 3(d) Daytime multi-target detection results for urban road 4.

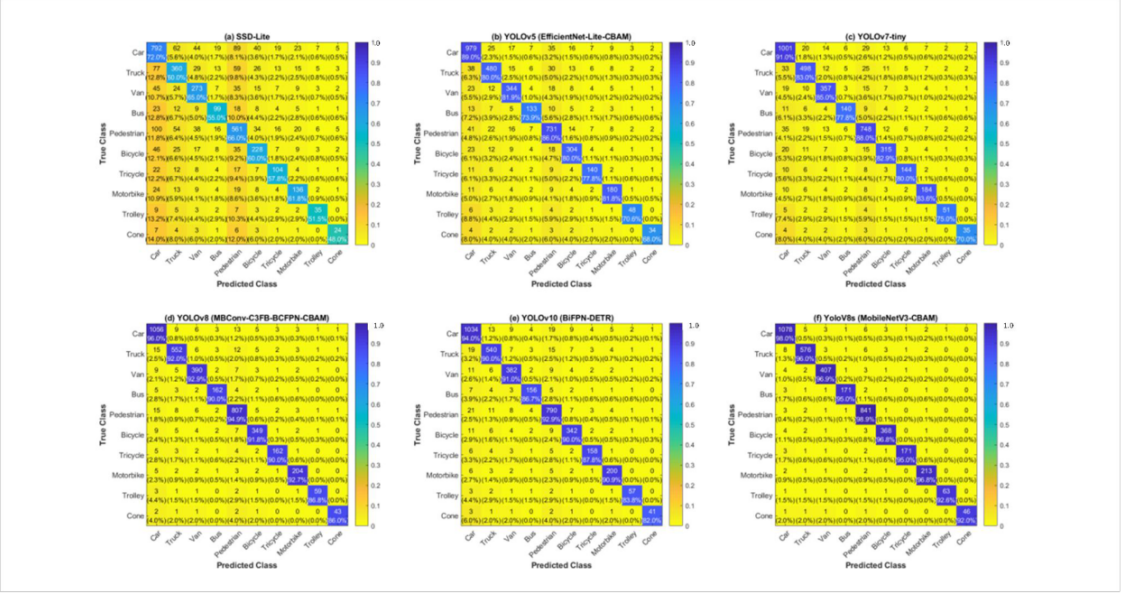

FIGURE 4. Confusion matrix. Figure 4(a) SSD-Lite; Figure 4(b) YOLOv5 (EfficientNet-Lite-CBAM); Figure 4(c) YOLOv7-tiny; Figure 4(d) YOLOv8(MBConv-C3FB-BCFPN-CBAM); Figure 4(e) YOLOv10 (BiFPN-DETR); Figure 4(f) YoloV8s (MobileNetV3-CBAM)

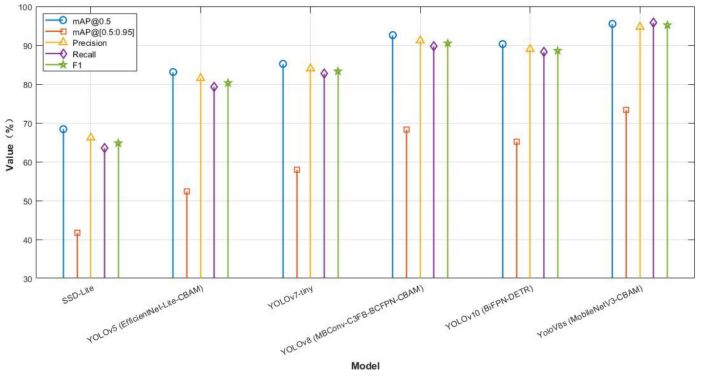

FIGURE 5. Overall detection performance

In Figure 5, YoloV8s (MobileNetV3-CBAM) achieves the highest mAP performance in both metrics, with mAP@0.5 of 95.5% and mAP@[0.5:0.95] of 73.4%. YOLOv8 (MBConv-C3FB-BCFPN-CBAM) achieves 92.6% and 68.3%, respectively, representing decreases of 2.9% (mAP@0.5) and 5.1% (mAP@[0.5:0.95]) compared to YoloV8s (MobileNetV3-CBAM). The YOLOv10 (BiFPN-DETR) is 90.3%/65.1%, which is a decrease of 5.2%/8.3% compared to YoloV8s (MobileNetV3-CBAM); YOLOv7-tiny, YOLOv5 (EfficientNet-Lite-CBAM), and SSD-Lite are 85.2%/57.9%, 83.1%/52.4%, and 68.4%/41.7%, respectively. As the IoU (Intersection over Union) threshold increases from 0.5 to [0.5:0.95], the gap between models generally widens. YoloV8s (MobileNetV3-CBAM) achieves a 31.7% improvement over SSD-Lite on the strict mAP@[0.5:0.95], demonstrating that the proposed method has significant advantages in bounding box localization and consistency under higher IoU requirements. However, intermediate models (such as YOLOv7-tiny and YOLOv5) experience significant performance degradation under high IoU conditions, indicating insufficient localization accuracy or robustness to small/occluded objects.

YoloV8s (MobileNetV3-CBAM) also leads in precision, recall, and F1, with Precision of 94.7%, Recall of 95.8%, and F1 of 95.2%, demonstrating that the model effectively suppresses false positives while minimizing false negatives. Compared to YOLOv8 (MBConv-C3FB-BCFPN-CBAM) (91.2%/89.8%/90.5%), YoloV8s achieves 3.5% higher Precision, 6.0% higher Recall, and 4.7% higher F1. YOLOv10 (BiFPN-DETR) (89.0%/88.2%/88.6%) is similar to, but slightly lower than, YOLOv8 (MBConv-C3FB-BCFPN-CBAM). YOLOv7-tiny (84.0%/82.7%/83.3%) and YOLOv5 (81.5%/79.2%/80.3%) are in the middle, while SSD-Lite is the lowest (66.2%/63.5%/64.8%).

DETECTION PERFORMANCE AT DIFFERENT OCCLUSION LEVELS

This paper studies the changes in detection performance at different occlusion levels to evaluate the differences in the robustness of various models to occlusion. The results are shown in Figure 6. This paper defines occlusion degree as the ratio of the area of the actual visible region of a target in an image to the area of its original labeled bounding box. The visible region is obtained through manual verification and semi-automatic segmentation of target instances within the original labeled box, used to remove pixel regions occluded by other objects or scene structures. When the target is unoccluded, this ratio is 1.0, gradually decreasing as occlusion worsens. Based on this visibility ratio, the test samples are divided into five discrete occlusion levels: no occlusion (1.0), slight occlusion (0.8), moderate occlusion (0.6), severe occlusion (0.4), and extreme occlusion (0.2). Each level includes only samples with visibility ratios falling within the corresponding range, thus allowing for grouped evaluation of the detection performance of each model under different occlusion conditions, ensuring that occlusion analysis has clear measurement criteria and repeatability.

Under the unoccluded condition with an occlusion level of 1.0, all models perform best, but there are still obvious hierarchies. YoloV8s (MobileNetV3-CBAM) has an mAP@0.5 of 98.2%, an mAP@[0.5:0.95] of 82.1%, and an F1 of 98.0%, leading the group; YOLOv8 (MBConv-C3FB-BCFPN-CBAM) reaches 96.1%/78.3%/95.4%, respectively, and YOLOv10 (BiFPN-DETR) reaches 94.8%/76.9%/93.8%, respectively; the medium performance is YOLOv7-tiny (92.0%/68.1%/90.6%) and YOLOv5 (EfficientNet-Lite-CBAM)

(90.5%/65.8%/88.7%); the lowest is SSD-Lite (78.6%/52.3%/75.4%). As can be seen, with complete visual information, the strong feature extraction + attention + multi-scale fusion solution has a clear advantage in strict localization (mAP@[0.5:0.95]) and classification/recall balance (F1).

The performance decreases gradually with the occlusion level from 0.8 to 0.4. At 0.8, YoloV8s is 97.4%/80.6%/97.1%, while SSD-Lite has dropped to 74.1%/48.2%/71.8%. At 0.6, YoloV8s drops to 93.6%/68.9%/92.8%; YOLOv8 (MBConv-C3FB-BCFPN-CBAM) is 90.7%/63.8%/89.9%; YOLOv10 (BiFPN-DETR) is 88.9%/61.5%/88.0%. The mid- and low-end models (YOLOv7-tiny, YOLOv5 (EfficientNet-Lite-CBAM), and SSD-Lite) show a steeper decline in this range. At an occlusion level of 0.4, YoloV8s (MobileNetV3-CBAM)

maintains 88.5%/58.2%/88.9%, approximately 4.6% higher than its comparable YOLOv8 (MBConv-C3FB-BCFPN-CBAM) (mAP@0.5). This demonstrates that YoloV8s is more robust in maintaining a balance between recall and precision under moderate occlusion conditions, particularly for small/partially occluded objects, where it retains more feature information.

When the occlusion level drops to 0.2, overall performance drops significantly, but YoloV8s still maintains its advantage with mAP@0.5 of 80.4%, mAP@[0.5:0.95] of 50.7%, and F1 of 80.2%. Compared with the weakest SSD-Lite (39.7%/18.6%/36.8%), YoloV8s outperforms SSD-Lite by 40.7% in mAP@0.5 and by 32.1% in mAP@[0.5:0.95]. The intermediate models YOLOv8 (MBConv-C3FB-BCFPN-CBAM) and YOLOv10 (BiFPN-DETR) achieve performances of 72.2%/42.0%/71.0% and 69.8%/39.0%/68.5% under severe occlusion, respectively.

This paper compares the performance of different models by considering the overlap of confidence intervals and the effect size index. If the confidence intervals do not overlap, the model can be considered to have a significant performance advantage at that occlusion level. Table 2 shows the mAP@0.5 (%) and 95% confidence intervals for each model under the extreme occlusion condition (0.2).

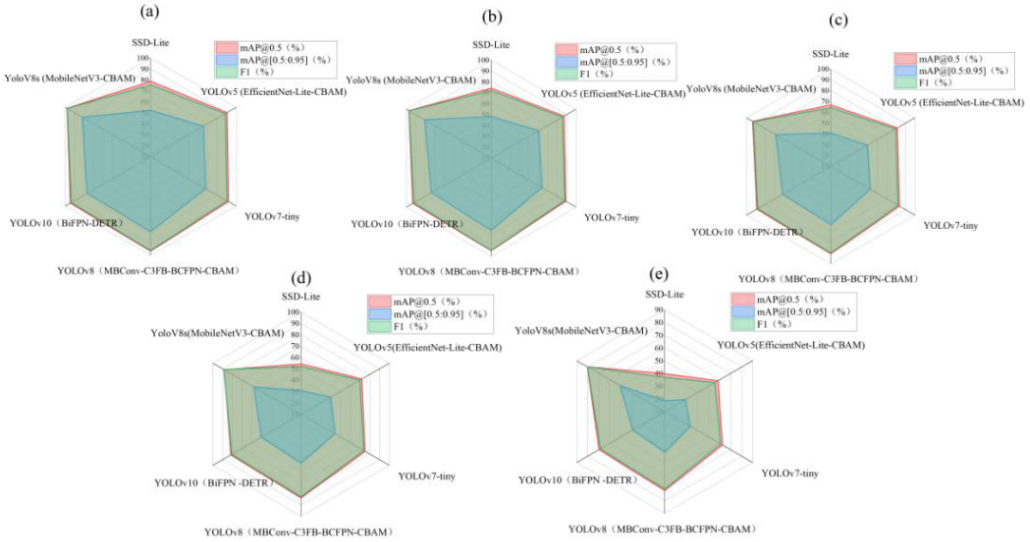

FIGURE 6. Detection performance at different occlusion levels. Figure 6(a) Occlusion level (1.0); Figure 6(b) Occlusion level 0.8; Figure 6(c) Occlusion level 0.6; Figure 6(d) Occlusion level 0.4; Figure 6(e) Occlusion level 0.2

TABLE 2. mAP@0.5 (%) and 95% confidence intervals for each model under the extreme occlusion condition (0.2)

Comparison model	mAP@0.5 (%)	95% CI (%)	mAP@[0.5:0.95] (%)	95% CI (%)
SSD-Lite	39.7	[36.5, 42.8]	18.6	[16.0, 21.2]
YOLOv5 (EfficientNet-Lite-CBAM)	58.3	[55.1, 61.2]	28.9	[26.0, 31.8]
YOLOv7-tiny	62.5	[59.6, 65.3]	33.4	[30.6, 36.2]
YOLOv8 (MBConv-C3FB-BCFPN-CBAM)	72.2	[69.8, 74.5]	42.0	[39.3, 44.7]
YOLOv10 (BiFPN-DETR)	69.8	[67.3, 72.2]	39.0	[36.3, 41.7]
YoloV8s (MobileNetV3-CBAM)	80.4	[78.2, 82.5]	50.7	[48.1, 53.3]

Note: The 95% CI is an approximate confidence interval based on the Bootstrap estimate, used to reflect the uncertainty of the mAP metric.

As shown in Table 2, under the extreme occlusion condition (0.2), YOLOV8s (MobileNetV3-CBAM) achieves an mAP@0.5 of 80.4%, with a 95% confidence interval of [78.2, 82.5], significantly higher than all the comparison models. The closest performing model is YOLOv8 (MBConv-C3FB-BCFPN-CBAM) at 72.2% [69.8, 74.5]. The confidence intervals of the two models do not overlap, indicating that YOLOV8s has a significant advantage under extreme occlusion conditions. The mid-range models YOLOv10 (BiFPN-DETR) and YOLOv7-tiny achieved 69.8% [67.3, 72.2] and 62.5% [59.6, 65.3] respectively, while SSD-Lite had the lowest performance at only 39.7% [36.5, 42.8]. In terms of mAP@[0.5:0.95], YoloV8s also performed best at 50.7% [48.1, 53.3], leading YOLOv8 (MBConv-C3FB-BCFPN-CBAM) by more than 8.7 percentage points at 42.0% [39.3, 44.7]. Overall, as the model's capabilities decrease, the confidence interval gradually shifts downward and does not overlap with YoloV8s (MobileNetV3-CBAM), indicating that the model can significantly maintain its robustness and advantages in detection performance when dealing with small targets and severely occluded objects.

SCALE SENSITIVITY RESULTS

The experiment evaluates the detection performance of each model at different object sizes (small, medium, and large) to quantify the impact of scale on recognition accuracy and verify the advantage of YoloV8s (MobileNetV3-CBAM) in small object scenarios. Small, medium, and large objects are grouped according to their bounding box pixel area (width×height) in the input image (640×640). Small objects are classified as having less than 1024 pixels, medium objects as having 1024-9216 pixels, and large objects as having 9216 pixels or more. The scale sensitivity results are shown in Figure 7.

For small objects, the difference between models is most significant. YoloV8s (MobileNetV3-CBAM) achieves mAP@0.5 of 88.1%, mAP@[0.5:0.95] of 59.5%, and F1 of 88.4%, respectively, which are 6.8%/6.3%/8.3% higher than YOLOv8 (MBConv-C3FB-BCFPN-CBAM) (81.3%/53.2%/80.1%). YoloV8s

(MobileNetV3-CBAM) also has advantages over YOLOv10 (BiFPN-DETR) (78.9%/50.7%/78.0%) by 9.2%/8.8%/10.4%. Mid-range lightweight models such as YOLOv7-tiny (68.5%/41.0%/67.0%) and YOLOv5 (EfficientNet-Lite-CBAM) (64.2%/36.8%/62.1%) all have worse mAP@0.5 on small objects than YoloV8s

(MobileNetV3-CBAM). The worst, SSD-Lite, achieves only 39.8%/22.1%/37.5%, a massive 48.3%/37.4%/50.9% difference from YoloV8s. This shows that YoloV8s (MobileNetV3-CBAM) leads significantly in small object scenarios, and its improvement in the strict IoU range

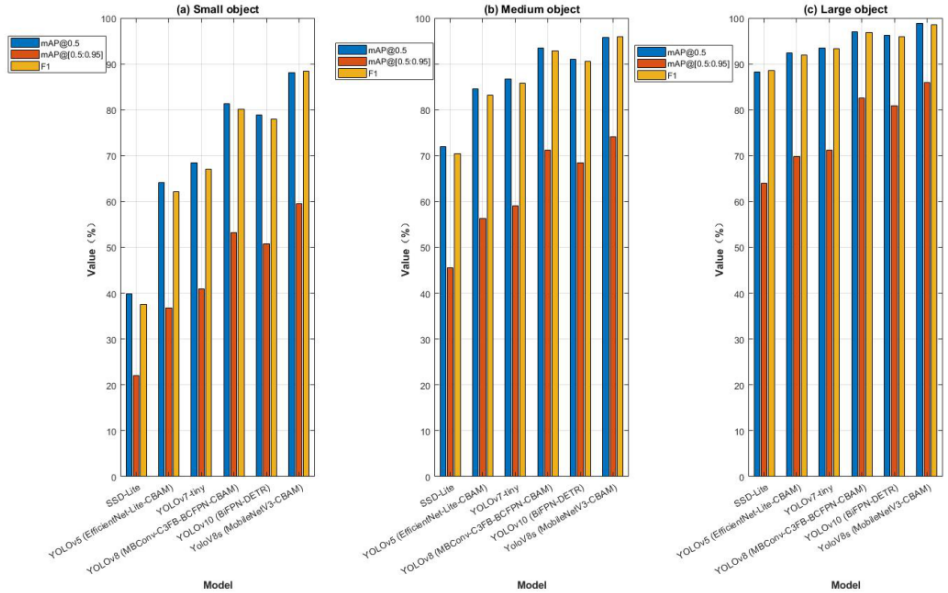

FIGURE 7. Scale sensitivity results. Figure 7(a) Small object; Figure 7(b) Medium object; Figure 7(c) Large object

(mAP@[0.5:0.95]) demonstrates that it can not only detect small objects but also more accurately localize them.

For medium and large objects, YoloV8s (MobileNetV3-CBAM) achieves 95.8%/74.1%/96.0% for medium

objects, compared to 93.4%/71.2%/92.9% for YOLOv8 (MBConv-C3FB-BCFPN-CBAM). YOLOv10 (BiFPN-DETR), YOLOv7-tiny, and YOLOv5 (EfficientNet-Lite-CBAM) also exceed the mAP@0.5 level of 84%. All models approach saturation for large objects, with YoloV8s achieving 98.8%/86.0%/98.6%, and YOLOv8 (MBConv-C3FB-BCFPN-CBAM) achieving 97.0%/82.5%/96.8%. Even the lightweight SSD-Lite

(88.2%/64.0%/88.5%) maintains high detection rates.

The experiment uses a paired t-test to compare the significance of mAP@0.5 (%) between the proposed model and the comparison models under the condition of small objects (area < 1024 px^2). The test results are shown in Table 3.

The results of the significance t-test (small objects) are displayed in Table 3.

TABLE 3. Significance t-test (small objects)

Comparison model	Present model mean ± std (%)	Comparison model mean ± std (%)	t-value	p-value	95% CI (difference, %)	Effect size (cohen's d)
SSD-Lite	88.1 ± 2.3	39.8 ± 3.9	58.8	< 0.0001	[46.6, 50.0]	10.7
YOLOv5 (EfficientNet-Lite-CBAM)	88.1 ± 2.3	64.2 ± 3.5	37.4	< 0.0001	[22.6, 25.2]	6.8
YOLOv7-tiny	88.1 ± 2.3	68.5 ± 3.2	33.6	< 0.0001	[18.4, 20.8]	6.1
YOLOv8 (MBConv-C3FB-BCFPN-CBAM)	88.1 ± 2.3	81.3 ± 2.1	17.8	< 0.0001	[6.0, 7.6]	3.2
YOLOv10 (BiFPN-DETR)	88.1 ± 2.3	78.9 ± 2.6	19.4	< 0.0001	[8.2, 10.1]	3.5

The paired t-test results in Table 3 show that under the small object condition, the mAP@0.5 differences between YoloV8s (MobileNetV3-CBAM) and all the comparison models are highly significant ($p < 0.0001$). Compared with the lightest SSD-Lite, the YoloV8s has a mean difference of 48.3%, a t-value of 58.8, and an

effect size Cohen's d of 10.7, showing a great advantage; the difference with YOLOv5 (EfficientNet-Lite-CBAM) is 23.9% (t=37.4, Cohen's d=6.8); the difference with YOLOv7-tiny is 19.6% (t=33.6, Cohen's d=6.1); the difference with YOLOv8 (MBConv-C3FB-CBAM) is 6.8% (t=17.8, Cohen's d=3.2); the difference with YOLOv10 (BiFPN-DETR) is 9.2% (t=19.4, Cohen's d=3.5).

RESOURCE CONSUMPTION

The operational efficiency and resource consumption of the model on the target edge platform are evaluated to measure the feasibility of deployment while meeting real-

time constraints. The results are displayed in Table 4. Table 4 compares the metrics including average inference latency, end-to-end response time, FLOPs, FPS, Params, and energy consumption.

In Table 4, the energy consumption data were all measured using the NVIDIA Jetson Xavier AGX edge computing platform. During the measurement process, the onboard power sensor was used to record the energy consumption during each frame of inference. The sampling window was the average of 1000 consecutive frames of inference to reduce the impact of occasional fluctuations. Baseline power consumption in the device's idle state was deducted during measurement; only the dynamic power consumption generated by the actual inference of the model was statistically analyzed. This more accurately reflects the energy consumption performance of each model on the edge platform, ensuring fairness and reproducibility in comparisons between different models.

TABLE 4. Operational efficiency and resource consumption

Model	Average Inference Latency (ms)	End-to-End Response Time (ms)	FLOPs/G	FPS (f/s)	Params (M)	Energy Consumption (J/frame)
SSD-Lite	9.2	14.5	2.3	108.7	3.1	0.9
YOLOv5 (EfficientNet-Lite-CBAM)	28.6	40.1	12.5	35.0	8.7	2.1
YOLOv7-tiny	18.9	26.4	8.9	52.9	6.3	1.4
YOLOv8 (MBConv-C3FB-BCFPN-CBAM)	46.8	62.2	25.7	21.4	18.4	3.5
YOLOv10 (BiFPN-DETR)	68.5	84.0	37.2	14.6	28.1	5.2
YoloV8s (MobileNetV3-CBAM)	13.5	20.3	6.4	74.1	4.9	1.1

The results in Table 4 show that YoloV8s (MobileNetV3-CBAM) achieves excellent operational efficiency and resource balance on the edge platform. Its average inference latency is 13.5 ms, and its end-to-end response time is 20.3 ms, which are much lower than the 46.8 ms and 62.2 ms of YOLOv8 (MBConv-C3FB-BCFPN-CBAM). In addition, it has 6.4G FLOPs, 4.9M parameters, and 1.1 J/frame energy consumption, which are significantly lower than the 37.2G FLOPs, 28.1M parameters, and 5.2 J/frame of YOLOv10 (BiFPN-DETR). The YoloV8s (MobileNetV3-CBAM) achieves an FPS of 74.1 f/s, significantly higher than the 21.4 f/s of YOLOv8 (MBConv-C3FB-BCFPN-CBAM) and 14.6 f/s of YOLOv10 (BiFPN-DETR), and close to the 108.7 f/s of the lightweight SSD-Lite.

DETECTION PERFORMANCE WITH DIFFERENT ATTENTION MECHANISM BLOCKS AND LIGHTWEIGHT FEATURE EXTRACTION MODULES

Based on the YoloV8s (MobileNetV3-CBAM) model, this paper examines the impact of replacing different attention mechanism blocks and lightweight feature extraction modules on detection performance and inference overhead. The results are shown in Figure 8. In Figure 8, the attention mechanism blocks include SE, SimAM, Shuffle, ECA (Efficient Channel Attention), and CBAM, and the lightweight feature extraction modules include EfficientNet-Lite (Efficient Network-Lite), ShuffleNetV2 (Shuffle Network Version 2), MobileNetV2 (Mobile Network version 2), MobileNetV3, and GhostNet (Ghost Network).

In Figure 8, comparing different attention mechanism blocks, CBAM achieves the best detection performance, with mAP@0.5 of 95.5% and mAP@[0.5:0.95] of 73.4%, slightly higher than SimAM's 94.6%/71.8% and ECA's 94.2%/71.0%, while maintaining a reasonable FLOPs of 6.4G and an average inference latency of 13.5 ms. In contrast, the Shuffle attention block achieves the lowest performance, with mAP@0.5 of 92.7% and mAP@[0.5:0.95] of 68.9%. Although its inference latency is slightly lower at 12.9 ms, its detection accuracy drops significantly. This shows that in multi-category complex object detection, CBAM can more effectively enhance feature expression and improve the recognition of small and occluded objects.

In a comparison of different lightweight feature extraction modules, the combination of MobileNetV3 and CBAM achieves the best performance, with mAP@0.5 of 95.5%, mAP@[0.5:0.95] of 73.4%, 6.4G of FLOPs, and an average inference latency of 13.5 ms. EfficientNet-Lite achieves 94.6% at mAP@0.5, but its FLOPs and inference latency are 7.4G and 17.8 ms, respectively, indicating a significant increase in resource consumption. Other lightweight modules, such as ShuffleNetV2 and GhostNet, show significant declines in accuracy (mAP@0.5 of 90.3% and 91.5%, respectively). While their latency is lower than MobileNetV3, their detection performance is insufficient to meet high reliability requirements in V2X environments. Furthermore, the average inference latency of the proposed model is similar to that of other feature extraction modules.

COMPARISON WITH RECENT STATE-OF-THE-ART (SOTA) METHODS

This paper presents comparative experiments with four representative and widely adopted strong baseline methods in recent years: YOLOv8-l (representative of the large-scale YOLO family), RTMDet-L (representative of a real-time optimized detector), DINO-Base (representative of the DETR series of improvements), and Swin-T + Cascade R-CNN (a strong detection baseline with a Transformer-backbone). All methods were trained and tested on the same data partition (DAIR-V2X-C, training/validation/testing = 70:20:10), the same preprocessing (images 640×640, normalization, enhancement strategies described in this paper), and consistent evaluation code. Inference was performed on the same edge reference platform (TensorRT FP16 acceleration, reference edge device configuration: NVIDIA Jetson Xavier AGX) to ensure comparability. Each model was run independently three times, and the average metrics were calculated (mAP@0.5, mAP@[0.5:0.95], small objective mAP@0.5, average inference latency, FLOPs, and number

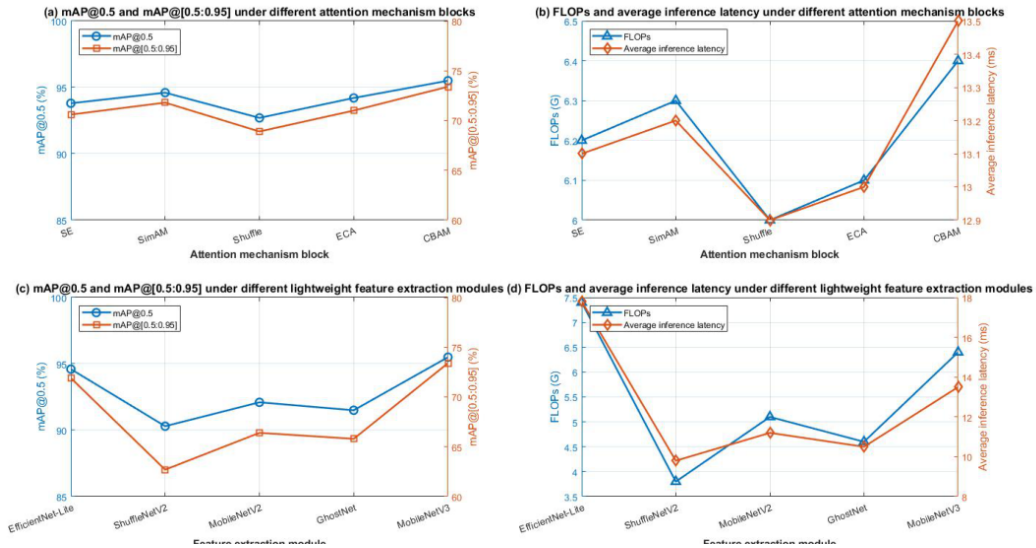

FIGURE 8. Detection performance for different attention blocks and lightweight feature extraction modules. Figure 8(a) mAP@0.5 and mAP@[0.5:0.95] for different attention blocks; Figure 8(b) FLOPs and average inference latency for different attention blocks; Figure 8(c) mAP@0.5 and mAP@[0.5:0.95] for different lightweight feature extraction modules; Figure 8(d) FLOPs and average inference latency for different lightweight feature extraction modules

of parameters). The comparison results with recent state-of-the-art methods are shown in Table 5.

TABLE 5. Comparison Results with Recent State-of-the-Art Methods

Model (Settings)	mAP@0.5 (%)	mAP@[0.5:0.95] (%)	Small target: mAP@0.5 (%)	Average inference latency (ms)	FLOPs (G)	Parameter (M)
YOLOV8s (MobileNetV3-CBAM) — This paper (TensorRT FP16)	95.5	73.4	88.1	13.5	6.4	4.9
YOLOv8-l (official large)	96.2	76.1	85.5	46.8	25.7	18.4
RTMDet-L (real-time large)	94.8	71.2	82.9	28.0	12.3	11.6
DINO-Base (DETR-type)	96.5	78.0	83.8	72.4	38.9	86.0
Swin-T + Cascade R-CNN	94.0	70.5	80.7	54.6	34.7	48.2

As shown in Table 5, the YOLOV8s (MobileNetV3-CBAM) model presented in this paper achieves excellent detection performance while maintaining extremely low computational cost and latency. With an mAP@0.5 accuracy of 95.5% and a small target mAP@0.5 accuracy of 88.1%, its average inference latency is only 13.5 ms, FLOPs are only 6.4 G, and the number of parameters is only 4.9 M. This is significantly lower than large-scale models such as YOLOv8-l (46.8 ms, 25.7 G, 18.4 M) and DINO-Base (72.4 ms, 38.9 G, 86.0 M), comprehensively demonstrating its real-time and efficiency advantages on edge devices. Although the mAP@[0.5:0.95] metric is slightly lower than DINO-Base (73.4% vs. 78.0%), its detection speed is improved by about 4–5 times, and it outperforms all the comparison models in small target recognition. This shows that the method in this paper achieves a good balance in terms of lightweight, robustness and engineering deployment performance, and fully meets the needs of real-time detection scenarios such as V2X.

VI. CONCLUSION

This paper constructs a lightweight CBAM-MobileNet detection framework based on YOLOv8s. By replacing the original backbone with MobileNetV3, this framework leverages depthwise separable convolutions and an inverted residual architecture to reduce computational overhead while preserving low-level, fine-grained information. CBAM is inserted into key residual blocks to enhance channel and spatial attention. Weighted bidirectional feature fusion is applied in the neck to enhance multi-scale context transfer. The detection head utilizes a disentangled architecture, SIOU regression constraints, and focal loss to address class imbalance. Experimental results show that the proposed model achieves 95.5% mAP@0.5 on the DAIR-V2X-C test set, with a small object mAP@0.5 of 88.1%. This represents a 6.8% improvement over YOLOv8 (MBConv-C3FB-BCFPN-CBAM), maintaining 80.4% even under extreme occlusion conditions. The average inference latency is only 13.5 ms, significantly improving the recognition accuracy of small and occluded objects and meeting the requirements of edge deployment. However, the model still has room for further optimization in extreme environments and complex multi-class scenarios. It should be noted that the experiments in this paper are mainly based on a single type of specific dataset (DAIR-V2X-C), therefore, conclusions regarding the model's generality and cross-platform applicability should be considered conditional. Future research could integrate multimodal information or adaptive feature enhancement strategies to further improve perception performance in V2X environments; further validation could be conducted on various hardware platforms and cross-domain data; and the integration of multimodal information (such as thermal

sensors or tactile signals) and adaptive feature enhancement strategies could be explored to improve the model's universal performance in complex V2X scenario.

AUTHOR CONTRIBUTIONS

Conceptualization – Fan Luo; methodology – Fan Luo and Xun Qin; investigation – Xun Qin; writing-original draft preparation – Fan Luo and Xun Qin. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] D. K. Dewangan and S. P. Sahu, "Towards the design of vision-based intelligent vehicle system: Methodologies and challenges," Evolutionary Intelligence, vol. 16, no. 3, pp. 759-800, 2023, doi: 10.1007/s12065-022-00713-2.

[2] F. Liu, Z. Lu, and X. Lin, "Vision-based environmental perception for autonomous driving," Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, vol. 239, no. 1, pp. 39-69, 2025, doi: 10.1177/09544070231203059.

[3] Y. Zhao, C. Lei, Y. Shen, Y. Du, and Q. Chen, "Improving autonomous vehicle visual perception by fusing human gaze and machine vision," IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 11, pp. 12716-12725, 2023, doi: 10.1109/TITS.2023.3290016.

[4] J. Wu and X. Zhang, "Vehicle target recognition method based on visible and infrared image fusion using bayesian inference," Applied Sciences, vol. 13, no. 14, pp. 8334-8350, 2023, doi: 10.3390/app13148334.

[5] J. Ruan, H. Cui, Y. Huang, T. Li, C. Wu, and K. Zhang, "A review of occluded objects detection in real complex scenarios for autonomous driving," Green Energy and Intelligent Transportation, vol. 2, no. 3, pp. 1-13, 2023, doi: 10.1016/j.geits.2023.100092 .

[6] Z. Zheng, J. Zhao, J. Fan, R. Bai, J. Zhao, and J. Liu, "A complex roadside object detection model based on multi-scale feature pyramid network," Scientific Reports, vol. 15, no. 1, pp. 1-15, 2025, doi: 10.1038/s41598-025-99544-1.

[7] F. Ma, R. Zhang, B. Zhu, and X. Yang, "A lightweight UAV target detection algorithm based on improved YOLOv8s model," Scientific Reports, vol. 15, no. 1, pp. 1-14, 2025, doi: 10.1038/s41598-025-00341-7.

[8] Q. Wang, G. Ye, Q. Chen, S. Zhang, and F. Wang, "A UAV perspective based lightweight target detection and tracking algorithm for intelligent transportation," Complex & Intelligent Systems, vol. 11, no. 1, pp. 73-90, 2025, doi: 10.1007/s40747-024-01687-7.

[9] J. Ni, S. Zhu, G. Tang, C. Ke, and T. Wang, "A small-object detection model based on improved YOLOv8s for UAV image scenarios," Remote Sensing, vol. 16, no. 13, pp. 2465-2483, 2024, doi: 10.3390/rs16132465.

[10] H. Zhang, Y. Gong, F. Yao, and Q. Zhang, "Research on real-time detection algorithm for pedestrian and vehicle in foggy weather based on lightweight XM-YOLOViT," IEEE Access, vol. 12, no. 1, pp. 7864-7883, 2023, doi: 10.1109/ACCESS.2023.3344666.

[11] G. Zhang, Z. Li, D. Huang, W. Luo, Z. Lu, and Y. Hu, "A Traffic Sign Recognition System Based on Lightweight Network Learning," Journal of Intelligent & Robotic Systems, vol. 110, no. 4, pp. 139-152, 2024, doi: 10.1007/s10846-024-02173-5.

[12] C. Li, Y. Zhu, and M. Zheng, "A multi-objective dynamic detection model in autonomous driving based on an improved YOLOv8," Alexandria Engineering Journal, vol. 122, no. 1, pp. 453-464, 2025, doi: 10.1016/j.aej.2025.03.020.

[13] P. Wang, X. Wang, Y. Liu, and J. Song, "Research on road object detection model based on YOLOv4 of autonomous vehicle," IEEE Access, vol. 12, no. 1, pp. 8198-8206, 2024, doi: 10.1109/ACCESS.2024.3351771.

[14] Q. Yu, H. Liu, and Q. Wu, "An improved yolo for road and vehicle target detection model," Journal of ICT Standardization, vol. 11, no. 2, pp. 197-216, 2023, doi: 10.13052/jicts2245-800X.1125.

[15] J. Yao, X. Fan, B. Li, and W. Qin, "Adverse weather target detection algorithm based on adaptive color levels and improved YOLOv5," Sensors, vol. 22, no. 21, pp. 8577-8597, 2022, doi: 10.3390/s22218577.

[16] B. Zhang, M. Simsek, M. Kulhandjian, and B. Kantarci, "Enhancing the Safety of Autonomous Vehicles in Adverse Weather by Deep Learning-Based Object Detection," Electronics, vol. 13, no. 9, pp. 1765-1788, 2024, doi: 10.3390/electronics13091765 .

[17] Y. Qiu, Y. Lu, Y. Wang, and H. Jiang, "ARODNet: Adaptive rain image enhancement object detection network for autonomous driving in adverse weather conditions," Optical Engineering, vol. 62, no. 11, pp. 118101-118101, 2023, doi: 10.1117/1.OE.62.11.118101.

[18] W. Sheng, X. Yu, J. Lin, and X. Chen, "Faster RCNN target detection algorithm integrating CBAM and FPN," Applied Sciences, vol. 13, no. 12, pp. 6913-6940, 2023, doi: 10.3390/app13126913.

[19] M. Reyad, A. M. Sarhan, and M. Arafa, "A modified Adam algorithm for deep neural network optimization," Neural Computing and Applications, vol. 35, no. 23, pp. 17095-17112, 2023, doi: 10.1007/s00521-023-08568-z.

[20] Y. Zhang, C. Chen, N. Shi, R. Sun, and Z. Q. Luo, "Adam can converge without any modification on update rules," Advances in neural information processing systems, vol. 35, no. 1, pp. 28386-28399, 2022, doi: 10.48550/arXiv.2208.09632.

[21] F. Mehmood, S. Ahmad, and T. K. Whangbo, "An efficient optimization technique for training deep neural networks," Mathematics, vol. 11, no. 6, pp. 1360-1381, 2023, doi: 10.3390/math11061360 .