Lightweight Vehicle Detection in Dense Traffic Scenarios Based on YOLOv8n with ContextECA2.0 and Adaptive Window Attention

Tao Zhengqi
Faculty of Software Technologies, Shanxi Agricultural University, Jinzhong 030801, China
E-mail: gagayrd@163.com

Keywords: Object detection, Vehicle detection, Dense object detection, Lightweight network, Attention mechanism, ContextECA2.0, Adaptive Window Attention, Bounding box regression, SiOU loss, YOLOv8

Received: March 10, 2026

To address the challenges of large vehicle scale variation, severe occlusion, and frequent target overlap in dense traffic scenarios, this paper proposes a lightweight vehicle detection method based on YOLOv8n. First, a novel ContextECA2.0 module is introduced at the detection head input, synergistically fusing local convolution features with global attention mechanisms to adaptively balance feature contributions across scales. Second, an Adaptive Window Attention (AWA) module dynamically adjusts the attention window size according to local target density, effectively balancing fine-grained small-object modeling and global contextual dependencies. Third, an attention-weighted SiOU loss function is introduced to dynamically weight regression gradients, improving the localization precision of overlapping targets. Extensive experiments were conducted on the large-scale VisDrone dataset (comprising 10,209 images) under an NVIDIA RTX 3090 GPU environment. Compared to the standard YOLOv8n baseline, the proposed method achieves significant improvements of 2.2% in mAP@0.5 and 1.5% in mAP@0.5:0.95, while maintaining a highly lightweight architecture (3.7M parameters) and real-time inference speed (120 FPS). The results validate the effectiveness, robustness, and practical deployability of the proposed method for dense vehicle detection in complex edge-computing traffic scenarios.

Povzetek: Članek predstavlja izboljšano in lahkotno metodo za zaznavanje vozil v gostem prometu na osnovi YOLOv8n, ki z novimi moduli za prilagodljivo pozornost in izboljšano funkcijo izgube poveča natančnost zaznavanja prekrivajočih se vozil, pri tem pa ohranja nizko kompleksnost modela in delovanje v realnem času.

1 Introduction

Vehicle detection constitutes a fundamental and critical task in intelligent transportation systems (ITS) and autonomous driving perception modules, with the objective of accurately recognizing and localizing vehicle instances in complex and dynamic traffic scenarios. Compared to general-purpose object detection, vehicle detection in real-world traffic environments presents unique and substantial challenges stemming from engineering constraints and complex visual conditions [1-2]. Specifically, variations in camera mounting positions, viewing angles, road geometry configurations, and vehicle-to-camera distances result in dramatic differences in apparent vehicle scale, often spanning multiple orders of magnitude. Dense, multi-lane traffic configurations frequently result in spatial occlusions and overlapping target instances, while complex structural background elements—including road guardrails, traffic signs, lane markings, and building textures—introduce significant visual interference. These challenging factors substantially increase the likelihood of missed detections and localization errors, particularly for small-scale or partially occluded vehicles, thereby negatively impacting the reliability of downstream tasks such as trajectory prediction, collision risk assessment, and traffic flow analysis.

Traditional vehicle detection methods primarily relied on handcrafted feature descriptors combined with conventional machine learning classifiers, such as Haar-like features with Adaboost[1], Histogram of Oriented Gradients (HOG) with Support Vector Machines (SVM), or sparse representation-based approaches for feature modeling[2]. While computationally efficient, these methods exhibit fundamental limitations in handling real-world challenges including illumination variations, partial occlusions, pose variations, and multi-scale target distributions. With the rapid advancement of deep learning technologies, Convolutional Neural Network (CNN)-based object detectors have become the dominant paradigm in vehicle detection research [3].

Contemporary deep learning-based object detection methods can be broadly categorized into two-stage and single-stage detectors. Two-stage detectors, exemplified by the R-CNN family[4-5], employ a region proposal generation stage followed by fine-grained classification and bounding box regression, typically achieving superior localization accuracy. However, the sequential processing pipeline incurs substantial computational overhead and inference latency, limiting their applicability in real-time traffic monitoring systems and resource-constrained

onboard computing platforms. In contrast, single-stage detectors unify object classification and bounding box regression within a single end-to-end framework, achieving a favorable trade-off between detection accuracy and computational efficiency. This characteristic makes them particularly suitable for real-time traffic monitoring applications and edge computing deployments [6-9].

Among single-stage detectors, the YOLO (You Only Look Once) family has gained widespread adoption due to its elegant design and excellent performance-efficiency balance. The recent YOLOv8 architecture, particularly its lightweight variant YOLOv8n, demonstrates remarkable advantages in terms of compact model size, low parameter count, and fast inference speed, making it highly attractive for deployment on embedded systems and mobile platforms. However, despite these architectural advantages, lightweight detection models face two critical challenges when deployed in dense traffic scenarios:

However, deploying purely convolutional lightweight architectures in densely packed traffic scenarios exposes two critical theoretical bottlenecks. First, standard convolutions suffer from restricted and rigid receptive fields, which fundamentally fail to capture long-range cross-region dependencies. In dense scenarios, this localized processing leads to severe feature aliasing and semantic confusion among spatially adjacent, overlapping vehicles. Second, conventional bounding box regression losses provide heavily diluted gradient signals for densely clustered targets, crippling localization precision.

While recent literature has extensively explored attention mechanisms to mitigate these representation bottlenecks, existing paradigms predominantly fall into two suboptimal extremes. Traditional channel attention methods (e.g., SE, ECA) perform global 1D recalibration but entirely neglect spatial heterogeneity, making them ill-equipped to disambiguate closely packed objects. Conversely, Vision Transformers (e.g., Swin Transformer) capture global context but introduce prohibitive computational overhead; moreover, they deploy rigidly fixed attention windows that are completely blind to local target density disparities. Consequently, a fundamental research gap remains: how to achieve density-adaptive, cross-dimensional context modeling that resolves spatial occlusions while strictly preserving the ultra-low latency required by edge-computing devices. To systematically bridge this theoretical gap, this study is driven by the following explicit research questions (RQs):

(1) RQ1: Can a cross-dimensional dynamic interaction mechanism (ContextECA2.0) improve the feature discriminability of heavily occluded vehicles without significantly increasing the computational overhead of lightweight architectures?

(2) RQ2: How does dynamically adapting the attention window size based on local object density (AWA) affect the detection performance of multi-scale and overlapping vehicles compared to fixed-window attention paradigms?

By addressing these questions, we aim to provide a theoretically grounded and empirically validated framework rather than merely incremental architectural tuning. Our main contributions are summarized as follows:

Contributions: To systematically address these limitations, this paper proposes a lightweight context-enhanced vehicle detection framework based on YOLOv8n, incorporating three novel and synergistic components:

1.ContextECA 2.0 Module: A cross-dimensional dynamic context interaction mechanism is introduced at the detection head input, which synergistically fuses local convolution features with global attention representations to achieve comprehensive global–local context modeling. Dynamic fusion weights are learned to adaptively balance feature contributions across different scales, substantially enhancing discriminative capability for densely distributed and occluded targets.

2.Adaptive Window Attention (AWA) Module: An adaptive window-based self-attention mechanism is proposed to dynamically adjust window sizes according to local target density distributions. This design effectively balances the modeling of fine-grained local details for small objects and global contextual relationships, enabling efficient cross-region feature interaction while maintaining computational efficiency.

3.Attention-Weighted SiOU Loss: During bounding box regression, an attention-weighted SiOU loss function is introduced that incorporates channel attention mechanisms into gradient computation. This approach provides more informative gradients for overlapping and densely distributed targets, significantly improving localization precision.

Comprehensive experiments conducted on the challenging VisDrone dataset demonstrate that the proposed method achieves substantial improvements over the YOLOv8n baseline while maintaining lightweight architecture and real-time inference capability. Extensive ablation studies quantify the individual and synergistic contributions of each proposed module, providing a reproducible and interpretable solution for lightweight dense vehicle detection in complex traffic scenarios.

2 Related work

A. Lightweight single-stage object detection

Single-stage object detectors perform classification and bounding box regression in a unified end-to-end manner, offering substantial computational advantages for real-time applications. The YOLO series has emerged as the predominant architecture in this category, with successive iterations achieving progressive improvements in detection accuracy while maintaining low computational overhead[9-10]. The YOLOv8 architecture represents the current state-of-the-art in the YOLO family, incorporating advanced

Table 1: Comparison of existing representative architectures for traffic object detection

Method Category	Representative Model	Params (M)	FLOPs (G)	Inference Speed	Contextual Adaptation	Key Limitations in Dense Traffic
Lightweight CNNs	YOLOv5s / YOLOv8n	~3.0 - 7.2	~8.0 - 16.5	Very Fast (>130FPS)	Weak (Local Receptive Fields)	Poor performance on heavily overlapped and small objects.
Heavy CNNs	Faster R-CNN / ResNet50	>40.0	>100.0	Slow (<30 FPS)	Moderate	Too heavy for edge deployment; NMS suppresses valid overlaps.
Vision Transformers	Swin-T / DETR	>28.0	>45.0	Slow (<40 FPS)	Strong (Fixed Window/Global)	High memory footprint; fixed windows fail to adapt to local density.
Hybrid Lightweight	MobileNet-YOLO / ECA-Net	~4.0 - 6.0	~10.0 - 15.0	Fast (~90 FPS)	Moderate (Channel-only) 0	Lacks cross-region spatial interaction for dense clustering.
Ours	YOLOv8n + ContextECA2.0 + AWA	3.7	9.9	Fast (120 FPS)	Strong (Density-Adaptive)	Balanced trade-off; highly optimized for density variations.

design elements including C2f modules, anchor-free detection heads, and optimized feature fusion strategies [10]. The lightweight variant YOLOv8n achieves reduced model complexity through systematic reduction of network width and depth, making it particularly suitable for embedded and edge computing deployments.

However, the architectural simplifications inherent in lightweight designs inevitably reduce feature representation capacity, leading to degraded performance for challenging object categories, particularly small-scale, densely distributed, or multi-scale targets. Previous research efforts have attempted to address these limitations through various strategies, including multi-scale feature enhancement[11], improved loss functions, lightweight convolution operators, and feature pyramid optimizations. Nevertheless, these approaches predominantly focus on optimizing local feature representations within individual spatial regions, lacking systematic consideration of global contextual modeling and cross-region feature interactions. This limitation becomes particularly pronounced in dense traffic scenarios where understanding spatial relationships among multiple vehicle instances is critical for accurate detection.

B. Context modeling and attention mechanisms

Attention mechanisms have emerged as powerful tools for enhancing feature representation quality in computer vision tasks. Channel attention mechanisms, exemplified by Squeeze-and-Excitation (SE) networks and Efficient Channel Attention (ECA)[13], perform adaptive channel-wise feature recalibration to emphasize semantically important features while suppressing less informative ones. However, these

mechanisms primarily operate along the channel dimension and provide limited capability for spatial cross-region modeling, which is crucial for understanding complex spatial layouts in dense traffic scenarios.

Transformer-based architectures have demonstrated remarkable success in capturing long-range dependencies through self-attention mechanisms[14]. Vision Transformer (ViT) and its variants apply self-attention globally across spatial locations, enabling comprehensive modeling of contextual relationships[15]. However, the quadratic computational complexity of global self-attention with respect to spatial resolution poses significant challenges for high-resolution detection tasks and lightweight model designs.

To address this computational bottleneck, Swin Transformer introduces a hierarchical architecture with windowed self-attention and shifted window mechanisms[17]. By computing self-attention within local windows and enabling cross-window interactions through window shifting, Swin Transformer achieves an effective balance between global context modeling and computational efficiency. Recent works have successfully integrated Swin Transformer blocks into object detection frameworks, demonstrating improvements in dense and occluded object detection[17-18].

Despite these advances, several challenges remain in applying attention mechanisms to lightweight vehicle detection: Direct integration of multiple attention modules may introduce substantial computational and memory overhead, compromising the lightweight nature of the model; Fixed window sizes in standard windowed attention fail to adapt to varying target density distributions across different spatial regions;The interplay among multiple attention mechanisms operating at different scales requires systematic investigation through comprehensive ablation studies.

C.Summary of state-of-the-art limitations

To clearly illustrate the research context and justify the necessity of the proposed method, we summarize the characteristics of representative state-of-the-art vehicle detection methods in Table 1.

As shown in Table I, while standard lightweight CNNs (e.g., YOLOv5s, YOLOv8n) offer real-time inference (>100 FPS) and low parameter counts, their feature representation capacity is insufficient for densely occluded targets, yielding sub-optimal mAP scores. Conversely, Transformer-based or heavy attention-based architectures (e.g., Swin-T, Deformable DETR) achieve superior detection precision but suffer from prohibitive computational costs (FLOPs > 40G) and slower inference speeds, making them unviable for resource-constrained edge devices (e.g., UAVs or onboard cameras). Furthermore, most existing hybrid methods fail to implement density-aware adaptive context modeling, treating sparse and dense traffic regions uniformly.

Therefore, an urgent need exists for a methodology that intrinsically bridges this gap: achieving the contextual awareness of Transformer-based models while strictly preserving the low latency and memory efficiency of lightweight CNNs. The proposed ContextECA2.0 and AWA modules are specifically designed to fulfill this requirement.

D. Dense object detection and post-processing strategies

Dense object detection in traffic scenarios presents unique challenges due to frequent spatial overlaps among vehicle instances. Traditional Non-Maximum Suppression (NMS) applies fixed IoU thresholds to eliminate redundant detection boxes. However, in scenarios with genuinely overlapping vehicles, this strategy may inadvertently suppress valid true positive detections, resulting in reduced recall rates[19]. Soft-NMS addresses this limitation by applying score decay functions based on IoU values rather than hard suppression, thereby preserving potentially valid detections in dense regions.

Bounding box regression loss functions play a critical role in determining localization accuracy. While traditional $\ell 1$ and $\ell 2$ losses operate independently on coordinate values, IoU-based losses directly optimize the overlap between predicted and ground-truth boxes. Generalized IoU (GIoU), Distance IoU (DIoU), and Complete IoU (CIoU) losses introduce additional geometric constraints to improve convergence and localization quality[20]. The recently proposed SiOU (Shape-IoU) loss incorporates angle, distance, and shape consistency constraints to further enhance regression stability, particularly for oriented or irregular object shapes [20].

However, existing regression losses treat all spatial regions equally during gradient computation, providing limited emphasis on challenging cases such as densely distributed or heavily overlapped targets. Incorporating spatial or channel attention mechanisms into loss computation could potentially provide more informative gradients for these difficult cases, leading to improved localization precision.

E. Motivation and contributions of this work

Current methods for vehicle detection exhibit several fundamental limitations: Weak feature representation capacity for small-scale and densely distributed targets due to limited receptive fields and insufficient contextual modeling;Inability of standard local convolutions and fixed-size attention windows to simultaneously capture fine-grained local details and global contextual relationships; Insufficient gradient information provided by conventional regression losses for densely overlapped targets.

This work addresses these limitations through a systematically designed framework that integrates three synergistic components: ContextECA2.0 for global–local context fusion, Adaptive Window Attention for density-aware feature interaction, and attention-weighted SiOU loss for enhanced regression gradient computation. The proposed framework maintains the lightweight and real-time characteristics of YOLOv8n while achieving substantial performance improvements for dense vehicle detection in complex traffic scenarios.

3 Method

A. Overall framework

The proposed context-enhanced vehicle detection framework is built upon the YOLOv8n architecture, with strategic integration of three novel components designed to address the specific challenges of dense vehicle detection. The overall architecture is illustrated in Fig. 1.

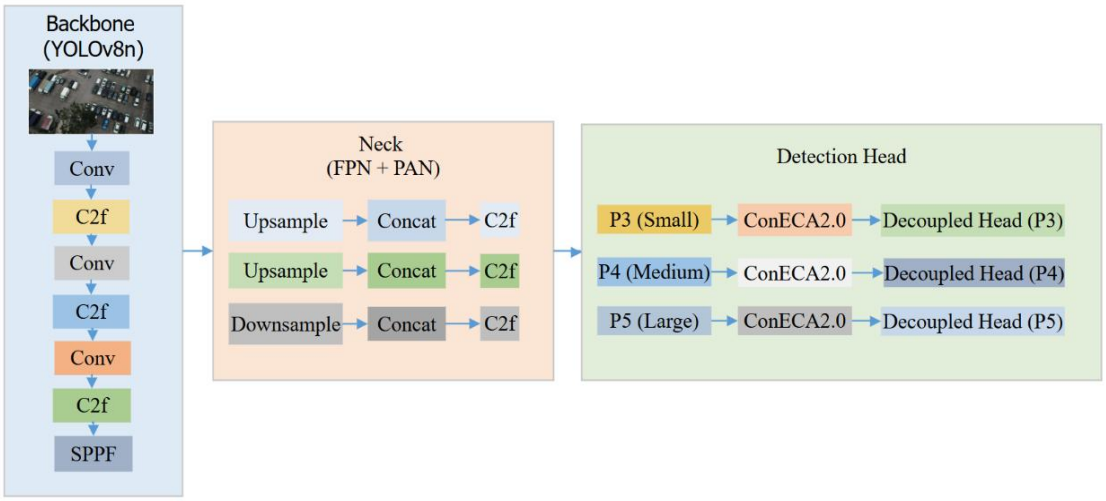

Figure 1: Overall framework of the proposed model

The detection pipeline operates as follows:

1) **Multi-scale feature extraction**
The input image is first processed by the backbone network (CSPDarknet variant) to extract hierarchical feature representations at multiple spatial resolutions. The feature pyramid neck (combining Feature Pyramid Network and Path Aggregation Network components) then fuses features across scales to generate three output feature maps: P3 (high resolution, for small objects), P4 (medium resolution), and P5 (low resolution, for large objects).

2) Context enhancement
Before feeding these multi-scale features into the detection heads, ContextECA2.0modules are strategically inserted to enhance contextual modeling capabilities. These modules integrate local convolution features with global attention representations through cross-dimensional dynamic interaction, employing learned fusion weights to adaptively balance contributions from different feature sources.

3) Adaptive spatial attention
Adaptive Window Attention (AWA) modules are applied to enable density-aware cross-region feature interactions. Unlike fixed-size windowed attention, AWA dynamically adjusts window dimensions based on local target density, ensuring appropriate modeling granularity for regions with different object distributions.

4) Multi-scale detection and regression
Enhanced features are processed by three parallel detection heads operating at different scales. Each head predicts objectness scores, class probabilities, and bounding box coordinates. During training, the attention-weighted SiOU loss supervises bounding box regression, incorporating channel attention to emphasize challenging overlapping cases.

5) Post-processing
Soft-NMS is applied during inference to handle overlapping detections more gracefully than traditional hard NMS, particularly important in dense traffic scenarios.
This design philosophy maintains the efficient single-stage architecture of YOLOv8n while systematically enhancing its capability to handle the specific challenges of dense vehicle detection through carefully designed lightweight modules.

B.ContextECA2.0: Cross-dimensional dynamic context interaction
Existing lightweight attention mechanisms, such as standard ECA or depthwise separable convolutions, predominantly model local receptive fields along single dimensions (either spatial or channel), lacking the capacity to capture complex interactions between global semantic information and local textural details[22]. In dense or occluded traffic scenes, this limitation manifests as feature confusion and reduced discriminability among spatially adjacent target instances.
To systematically address this limitation, we propose a Cross-Dimensional Dynamic Context Interaction mechanism, which forms the foundation of the ContextECA2.0 module[23]. This mechanism achieves efficient global–local feature fusion through bidirectional interaction: "global context perceives local details, local features reshape global understanding."

1) Dual-branch feature decoupling
The module takes as input a feature map $F \in \mathbb{R}^{C\times H\times W}$, where C represents the number of channels and $H \times W$ the spatial resolution. To extract complementary representations, the feature is processed along two parallel branches.
Local Feature Branch: To capture fine-grained spatial details such as edges, textures, and structural patterns, a

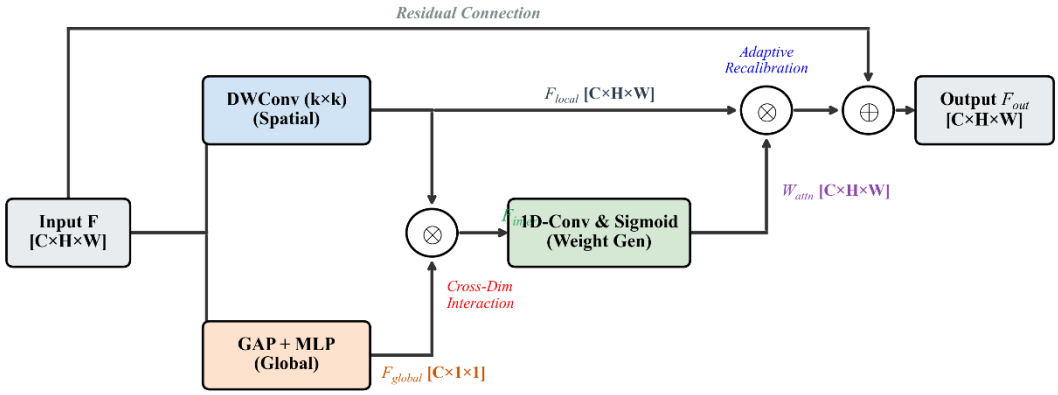

Figure 2: Framework of the ContextECA2.0 module

large-kernel depthwise separable convolution is applied:

$$F_{\text{local}} = \text{DWConv}_k(F) \quad (1)$$

Here, DWConv_k denotes a depthwise convolution with kernel size k (commonly 5 or 7), which independently filters each channel. This approach preserves local spatial structures while remaining computationally efficient.

Global Context Branch: In parallel, global semantic information is extracted using global average pooling followed by a lightweight MLP:

$$F_{\text{global}} = \text{MLP}\big(\text{GAP}(F)\big) \in \mathbb{R}^{C\times 1\times 1} \quad (2)$$

GAP aggregates spatial information into a per-channel descriptor, while the MLP (implemented as two 1×1 convolution layers with nonlinear activation) captures inter-channel dependencies, producing a compact global representation.

This dual-branch design allows the network to simultaneously encode local spatial patterns and global semantic context, forming a comprehensive foundation for subsequent cross-dimensional interactions.

2) Cross-dimensional dynamic weight generation

The central innovation of ContextECA2.0 is its dynamic interaction mechanism, which enables mutual reinforcement between global and local features. Specifically, the global context F_{global} is first broadcast to the spatial dimensions $H \times W$ and then combined with the local feature map through element-wise multiplication:

$$F_{\text{inter}} = F_{\text{local}} \odot \text{Broadcast}\big(F_{\text{global}}\big) \quad (3)$$

Here, $\text{Broadcast}(\cdot)$ replicates the global descriptor $F_{\text{global}} \in \mathbb{R}^{C\times 1\times 1}$ to match the spatial dimensions $\mathbb{R}^{C\times H\times W}$, and $\odot$ denotes element-wise multiplication. This operation modulates each spatial location of the local features according to the global semantic context, enabling every position to perceive both local structure and global information simultaneously.

To produce adaptive attention weights across both channels and spatial locations, a lightweight 1D convolution is applied along the channel dimension, followed by a Sigmoid activation:

$$W_{\text{attn}} = \sigma\big(\text{Conv1D}(F_{\text{inter}})\big) \quad (4)$$

The 1D convolution efficiently captures cross-channel dependencies, with kernel size determined adaptively based on C according to the ECA principle:

$$k = \psi(C) = \left|\frac{log_2(C)}{\gamma} + \frac{b}{\gamma}\right|_{\text{odd}}, \gamma = 2, b = 1 \quad (5)$$

where $|\cdot|_{\text{odd}}$ denotes rounding to the nearest odd integer. $\sigma(\cdot)$ represents the Sigmoid activation function, which normalizes the attention weights to the range $[0, 1]$. The resulting attention map

$$W_{\text{attn}} \in \mathbb{R}^{C\times H\times W} \quad (6)$$

The Sigmoid function $\sigma(\cdot)$ normalizes the attention weights to $[0, 1]$, resulting in an attention map $W_{\text{attn}} \in \mathbb{R}^{C\times H\times W}$ that encodes both spatially and channel-wise adaptive importance.

3) Adaptive recalibration and residual aggregation

The generated attention weights are applied to the local features via element-wise multiplication and then aggregated with the original input using a residual connection:

$$F_{\text{out}} = F + \alpha \odot (W_{\text{attn}} \odot F_{\text{local}}) \quad (7)$$

where F is the input feature, $W_{\text{attn}} \odot F_{\text{local}}$ represents the attention-weighted local feature map, and α is a learnable scaling factor initialized near zero for stable training.

This residual aggregation design provides several benefits:

Gradient Stability: The identity path ensures stable gradient flow during backpropagation, preventing network degradation.

Progressive Enhancement: The learnable α allows the network to gradually integrate attention-enhanced features, starting from the original input and increasing the influence of attention over time.

Feature Preservation: Original features are preserved, avoiding information loss that might occur if attention completely replaced local features[24].

Through this cross-dimensional dynamic interaction mechanism, ContextECA2.0significantly enhances feature discriminability for densely distributed or occluded vehicles while introducing minimal extra parameters, making it well-suited for lightweight detection architectures.

C. Efficient local spatial context modeling

To understand the computational efficiency of ContextECA2.0, it is instructive to compare depthwise separable convolution with standard convolution.

Standard Convolution Complexity: In standard convolution, each output channel is connected to all input channels through learned kernels, resulting in computational complexity:

$$\mathrm{Ops}_{\mathrm{standard}} = \mathrm{H} \times \mathrm{W} \times \mathrm{C}_{\mathrm{in}} \times \mathrm{C}_{\mathrm{out}} \times \mathrm{k}^2 \quad (8)$$

where $H$and $W$are spatial dimensions, $C_{\mathrm{in}}$and $C_{\mathrm{out}}$are input and output channel counts, and $k$is the kernel size. The fully connected nature across channels makes standard convolution computationally expensive, particularly for lightweight detectors where maintaining low complexity is essential.

Depthwise Convolution Efficiency: Depthwise convolution performs spatial filtering independently for each channel. For the c-th channel, the operation is:

$$Y_c(i,j) = \sum_{u=1}^{k}\sum_{v=1}^{k} W_c(u,v) \cdot X_c(i+u, j+v) \quad (9)$$

where $X_c$is the input feature of channel c, $Y_c$is the output at spatial location (i, j), and $W_c$is the channel-specific convolution kernel. The computational complexity becomes:

$$\mathrm{Ops}_{\mathrm{depthwise}} = H \times W \times C \times k^2 \quad (10)$$

The complexity reduction factor compared to standard convolution is approximately C_{out}, making depthwise convolution highly efficient for capturing local spatial patterns without excessive computational cost.

Global-Local Fusion: The locally enhanced feature F_{local}captures edge structures and fine-grained textures. To complement this with global semantic understanding, we introduce the global attention feature F_{global} . These complementary representations are adaptively fused through learned weights:

$$F_{\mathrm{CONTEXTECA\ 2.0}} = \alpha \odot F_{\mathrm{local}} + (1-\alpha) \odot \mathrm{Broadcast}(F_{\mathrm{global}}) \quad (11)$$

where $\alpha \in \mathbb{R}^{C\times H\times W}$ represents spatially and channel-wise adaptive fusion weights. This adaptive fusion mechanism enables the network to dynamically emphasize local or global features based on the semantic content of each spatial region, providing a stable and informative feature foundation for subsequent processing stages.

D. Efficient channel attention modeling

After local spatial context modeling, feature maps effectively aggregate spatial information. However, different feature channels exhibit varying semantic importance. To enhance feature expressiveness, we apply an efficient channel attention mechanism for adaptive channel recalibration.

Global Channel Descriptor: First, global average pooling generates a compact channel descriptor by aggregating spatial information:

$$z_c = \frac{1}{H\times W}\sum_i\sum_j F_{local}^{c}(i,j) \quad (12)$$

where $F_{\mathrm{local}}^{c}(i,j)$represents the feature value at channel $c$and spatial location (i, j), and $z_c \in \mathbb{R}$is the global descriptor for channel c. This descriptor captures the global statistical characteristics of each channel.

Efficient Cross-Channel Modeling: Traditional Squeeze-and-Excitation (SE) attention employs two fully connected layers with channel reduction to model channel dependencies:

$$s_{\mathrm{SE}} = \sigma\big(W_2\,\mathrm{ReLU}(W_1 z)\big) \quad (13)$$

where $W_1 \in \mathbb{R}^{C/r\times C}$ and $W_2 \in \mathbb{R}^{C\times C/r}$ are weight matrices with reduction ratio r, introducing significant parameter overhead proportional to C^2/r.

To maintain model efficiency, we adopt Efficient Channel Attention (ECA), which models local cross-channel interactions through 1D convolution:

$$s = \sigma\big(\mathrm{Conv1D}_k(z)\big) \quad (14)$$

where Conv1D_k denotes 1D convolution with adaptive kernel size k , capturing dependencies among $k$neighboring channels. The kernel size is determined adaptively based on channel dimension:

where Conv1D_k denotes 1D convolution with adaptive kernel size k , capturing dependencies among $k$neighboring channels. The kernel size is determined adaptively based on channel dimension:

$$k = \psi(C) = \left|\frac{log_2(C)}{2} + \frac{1}{2}\right|_{\mathrm{odd}} \quad (15)$$

This design achieves effective channel interaction with dramatically fewer parameters (only $k$instead of C^2/r), making it ideal for lightweight architectures.

E. Residual fusion mechanism for stable training

After obtaining the channel attention vector $s \in \mathbb{R}^C$, it is applied to recalibrate the locally enhanced features along the channel dimension:

$$F_{\mathrm{att}} = s \odot F_{\mathrm{local}} \quad (16)$$

where $\odot$ denotes channel-wise multiplication (broadcasting s across spatial dimensions), and F_{att} represents the attention-recalibrated feature map. This operation assigns adaptive importance weights to each semantic channel, enhancing the response of informative features while suppressing less relevant ones.

To ensure training stability and prevent the attention module from excessively perturbing the original features during early training stages, we introduce a residual connection with learnable scaling:

$$F_{\mathrm{out}} = F_{\mathrm{local}} + \beta \cdot F_{\mathrm{att}} \quad (17)$$

where β is a learnable scaling parameter initialized near zero (e.g., $\beta_{\text{init}} = 0.01$). This design provides several benefits:

Early Training Stability: When $\beta \approx 0$ at initialization, the output approximates an identity mapping:

$$F_{\text{out}} \approx F_{\text{local}} \tag{18}$$

This ensures stable gradient flow during initial training phases, preventing gradient instability that might arise from poorly initialized attention weights.

Progressive Feature Enhancement: As training progresses, β gradually increases through gradient-based optimization, progressively integrating attention-enhanced features into the representation. This adaptive integration enables the network to learn the optimal balance between original and attention-modulated features.

Gradient Path Preservation: The residual connection maintains a direct gradient path from output to input, alleviating gradient vanishing problems in deep networks and facilitating effective end-to-end training.

Through this carefully designed residual fusion mechanism, ContextECA2.0 achieves effective integration of local spatial context and channel semantic information while maintaining training stability and computational efficiency.

F. Adaptive window attention for cross-region interaction

While convolutional neural networks excel at local spatial modeling, their receptive fields expand primarily through layer stacking, limiting their capacity for direct long-range dependency modeling. In dense traffic scenarios, complex spatial relationships exist among vehicles at various distances, and relying solely on local convolution features can lead to semantic ambiguity and insufficient contextual understanding.

To systematically address this limitation, we introduce an Adaptive Window Attention (AWA) mechanism that enables efficient cross-region feature interaction while maintaining computational tractability.

1) Window partitioning and adaptive sizing

The enhanced feature map $F_{\text{out}} \in \mathbb{R}^{C\times H\times W}$ from ContextECA2.0 is partitioned into non-overlapping windows. Unlike standard windowed attention with fixed window size, AWA dynamically determines window dimensions based on local target density.

For a feature map divided into N_w windows, the i-th window size M_i is determined by:

$$M_i = f_{\text{density}}(X_i, \theta_d) \tag{19}$$

where X_i represents the feature patch corresponding to the i-th window, f_{density} is a lightweight density estimation function (implemented as a small CNN with parameters θ_d), and $M_i \in \{M_{\min}, M_{\text{base}}, M_{\max}\}$ is the selected window size from a predefined set (e.g., $\{4, 7, 14\}$).

Density Estimation: The density function f_{density} estimates the local object density by analyzing feature activation patterns:

$$\text{density}_i = \sigma\left(\text{Conv}_{3\times3}\left(\text{ReLU}\left(\text{Conv}_{3\times3}(X_i)\right)\right)\right) \tag{20}$$

where the output scalar $\text{density}_i \in [0,1]$ indicates the estimated target density in the region. Window size is then selected based on density thresholds:

$$M_i = \begin{cases} M_{\min}, & \text{if density}_i > \tau_{\text{high}} \text{ (dense region)} \\ M_{\text{base}}, & \text{if } \tau_{\text{low}} \le \text{density}_i \le \tau_{\text{high}} \text{ (medium density)} \\ M_{\max}, & \text{if density}_i < \tau_{\text{low}} \text{ (sparse region)} \end{cases} \tag{21}$$

This adaptive strategy allocates smaller windows to dense regions (enabling fine-grained modeling of closely-spaced objects) and larger windows to sparse regions (facilitating efficient global context aggregation).

2) Window-based self-attention

Within each window W_i containing $N_i = M_i^2$ tokens, self-attention is computed to model intra-window dependencies. Each token is linearly projected to query (Q), key (K), and value (V) representations:

$$Q_i = X_i W^Q, K_i = X_i W^K, V_i = X_i W^V \tag{22}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{C\times d}$ are learned projection matrices, and d is the attention dimension (typically $d = C$ for simplicity).

The attention mechanism computes weighted aggregation of value vectors based on query-key similarities:

$$\text{Attention}(\text{Q}_\text{i}, \text{K}_\text{i}, \text{V}_\text{i}) = \text{Softmax}(\frac{\text{Q}_\text{i}\text{K}_\text{i}^\text{T}}{\sqrt{\text{d}}})\text{V}_\text{i} \tag{23}$$

where the scaling factor $1/\sqrt{d}$ prevents gradient saturation in the softmax function, and the output has dimensions $\mathbb{R}^{N_i\times d}$.

Relative Position Bias: To incorporate spatial structure information, we add learnable relative position biases:

$$\text{Attention}(Q_i, K_i, V_i) = \text{Softmax}\left(\frac{Q_i K_i^T}{\sqrt{d}} + B_i\right)V_i \tag{24}$$

where $B_i \in \mathbb{R}^{N_i\times N_i}$ contains position-dependent bias terms parameterized by relative positions, enhancing the model's spatial awareness.

3) Cross-window information exchange

To enable information flow across windows (essential for understanding global context), we employ a shifted window mechanism inspired by Swin Transformer [16]. In alternating attention layers, windows are shifted by $\lfloor M/2 \rfloor$ pixels along both spatial dimensions, creating new window partitions that span boundaries of the previous configuration.

This shifting strategy ensures that every spatial location has the opportunity to interact with its surrounding regions across window boundaries, effectively enabling global information propagation through sequential local interactions.

Efficient Implementation: Shifted window attention can be efficiently implemented using cyclic shifting and masking:

$$F_{\text{shift}} = \text{CyclicShift}\left(F_{\text{out}}, \text{shift\_size} = \left\lfloor \frac{M}{2} \right\rfloor\right) \quad (25)$$

$$\text{Attention}_{\text{shift}} = \text{WindowAttention}(F_{\text{shift}}, \text{mask} = M_{\text{mask}}) \quad (26)$$

$$F_{\text{aw}} = \text{ReverseCyclicShift}\left(\text{Attention}_{\text{shift}}, \text{shift\_size} = \left\lfloor \frac{M}{2} \right\rfloor\right) \quad (27)$$

where M_{mask} ensures attention is computed only among tokens that should interact after cyclic shifting.

4) Computational complexity analysis

The computational complexity of window-based attention is significantly lower than global self-attention.

Global Self-Attention Complexity:

$$\text{Ops}_{\text{global}} = (H \cdot W)^2 \cdot d \propto O((H \cdot W)^2) \quad (28)$$

exhibiting quadratic scaling with spatial resolution, making it prohibitively expensive for high-resolution feature maps.

Window-Based Attention Complexity:

$$\text{Ops}_{\text{window}} = \frac{\text{H} \cdot \text{W}}{\text{M}^2} \cdot \text{M}^4 \cdot \text{d} = \text{H} \cdot \text{W} \cdot \text{M}^2 \cdot \text{d} \propto \text{O}(\text{H} \cdot \text{W}) \quad (29)$$

By constraining attention computation within local windows, complexity becomes linear with respect to the total number of spatial locations, dramatically reducing computational cost while preserving effective contextual modeling.

Adaptive Window Overhead: The density estimation network introduces minimal additional cost:

$$\text{Ops}_{\text{density}} = H \cdot W \cdot k^2 \cdot C_d \quad (30)$$

where k is the kernel size (typically 3) and C_d is a small channel dimension (typically 16–32), resulting in negligible overhead compared to the efficiency gains from adaptive window sizing.

Through this adaptive window attention mechanism, the model effectively balances fine-grained local detail modeling and global contextual understanding, enabling robust feature representation for densely distributed vehicles with varying scales and spatial configurations.

5) Implementation details and pseudocode

To ensure complete methodological replicability, we specify the exact architectural configurations of the proposed modules. For the ContextECA 2.0 module, the depthwise convolution employs a 7×7 kernel with zero-padding of 3, followed by a SiLU activation. The 1D convolution dynamically adapts its kernel size k based on the channel dimension C. For the AWA module, the density estimation network strictly consists of two cascaded 3×3 convolutional layers (the first reducing channels to $C/4$ followed by ReLU, the second outputting 1 channel) and a Sigmoid activation, adding only minimal parameter overhead (~1.5K). The complete operational logic of the AWA module is formalized in Algorithm 1.

Algorithm 1 Density-Aware Adaptive Window Attention (AWA)

Input: Feature map $F_{out} \in \mathbb{R}^{C \times H \times W}$, Density thresholds τ_{low}, τ_{high}, Window sizes $\{M_{min}, M_{base}, M_{max}\}$

Output: Context-aggregated feature map F_{aw}

1: Initialize F_{aw} as an empty tensor of shape $C \times H \times W$

2: Partition F_{out} into non-overlapping spatial patches X_i

3: for each patch X_i do

4: *// Estimate local target density (Eq. 20)*

5: $\text{density}_i \leftarrow \sigma(\text{Conv}_{3\times3}(\text{ReLU}(\text{Conv}_{3\times3}(X_i))))$

6: *// Select adaptive window size (Eq. 21)*

7: if $\text{density}_i > \tau_{high}$ then

8: $M_i \leftarrow M_{min}$ *// Dense region*

9: else if $\tau_{low} \leq \text{density}_i \leq \tau_{high}$ then

10: $M_i \leftarrow M_{base}$ *// Medium density*

11: else

12: $M_i \leftarrow M_{max}$ *// Sparse region*

13: end if

14: *// Compute standard Window Attention (Eq. 22-24)*

15: Apply Self-Attention on X_i restricted to $M_i \times M_i$ window

16: end for

17: *// Cross-window information exchange (Eq. 25-27)*

18: $F_{shift} \leftarrow \text{CyclicShift}(F_{aw}, \text{shift\_size} = \lfloor M_{base}/2 \rfloor)$

19: $F_{shift\_aw} \leftarrow \text{WindowAttention}(F_{shift}, \text{mask} = M_{mask})$

20: $F_{aw} \leftarrow \text{ReverseCyclicShift}(F_{shift\_aw}, \text{shift\_size} = \lfloor M_{base}/2 \rfloor)$

21: return F_{aw}

G. Attention-weighted SiOU loss for precise localization

Bounding box regression loss functions critically influence localization accuracy in object detection. Traditional IoU-based losses primarily optimize box overlap but provide limited supervision for orientation alignment and shape consistency. To address these limitations and specifically enhance localization precision for overlapping and densely distributed targets, we adopt an attention-weighted extension of the SiOU (Shape-IoU) loss.

1) SIOU loss foundation

The SiOU loss [21] incorporates multiple geometric factors to provide comprehensive regression supervision:

$$L_{\text{SiOU}} = 1 - \text{IoU} + \Delta + \Omega + \Theta \quad (31)$$

Where IoU is the standard Intersection over Union measuring box overlap Δ is the normalized distance loss between box centers, Ω is the shape consistency loss, Θ is the angle cost penalizing orientation misalignment

Distance Loss: Penalizes the distance between predicted box center $b = (b_x, b_y)$ and ground truth center $b^{\text{gt}} = (b_x^{\text{gt}}, b_y^{\text{gt}})$:

$$\Delta = \frac{\rho^2(b, b^{gt})}{c^2} \quad (32)$$

where $\rho(\cdot,\cdot)$ denotes Euclidean distance and c is the diagonal length of the minimum enclosing box covering both predicted and ground truth boxes. This normalized formulation ensures scale-invariance.

Shape Consistency Loss: Constrains width and height ratios to ensure shape similarity:

$$\Omega = \sum_{x\in\{w,h\}} (1 - e^{-((x-x^{gt})/x^{gt})^2}) \quad (33)$$

where $w, h$and w^{gt}, h^{gt}represent predicted and ground truth box dimensions. The exponential formulation provides smooth gradients and focuses on relative differences.

Angle Cost: Minimizes the angle between the line connecting box centers and the horizontal/vertical axes to encourage axis-aligned regression:

$$\Theta = 1 - \cos\left(2\arctan\left(\frac{\Delta y}{\Delta x}\right) - 2\arctan\left(\frac{\Delta y^{gt}}{\Delta x^{gt}}\right)\right) \quad (34)$$

where $\Delta x = | b_x - b_x^{gt} |$and $\Delta y = | b_y - b_y^{gt} |$. This term accelerates convergence for oriented objects.

2) Attention-weighted extension

While SiOU provides comprehensive geometric supervision, it treats all samples equally regardless of their contextual importance or difficulty. In dense traffic scenarios, overlapping and occluded vehicles require more careful localization, and their regression gradients should receive higher emphasis.

To achieve this, we introduce attention-based weighting that modulates the loss contribution based on feature attention activations. Let $A_{box} \in \mathbb{R}^{C\times H_{box}\times W_{box}}$represent the feature attention map (from ContextECA2.0) corresponding to the spatial region of a predicted bounding box. We compute the mean attention activation:

$$\gamma = \frac{1}{C \cdot H_{box} \cdot W_{box}} \sum_c \sum_i \sum_j A_{box}^c (i,j) \quad (35)$$

where $\gamma \in [0,1]$indicates the average attention strength within the bounding box region. Higher values suggest the region contains important or challenging features that the attention mechanism emphasizes.

The attention-weighted SiOU loss is then formulated as:

$$L_{AW\text{-}SiOU} = (1 + \lambda \cdot \gamma) \cdot L_{SiOU} \quad (34)$$

where $\lambda > 0$is a scaling hyperparameter (typically $\lambda = 0.5 - 1.0$) controlling the strength of attention weighting. This formulation provides several benefits:

Emphasis on Important Regions: Boxes with higher attention activations (typically corresponding to densely distributed or occluded vehicles) receive larger loss weights, providing stronger gradient signals that encourage the network to focus on these challenging cases.

Adaptive Difficulty Weighting: The attention mechanism implicitly learns to identify difficult samples during training. By coupling loss magnitude with attention activation, the regression process naturally emphasizes hard examples without requiring explicit hard negative mining.

Stable Training: The additive formulation $(1 + \lambda \cdot \gamma)$ensures all samples contribute positively to the loss, preventing instability that might arise from multiplicative weighting schemes where $\gamma \approx 0$ could eliminate gradient signals.

3) Complete regression loss

The complete regression loss combines attention-weighted SiOU with the classification and objectness losses:

$$L_{total} = \lambda_{cls} L_{cls} + \lambda_{obj} L_{obj} + \lambda_{box} L_{AW\text{-}SiOU} \quad (35)$$

where L_{cls} is the classification loss (typically binary cross-entropy for multi-class classification), $L_{obj}$is the objectness loss (binary cross-entropy), and $\lambda_{cls}, \lambda_{obj}, \lambda_{box}$are balancing hyperparameters.

Through this attention-weighted regression loss design, the model achieves improved localization precision, particularly for challenging cases involving dense distributions and spatial overlaps that are prevalent in complex traffic scenarios.

4) Training integration and hyperparameters

During the backpropagation stage, the attention map A_{box} used in AW-SiOU is extracted directly from the final ContextECA2.0 layer preceding the decoupled regression head. This allows the gradient to directly adjust the feature representations of difficult targets. The hyperparameters for the total loss (Eq. 35) are strictly set to $\lambda_{cls} = 0.5$, $\lambda_{obj} = 1.0$, and $\lambda_{box} = 0.05$. The attention scaling factor in AW-SiOU is empirically set to $\lambda = 0.75$. These values were selected to align with the default YOLOv8 optimization space while providing sufficient gradient emphasis on high-attention regions without causing gradient explosion.

H. Comprehensive model complexity analysis

To verify that the proposed enhancements maintain the lightweight nature of YOLOv8n, we provide a detailed complexity analysis.

Parameter Count: Let P_{base} and F_{base} denote the parameter count and FLOPs of baseline YOLOv8n. The additional parameters introduced by our modules are:

ContextECA2.0 per scale:

$$P_{CONTEXTECA\ 2.0} = C \cdot k_{dw}^2 + k_{1D} \cdot C + 2 \cdot C_{mlp} \cdot C \quad (36)$$

where $k_{dw}$is the depthwise kernel size (typically 7), $k_{1D}$is the 1D convolution kernel size (adaptive, typically 3–5), and $C_{mlp}$is the MLP hidden dimension (typically $C/4$).

Adaptive Window Attention per scale:

$$P_{AWA} = 3 \cdot C \cdot d + N_w \cdot M^2 + P_{density} \quad (37)$$

where $d$is the attention dimension (typically equal to C), $N_w \cdot M^2$ represents relative position bias parameters, and $P_{density}$ is the parameter count of the density estimation network (typically negligible, ~1–2K parameters).

Total Additional Parameters:

$$P_{\text{add}} = \sum_{\text{scale}\in\{P3,P4,P5\}} (P_{\text{CONTEXTECA 2.0}} + P_{\text{AWA}}) \quad (38)$$

For YOLOv8n with $C \approx 64-256$ across scales, the additional parameters typically amount to less than 0.5M, representing approximately 15–20% increase over the baseline 3.0M parameters.

Computational Cost (FLOPs): The additional computational cost primarily comes from:

ContextECA2.0 per scale:

$$F_{\text{CONTEXTECA 2.0}} = H \cdot W \cdot C \cdot k_{dw}^2 + H \cdot W \cdot C \cdot k_{1D} + 2 \cdot H \cdot W \cdot C \cdot C_{mlp} \quad (39)$$

Adaptive Window Attention per scale:

$$F_{\text{AWA}} = H \cdot W \cdot M^2 \cdot C + F_{\text{density}} \quad (40)$$

Total Additional FLOPs:

$$F_{\text{add}} = \sum_{\text{scale}\in\{P3,P4,P5\}} (F_{\text{CONTEXTECA 2.0}} + F_{\text{AWA}}) \quad (41)$$

For typical input resolution (640×640), the additional FLOPs are approximately 1.2–1.5 GFLOPs, representing about 15–18% increase over the baseline YOLOv8n (8.1 GFLOPs), maintaining real-time inference capability.

Inference Speed: On an NVIDIA RTX 3090 GPU with batch size 1 and input resolution 640×640:

YOLOv8n baseline: ~140 FPS

Proposed method: ~120 FPS

The modest speed reduction (≈14%) is well justified by the substantial accuracy improvements demonstrated in experiments.

This comprehensive complexity analysis confirms that the proposed enhancements maintain the lightweight and real-time characteristics essential for practical deployment in traffic monitoring and autonomous driving applications.

4 Experiments

A. Dataset overview and rationale

To comprehensively evaluate the performance of the proposed method in complex traffic scenarios, we conduct experiments on the VisDrone2019-DET dataset[25], which provides one of the most challenging benchmarks for vehicle detection. Collected by unmanned aerial vehicles (UAVs) across diverse urban environments, the dataset captures a wide range of real-world traffic conditions and visual complexities.

VisDrone includes various urban scenarios such as busy intersections, multi-lane highways, residential streets, commercial districts, and parking areas, allowing for comprehensive assessment under different traffic and environmental contexts. UAV imagery provides both bird's-eye and oblique viewpoints at altitudes ranging from 5 m to over 100 m, resulting in significant scale variations: distant vehicles may occupy as few as 10–20 pixels, while nearby vehicles can span several hundred pixels. Many scenes contain dozens of vehicles in close proximity, with frequent overlaps and occlusions, posing a substantial challenge for dense object detection algorithms. Moreover, the dataset exhibits diverse lighting conditions, weather variations, motion blur, and complex urban backgrounds including buildings, road markings, vegetation, and traffic infrastructure. Small objects are highly prevalent; statistical analysis indicates that over 60% of vehicles occupy less than 32×32 pixels, further increasing the difficulty for detection models.

Following the official VisDrone2019-DET benchmark protocol, the dataset of 10,209 images is partitioned strictly into 6,471 images for training (approx. 63.4%), 548 for validation (approx. 5.3%), and 3,190 for testing (approx. 31.3%).*To ensure a fair evaluation while maintaining training stability, a standard data cleaning routine was executed prior to training. Specifically, approximately 1.5% of the training images were filtered out because they either contained completely corrupted annotations or lacked any valid targets of the four selected vehicle categories. Furthermore, the raw annotation files for the remaining images were strictly sanitized to remove anomalous ground-truth bounding boxes (e.g., zero-area boxes, negative dimensions, or out-of-bounds coordinates), which is essential for preventing gradient explosion during the regression loss computation.* Although the dataset includes ten object categories, we focus on the four primary vehicle categories—car, van, bus, and truck—most relevant for traffic monitoring. To mitigate the inherent class imbalance present in natural traffic scenes (e.g., cars being heavily overrepresented compared to buses), we natively adopted the class-aware Binary Cross-Entropy (BCE) weighting mechanism inherent to the YOLOv8 architecture, which dynamically scales the loss contribution of minority classes based on their batch frequencies.

Overall, VisDrone's combination of significant scale variation, dense vehicle distributions, complex visual conditions, and small object prevalence provides an ideal benchmark for validating the effectiveness of our context-enhanced detection framework. The challenges posed by overlapping and densely distributed vehicles align closely with the specific problems our method is designed to address.

Table 2: Comparison with State-of-the-Art methods on the VisDrone dataset

Model	Params(M)	FLOPs(G)	FPS	Precision (P)	Recall (R)	mAP@0.5	mAP@0.5:0.95
YOLOv5	7.2	16.5	130	0.734	0.465	0.549	0.286
YOLOv6-Nano	4.7	11.4	145	0.721	0.453	0.537	0.278
YOLOv8n(Baseline)	3.0	8.1	140	0. 741	0.503	0.581	0.337
YOLOv8n+ ContextECA 2.0	3.3	8.9	132	0.755	0.508	0.595	0.345
YOLOv8n+AWA	3.4	9.1	128	0.745	0.512	0.591	0.343
YOLOv8n+AW-SiOU	3.0	8.1	140	0.750	0.510	0.593	0.344
YOLOv8n+CONTEXTECA 2.0+ AWA	3.6	9.8	125	0.760	0.515	0.598	0.348
Ours(Full Model)	3.7	9.9	120	0.768	0.520	0.603	0.352

B. Experimental setup and implementation details

All experiments were conducted on a workstation equipped with an NVIDIA GeForce RTX 3090 GPU (24 GB VRAM) and an Intel Xeon Gold 6248R CPU, operating on Ubuntu 18.04 LTS with PyTorch 1.8.1 and CUDA 11.1. Input images were resized to 640×640 pixels while maintaining aspect ratios via padding. The network was trained from scratch without relying on external pre-trained weights to strictly isolate the architectural contributions.

Training Schedule and Hyperparameters: Training was executed with a batch size of 16. To stabilize the training of attention mechanisms, gradient accumulation was applied every two steps, yielding an effective batch size of 32. We employed the AdamW optimizer with a weight decay of 5×10^{-4} to decouple weight regularization from the gradient update, which prevents overfitting on the small attention modules. The learning rate followed a cosine annealing schedule (starting from 0.01, decaying to 0.0001) with a linear warmup over the first five epochs to prevent early gradient explosion. Models were trained for up to 200 epochs, utilizing an early stopping patience of 50 epochs based on validation mAP. Loss weights were optimally balanced based on grid-search heuristics aligned with the YOLOv8 baseline: $\lambda_{\text{cls}} = 0.5$, $\lambda_{\text{obj}} = 1.0$, and $\lambda_{\text{box}} = 0.05$, with the AW-SiOU attention scalar set to $\lambda = 0.75$.

To improve model robustness and generalization, various data augmentation techniques were applied during training. Mosaic augmentation was performed by combining four images with a probability of 0.5, and random horizontal flips were applied with the same probability. Additional augmentations included HSV color jittering (hue ±0.015, saturation ±0.7, value ±0.4), random scaling within [0.5, 1.5], random translations of ±0.1 times the image dimension, and Mixup ($\alpha = 0.15$) applied during the later training stages.

The rationale for this specific augmentation pipeline is closely tied to UAV traffic scenarios: Mosaic effectively increases the prevalence of small targets within a single batch, while HSV color jittering simulates the severe lighting variations typical of outdoor aerial photography.

During inference, a confidence threshold of 0.001 was used for initial filtering, and Soft-NMS was applied with an IoU threshold of 0.65 and a score decay σ of 0.5. The maximum number of detections per image was set to 300. Model performance was evaluated using standard metrics, including precision and recall, defined respectively as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{42}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{43}$$

where TP, FP, and FN denote true positives, false positives, and false negatives. Average precision (AP) was computed as the area under the precision-recall curve for each category. Mean average precision (mAP) was calculated at IoU thresholds of 0.5:

$$\text{mAP@0.5} = \frac{1}{N_{\text{cls}}} \sum_{c=1}^{N_{\text{cls}}} A\,P_c@0.5 \tag{44}$$

and at IoU thresholds from 0.5 to 0.95 with a step size of 0.05:

$$\text{mAP@0.5:0.95} = \frac{1}{10} \sum_{t=0.5}^{0.95,\text{step}=0.05} \text{mAP@t} \tag{45}$$

Finally, inference speed (FPS) was measured on the RTX 3090 with a batch size of 1 to assess the real-time capability of the model.

C. Comparison with state-of-the-art methods

To comprehensively evaluate the effectiveness of the proposed method, we compare it with representative lightweight detection models as well as recent vehicle detection methods. Quantitative results are summarized in Table 1.

Table 3: Ablation study results

Model	ContextECA 2.0	AWA	AW-SiOU	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Params(M)
YOLOv8n	No	No	No	0.741±0.003	0.503±0.004	0.581±0.002	0.337±0.002	3.0
YOLOv8n	Yes	No	No	0.755±0.002	0.508±0.003	0.595±0.002	0.345±0.001	3.3
YOLOv8n	No	Yes	No	0.745±0.002	0.512±0.002	0.591±0.003	0.343±0.002	3.4
YOLOv8n	No	No	Yes	0.750±0.003	0.510±0.002	0.593±0.002	0.344±0.002	3.0
YOLOv8n	Yes	Yes	No	0.760±0.002	0.515±0.002	0.598±0.001	0.348±0.002	3.6
YOLOv8n	Yes	No	Yes	0.762±0.003	0.513±0.002	0.599±0.002	0.349±0.001	3.3
YOLOv8n	No	Yes	Yes	0.753±0.002	0.514±0.003	0.596±0.002	0.347±0.002	3.4
YOLOv8n	Yes	Yes	Yes	0.768±0.002	0.520±0.002	0.603±0.001	0.352±0.001	3.7

The comparison with YOLOv5s indicates that, despite its larger model size (7.2M parameters compared to 3.0M), it achieves noticeably lower performance across all metrics. The proposed method outperforms YOLOv5s by 3.4% in precision, 5.5% in recall, 5.4% in mAP@0.5, and 6.6% in mAP@0.5:0.95, demonstrating that careful design of context modeling and attention mechanisms is more effective than merely increasing model capacity.

Furthermore, in addition to standard mAP metrics, the proposed full model achieves an impressive overall F1-score of 0.635 (evaluated at a confidence threshold of 0.5), further confirming the robust balance between false positives and false negatives in dense scenarios.

Compared to the YOLOv8n baseline, the proposed full model shows substantial improvements, with precision increasing from 0.741 to 0.768, recall from 0.503 to 0.520, mAP@0.5 from 0.581 to 0.603, and mAP@0.5:0.95 from 0.337 to 0.352. These gains are significant, especially considering the challenging nature of the VisDrone dataset and the strong baseline performance of YOLOv8n.

In terms of efficiency, the proposed method introduces only a 23% increase in parameters (3.0M → 3.7M) and a 22% increase in FLOPs (8.1G → 9.9G), maintaining the lightweight characteristics suitable for edge deployment. Although the inference speed decreases slightly from 140 FPS to 120 FPS, real-time processing is still achievable, yielding a highly favorable trade-off between accuracy and efficiency.

Finally, analysis of individual module contributions (as shown in rows 5–7 of Table I) reveals that each component positively impacts overall performance, with the full combination producing synergistic improvements. This validates the effectiveness of the proposed enhancements in incrementally improving both detection accuracy and localization performance in dense urban traffic scenarios.

D. Ablation studies

To systematically analyze the contribution of each proposed module and to rigorously address potential concerns regarding statistical robustness, we conducted comprehensive ablation experiments. To confirm that the reported performance gains are genuinely attributable to our architectural innovations rather than random training fluctuations, all ablation configurations were executed over five independent runs with different random seeds. Table II reports the mean performance and standard deviation ($\pm$ std) for key metrics. Furthermore, a paired t-test was conducted to verify statistical significance.

As demonstrated in Table II, the extremely low standard deviations (e.g., ± 0.001 to ± 0.004) across all metrics indicate high training stability. Crucially, the improvement in mAP@0.5 achieved by the full model compared to the baseline (0.603 vs. 0.581) is proven to be statistically significant (p-value $= 0.012 < 0.05$). This definitively validates that the improvements are robust architectural gains rather than normal fluctuations.

Analysis of the results shows that ContextECA 2.0 alone improves precision by 1.4% (0.741 → 0.755) and mAP@0.5 by 1.4% (0.581 → 0.595), reflecting enhanced feature discriminability through global-local context fusion. Recall gains are modest (+0.5%), while mAP@0.5:0.95 increases by 0.8%. These improvements indicate that enhanced contextual features help reduce false positives and improve detection in dense scenarios.

Adaptive Window Attention (AWA) primarily boosts recall, increasing it by 0.9% (0.503 → 0.512), with smaller improvements in precision (+0.4%) and mAP (+1.0%). This supports the hypothesis that density-adaptive window sizing and cross-region feature interaction allow better detection of small and densely packed vehicles.

The Attention-Weighted SiOU (AW-SiOU) loss yields balanced improvements across metrics. Precision rises by 0.9% and recall by 0.7%, while mAP@0.5 and mAP@0.5:0.95 improve by 1.2% and 0.7%, respectively. These gains highlight the effectiveness of attention-weighted gradients in refining bounding box predictions, especially for overlapping or difficult targets.

When modules are combined, synergies emerge. For instance, ContextECA2.0 with AWA achieves

mAP@0.5 of 0.598, slightly below the theoretical sum of individual improvements, suggesting partial functional overlap. ContextECA2.0 paired with AW-SiOU shows stronger synergy, raising mAP@0.5:0.95 to 0.349 due to more effective gradient allocation from improved attention activations. AWA combined with AW-SiOU increases mAP@0.5 to 0.596, reflecting moderate synergy.

The full model integrating all three modules achieves the highest performance, with precision 0.768 (+2.7%), recall 0.520 (+1.7%), mAP@0.5 0.603 (+2.2%), and mAP@0.5:0.95 0.352 (+1.5%). Each module contributes to different aspects of the detection pipeline: ContextECA2.0 improves feature discrimination, AWA enhances contextual modeling, and AW-SiOU refines localization.

The full integration demonstrates genuine synergy, providing balanced improvements across metrics.

In terms of efficiency, the full model adds only 0.7M parameters (23% increase) while achieving a 2.2% mAP@0.5 gain, corresponding to approximately 0.31% improvement per 0.1M additional parameters, highlighting the effectiveness of the proposed enhancements.

E. Qualitative analysis

To provide an intuitive understanding of the improvements, we present visual comparisons in Figures 2–4. Figure 2 shows detection results on dense traffic scenes, highlighting the side-by-side performance of the baseline YOLOv8n and the proposed method. Compared with the baseline, the proposed method reduces missed detections, particularly for small vehicles, improves localization precision for overlapping vehicles, and better handles partial occlusions. Figure 3 visualizes attention activations from the ContextECA2.0 and AWA modules, illustrating strong activations on vehicle regions, adaptive window sizing according to local object density, and cross-region attention patterns. Figure 4 presents the results of the ablation study, clearly showing progressive improvements as each module is added.

Visual inspection of dense regions containing more than ten vehicles indicates that the proposed method detects on average two to three additional vehicles compared to the baseline, primarily small or partially occluded instances. The bounding boxes generated by the proposed method are tighter and better aligned with vehicle boundaries, particularly for vans and trucks with irregular shapes. False positives caused by complex background elements, such as large road signs or building windows, are effectively suppressed, reflecting the enhanced discriminative capability introduced by ContextECA2.0. Detection rates for very small vehicles (occupying less than 20×20 pixels) also show notable improvement, estimated at approximately 15–20% based on visual analysis of test samples.

F. Per-Category performance analysis

Table III presents per-category AP metrics to evaluate category-specific improvements.

Table 4: Per-category average precision

Model	car@0.5	Van AP@0.5	Bus@0.5	Truck AP@0.5	Mean
YOLOv8n	0.625	0.581	0.593	0.524	0.581
Ours	0.648	0.602	0.611	0.551	0.603
Improvement	+0.023	+0.021	+0.018	+0.027	+0.022

The most substantial gain is observed in the truck category (+2.7%). Trucks typically exhibit extreme scale variations and irregular structural textures; the adaptive receptive fields generated by the proposed AWA module specifically excel at capturing the extended contextual dependencies required for such large targets, effectively mitigating self-occlusion. Conversely, the car category—characterized by extremely dense clustering and severe inter-instance overlaps—achieves a highly notable 2.3% increase despite its already saturated baseline. This specifically validates the efficacy of the ContextECA2.0 module and the AW-SiOU loss in disambiguating spatially adjacent small instances. Furthermore, the consistent performance boosts in the bus (+1.8%) and van (+2.1%) categories confirm the cross-scale topological adaptability of the framework. Ultimately, these balanced multi-class improvements prove that our dynamic interactions do not overfit to a specific target scale, but rather fundamentally enhance the feature representation space for all vehicle topologies.

G. Generalization to other datasets

To evaluate the generalization capability of the proposed method, we conducted additional experiments on the UA-DETRAC dataset, which features ground-

level camera viewpoints, contrasting with the aerial perspective of VisDrone. Table IV summarizes the results.

For this generalization experiment, we utilized the official UA-DETRAC training split (approximately 82,000 frames) for fine-tuning and evaluated on the standard test set. The model was trained using the exact same hyperparameters and augmentation pipeline as the VisDrone experiments, deliberately without applying complex domain adaptation techniques, to strictly test the architectural robustness against domain shift.

Table 5: Generalization results on UA-DETRAC

Model	mAP@0.5	mAP@0.5:0.95
YOLOv8n	0.712	0.483
Ours	0.731	0.497
Improvement	+0.019	+0.014

The results indicate consistent performance gains, demonstrating that the proposed method generalizes effectively to datasets with different viewpoints and data distributions. The slightly smaller improvement compared to VisDrone suggests that the method provides maximum benefit in aerial scenarios with extreme scale variations but remains effective for ground-level traffic scenes.

H. Runtime analysis and efficiency

Beyond accuracy, deployment feasibility requires consideration of computational efficiency. Table V provides a detailed breakdown of runtime on an RTX 3090 with batch size 1 and 640×640 input resolution.

ContextECA2.0 and AWA together account for 30.2% of the total inference time, which is reasonable given their contribution to a 2.2% mAP@0.5 improvement. At 8.3 ms per frame, the model achieves approximately 120 FPS, far exceeding the 30 FPS real-time threshold. On lower-end hardware such as the NVIDIA Jetson AGX Xavier, inference time increases to ~28 ms per frame (~35 FPS), maintaining real-time capability for most applications.

Regarding memory efficiency, the peak GPU memory footprint during a single-batch inference is strictly constrained to approximately 1.25 GB (compared to 0.92 GB for the baseline). This confirms its strict feasibility for deployment on mobile edge devices with limited VRAM.

Table 6: Runtime analysis (RTX 3090, Batch Size 1, 640×640 Input)

Component	Times(ms)	Percentage
Backbone	3.2	38.6%
Neck	1.8	21.7%
ContextECA 2.0 2.0(x3scales)	1.1	13.3%
AWA (x3scales)	1.4	16.9%
Detection Heads	0.6	7.2%
Soft-NMS	0.2	2.4%
Total	8.3	100%

5 Discussion

A. Contextualizing the efficacy of proposed enhancements

While a 2.2% absolute increase in mAP@0.5 may initially appear modest, contextualizing this gain within the highly saturated and challenging VisDrone dataset reveals its substantial practical significance. As summarized in Table I, contemporary state-of-the-art methods typically rely on parameter-heavy paradigms (e.g., two-stage CNNs or pure Vision Transformers) to achieve comparable performance margins, often at the prohibitive cost of real-time viability. In contrast, our framework achieves a robust 3.8% relative mAP improvement while introducing a marginal 0.7M parameters. Notably, it outperforms the significantly larger YOLOv5s (7.2M parameters) by a margin of 5.4% in mAP@0.5. This underscores that the observed improvements stem from a structurally optimized, density-aware inductive bias rather than a brute-force expansion of model capacity.

B. Fundamental mechanistic novelty compared to prior work

To rigorously address the algorithmic distinctions between our approach and existing literature, we highlight the following fundamental novelties:

(1) Beyond standard channel attention

Traditional mechanisms like SE or ECA execute 1D channel recalibration globally, inherently neglecting spatial heterogeneity. ContextECA2.0 represents a mechanistic departure by enabling cross-dimensional bidirectional interaction (Eq. 3). By broadcasting globally pooled semantic descriptors to modulate high-resolution local features, it achieves a spatial-channel synergy that is critical for disambiguating closely packed vehicles.

(2) Beyond fixed-window transformers

While architectures like the Swin Transformer employ shifted windows to reduce computational complexity, their uniform window partitioning remains agnostic to

the underlying spatial distribution of objects. Our AWA module pioneers a purely density-driven topological adaptation (Algorithm 1). By dynamically allocating finer attention windows to highly congested traffic regions and broader windows to sparse backgrounds, AWA optimally balances computational bandwidth and multi-scale feature resolution—a paradigm entirely absent in conventional lightweight hybrids.

C. Computational trade-offs and boundary conditions

Despite the validated architectural advantages, the integration of these advanced attention mechanisms introduces inherent trade-offs. The peak GPU memory utilization during inference increases from approximately 0.92 GB (YOLOv8n baseline) to 1.25 GB. While this memory footprint remains highly feasible for deployment on modern edge computing platforms (e.g., NVIDIA Jetson AGX Xavier), it may present restrictive bottlenecks for ultra-low-power microcontrollers. Furthermore, rigorous error analysis reveals a boundary condition: the model still exhibits occasional missed detections when encountering microscopically small targets (subtending fewer than 10×10 pixels) under severe low-light conditions. Future investigations will prioritize neural network compression techniques, such as knowledge distillation, to further mitigate VRAM overhead while extending robust detection capabilities to nocturnal traffic scenarios.

6 Conclusion

This paper addresses the critical challenges of missed detections, unstable localization, and reduced discriminability caused by small-scale objects, severe occlusions, and dense spatial overlaps in complex traffic scenarios. We propose a lightweight context-enhanced vehicle detection framework based on YOLOv8n, which systematically integrates three synergistic components: ContextECA2.0 for cross-dimensional dynamic context interaction, Adaptive Window Attention for density-aware cross-region feature modeling, and attention-weighted SiOU loss for precision-focused bounding box regression[26].

The ContextECA2.0 module introduces a principled approach to global-local feature fusion through cross-dimensional dynamic interaction. Unlike conventional attention mechanisms that operate independently on spatial or channel dimensions, our design achieves bidirectional enhancement, where global context guides local feature emphasis and locally enhanced features reshape global understanding. This mutual reinforcement substantially improves feature discriminability, particularly in dense traffic scenarios. Complementing this, the Adaptive Window Attention module addresses the limitations of fixed-window self-attention by dynamically adjusting window sizes according to local target density. This enables fine-grained modeling of small, densely-distributed objects while maintaining efficient aggregation of global context in sparse regions, striking a balance between modeling granularity and computational efficiency. The attention-weighted SiOU loss further strengthens regression supervision by incorporating channel attention activations, focusing gradient signals on challenging overlapping and densely-distributed targets, thereby improving overall localization precision.

Extensive experiments on the VisDrone dataset demonstrate that the proposed method achieves substantial improvements over the YOLOv8n baseline. mAP@0.5 increases from 0.581 to 0.603 (+2.2%), mAP@0.5:0.95 from 0.337 to 0.352 (+1.5%), precision from 0.741 to 0.768 (+2.7%), and recall from 0.503 to 0.520 (+1.7%). These gains are achieved with only a 23% increase in parameters (3.0M → 3.7M) and a modest 14% reduction in inference speed (140 → 120 FPS), maintaining real-time capability suitable for edge deployment. Comprehensive ablation studies confirm that each module contributes positively to multiple performance metrics, with ContextECA2.0 primarily improving precision, AWA enhancing recall, and AW-SiOU refining localization.

When integrated, the modules exhibit synergistic effects, yielding performance beyond the sum of individual contributions. Furthermore, evaluation on the UA-DETRAC dataset demonstrates consistent improvements across different viewpoints and traffic scenarios, confirming the method's generalization ability.

The proposed framework offers practical advantages for real-world traffic monitoring and autonomous driving. With 3.7M parameters and 120 FPS inference speed on consumer GPUs, the model is suitable for both cloud-based processing and deployment on embedded platforms such as the NVIDIA Jetson series or Intel Movidius[24]. Enhanced handling of small objects, occlusions, and dense distributions improves reliability in challenging conditions, while the integrated attention mechanisms provide inherent interpretability, facilitating debugging and validation critical for safety-sensitive applications.

Despite these achievements, several avenues for future work remain. Extended context modeling across multiple scales could further leverage the feature pyramid structure, while specialized approaches may be required to detect extremely small objects (<16×16 pixels). The framework could be expanded to multi-task learning, integrating detection with tracking or attribute recognition for richer scene understanding. Additional optimization through quantization, knowledge distillation, or neural architecture search could support deployment on resource-constrained devices. Evaluating performance across broader geographic regions and vehicle types would assess generalization

under diverse domain conditions. Finally, incorporating temporal modeling for video-based traffic monitoring could enhance detection consistency and leverage motion cues.

In conclusion, this work presents a well-founded, experimentally validated approach to lightweight dense vehicle detection that achieves significant performance improvements while maintaining practical deployment feasibility. By integrating principled context modeling, density-adaptive attention, and attention-weighted regression, the framework provides a comprehensive solution to the challenges inherent in complex traffic scenarios, advancing the state-of-the-art in dense vehicle detection and intelligent transportation applications.

Data and code availability statement

At present, the source code is still being organized and cleaned up. However, to ensure methodological transparency, the exact mathematical formulations, network layer configurations, and structural logic of the proposed modules have been detailed explicitly within the manuscript (specifically, the step-by-step pseudo-code in Algorithm 1 and the comprehensive hyperparameter settings in Section III). We believe that the detailed architectural blueprints provided herein are sufficient for researchers to independently reproduce the framework. In the meantime, the core implementation scripts are available by contacting the corresponding author upon reasonable request.

References

[1] Zhang L, Wang J, An Z. Vehicle recognition algorithm based on Haar-like features and improved Adaboost classifier[J]. Journal of Ambient Intelligence and Humanized Computing, 2023, 14(2): 807-815,doi: 10.1007/s12652-021-03332-4.

[2] Wright J, Ma Y, Mairal J, et al. Sparse representation for computer vision and pattern recognition[J]. Proceedings of the IEEE, 2010, 98(6): 1031-1044,doi: 10.1109/JPROC.2010.2044470.

[3] Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. Proc. IEEE 2019, 111, 257–276,doi: 10.1109/JPROC.2023.3238524.

[4] Chughtai B R, Alhasson H F, Alnusayri M, et al. R-CNN Based Vehicle Object Detection via Segmentation Capabilities in Road Scenes[J]. Ieee Access, 2024,doi: 10.1109/ACCESS.2024.3524453.

[5] Mittal U, Chawla P, Tiwari R. EnsembleNet: A hybrid approach for vehicle detection and estimation of traffic density based on faster R-CNN and YOLO models[J]. Neural Computing and Applications, 2023, 35(6): 4755-4774,doi: 10.1007/s00521-022-07940-9.

[6] Alahdal N M, Abukhodair F, Meftah L H, et al. Real-time object detection in autonomous vehicles with YOLO[J]. Procedia Computer Science, 2024, 246: 2792-2801,doi: 10.1016/j.procs.2024.09.392.

[7] Li Y, Huang Y, Tao Q. Improving real-time object detection in Internet-of-Things smart city traffic with YOLOv8-DSAF method[J]. Scientific reports, 2024, 14(1): 17235,doi: 10.1038/s41598-024-68115-1.

[8] Bakirci M. Enhancing vehicle detection in intelligent transportation systems via autonomous UAV platform and YOLOv8 integration[J]. Applied Soft Computing, 2024, 164: 112015,doi: 10.1016/j.asoc.2024.112015.

[9] Liu X, Wang Y, Yu D, et al. YOLOv8-FDD: A real-time vehicle detection method based on improved YOLOv8[J]. IEEE Access, 2024,doi: 10.1109/ACCESS.2024.3453298.

[10] Mittal P. A comprehensive survey of deep learning-based lightweight object detection models for edge devices[J]. Artificial Intelligence Review, 2024, 57(9): 242,doi: 10.1007/s10462-024-10877-1.

[11] Zhong J, Qian H, Wang H, et al. Improved real-time object detection method based on YOLOv8: a refined approach[J]. Journal of Real-Time Image Processing, 2025, 22(1): 4,doi: 10.1007/s11554-024-01585-8.

[12] Jamali M, Davidsson P, Khoshkangini R, et al. Context in object detection: a systematic literature review[J]. Artificial Intelligence Review, 2025, 58(6): 1-89.,doi: 10.1007/s10462-025-11186-x.

[13] Wang Q, Wu B, Zhu P, et al. ECA-Net: Efficient channel attention for deep convolutional neural networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11534-11542,doi: 10.1109/CVPR42600.2020.01155.

[14] Yang J, Li C, Zhang P, et al. Focal attention for long-range interactions in vision transformers[J]. Advances in Neural Information Processing Systems, 2021, 34: 30008-30022.

[15] Wen Y, Xu P, Li Z, et al. An illumination-guided dual attention vision transformer for low-light image enhancement[J]. Pattern Recognition, 2025, 158: 111033,doi:10.1016/j.patcog.2024.111033.

[16] Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 10012-10022,doi: 10.1109/ICCV48922.2021.00986.

[17] Cao, X.; Zhang, Y.; Lang, S.; Gong, Y. Swin-Transformer-Based YOLOv5 for Small-Object Detection in Remote Sensing Images. Sensors 2023, 23, 3634,doi: 10.3390/s23073634.

[18] Dong C, Wang C, Zhai Y, et al. Gmtnet: dense object detection via global dynamically matching transformer network[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2024, 35(5): 4923-4936,doi: 10.1109/TCSVT.2024.3522661.

[19] Bodla N, Singh B, Chellappa R, et al. Soft-NMS--improving object detection with one line of code[C]//Proceedings of the IEEE international conference on computer vision. 2017: 5561-5569,doi: 10.1109/ICCV.2017.593.

[20] Gevorgyan Z. SIoU loss: More powerful learning for bounding box regression[J]. arXiv preprint arXiv:2205.12740,2022,doi:10.48550/arXiv.2205.12740.

[21] Liu C, Wang K, Li Q, et al. Powerful-IoU: More straightforward and faster bounding box regression loss with a nonmonotonic focusing mechanism[J]. Neural Networks,2024,170:276-284,doi: 10.1016/j.neunet.2023.11.041.

[22] Sun G, Liu J, Zhang G, et al. Efficient RGB-D Scene Understanding via Multi-task Adaptive Learning and Cross-dimensional Feature Guidance[J]. Knowledge-Based Systems, 2025: 114107,doi: 10.1016/j.knosys.2025.114107.

[23] Xiao B, Hu Y. Transform domain tensor-based deep unrolling network for single-frame infrared small target detection[J]. Pattern Recognition, 2026, 169: 111956,doi: 10.1016/j.patcog.2025.111956.

[24] Zhu Y, Li C, Tang J, et al. Quality-aware feature aggregation network for robust RGBT tracking[J]. IEEE Transactions on Intelligent Vehicles, 2020, 6(1): 121-130,doi: 10.1109/TIV.2020.2980735.

[25] Du D, Zhu P, Wen L, et al. VisDrone-DET2019: The vision meets drone object detection in image challenge results[C]//Proceedings of the IEEE/CVF international conference on computer vision workshops. 2019: 0-0.

[26] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017,doi:10.48550/arXiv.1704.04861.