electronics

MDPI

Article

An Improved YOLOv11 Recognition Algorithm for Heavy-Duty Trucks on Highways

Junkai Guo [1,2] and Mingjiang Zhang [1,*]

[1] College of Physics, Taiyuan University of Technology, Taiyuan 030024, China; kaixinlaugh901@163.com
[2] Shanxi Intelligent Transportation Institute Co., Ltd., Taiyuan 030036, China
* Correspondence: zhangmingjiang@tyut.edu.cn

Abstract

This paper presents an enhanced YOLOv11-based algorithm for highway freight truck tarpaulin recognition to enhance real-time performance and accuracy in identifying truck axle types and tarpaulin materials. The proposed methodology incorporates four key innovations. First, the lightweight Spatial and Channel Reconstruction Convolution (SCConv) module is introduced to replace standard convolutional layers in the YOLOv11 backbone feature extraction network, which enables maintaining strong feature extraction capabilities while reducing model parameters and computational complexity. Second, a Channel-Spatial Multi-scale Attention Module (CSMAM) is integrated with the C3k2 module of the YOLOv11 feature fusion network, thereby strengthening the network's capacity to learn both truck body features and tarpaulin coverage characteristics. Third, a novel Dual-Enhanced Channel Detection Head (DEC-Head) detector is designed to improve recognition performance under ambiguous conditions and reduce parameter quality. Finally, the SIoU loss function is adopted to replace the conventional bounding box loss function, substantially improving prediction box accuracy. Comprehensive experimental results demonstrate that compared to the baseline YOLOv11 algorithm, our proposed method achieves an approximate 4.4% increase in precision, 5.2% improvement in recall rate, and 7.2% higher mean Average Precision (mAP), while also achieving a significant improvement in inference speed (Frames Per Second, FPS), establishing superior recognition performance for truck tarpaulin detection tasks.

Keywords: YOLOv11; truck tarpaulin recognition; attention mechanism; detection head; loss function

check for updates

Academic Editor: Chiman Kwan

Received: 3 November 2025
Revised: 19 November 2025
Accepted: 22 November 2025
Published: 25 November 2025

Citation: Guo, J.; Zhang, M. An Improved YOLOv11 Recognition Algorithm for Heavy-Duty Trucks on Highways. *Electronics* **2025**, *14*, 4621. https://doi.org/10.3390/electronics14234621

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The intelligent construction of expressways constitutes a critical component in implementing China's national strategy to build a transportation powerhouse [1], which aims to enhance transportation efficiency, ensure public safety, and protect life and property. Within this framework, ensuring the safety of "two-passenger, one-hazardous, one-heavy-freight" road transport vehicles (i.e., long-distance passenger coaches, tourist buses, hazardous material carriers, and heavy trucks) is crucial for preventing major transportation accidents [2]. Incomplete statistics show that heavy trucks account for less than 10% of China's total motor vehicle fleet. Yet they are involved in approximately 25% of traffic accidents, and their accident severity rates are disproportionately higher than those of other vehicle types [3]. Among such incidents, regulatory violations during cargo transportation, particularly safety hazards arising from improper tarpaulin usage, demand urgent attention.

This highlights the critical importance of tarpaulin recognition technology in intelligent expressway systems. Strengthening the application and deployment of tarpaulin recognition can effectively mitigate road accidents caused by cargo spillage, elevate expressway operational safety and traffic efficiency, thereby providing robust support for implementing the transportation powerhouse strategy.

In recent years, deep learning techniques, particularly the YOLO series of object detection algorithms, have been extensively adopted in diverse fields of image processing [4], ranging from the common vehicle and cargo recognition in intelligent transportation systems to more specialized industrial inspection tasks such as nondestructive testing of rail defects on few-shot occasion [5]. Vehicle and cargo feature recognition represents a critical application of object detection in intelligent transportation systems and serves as an active response to the "Internet+" paradigm [6]. Sensor-based approaches have been explored for truck tarpaulin monitoring, for instance, by installing position sensors along tarpaulin edges and integrating axle load sensors to determine truck status [7]. However, sensor vulnerability to interference and the limited adaptability of image fusion algorithms in complex scenarios remain concerns. Wei et al. [8] developed a Bi-Linear CNN model that combines ResNet and VGGNet architectures to construct three bilinear convolutional neural network structures for vehicle type recognition. However, insufficient comparison with state-of-the-art models leaves the approach's comparative advantages unclear. Negti and Clady [9] utilized vehicle frontal images' four-directional edges as features, employing an oriented contour point model for vehicle type classification, achieving an average recognition accuracy of approximately 85.6%. Nevertheless, the model's simplicity may inadequately represent complex vehicle features, exhibiting weak differentiation capability for visually similar yet distinct vehicle types. Sun et al. [10] proposed an improved deep learning method based on CNN and transfer learning for vehicle type recognition in surveillance imagery. Wang et al. [11] designed an enhanced RPN network within the Faster R-CNN framework to detect and recognize common vehicles (sedans, buses, coaches) in traffic scenes. Notably, neither study addressed adaptability challenges in low-visibility conditions (nighttime, rain, fog), where recognition performance may degrade significantly. Li et al. [12] proposed a ResNet-based lightweight vehicle detection algorithm for similar-type vehicle classification, achieving approximately 3% higher mean Average Precision (mAP) on public datasets, including the BIT-Vehicle dataset under nearly identical inference time conditions. Sun et al. [13] developed a vehicle detection method based on a MobileViT lightweight network, which maintained detection accuracy while ensuring real-time inference performance. Li et al. [14] introduced an improved Faster R-CNN method for urban road truck detection, attaining approximately 92.61% Average Precision (AP) on the KITTI dataset. Trivedi et al. [15] investigated computer-vision-based vehicle safety systems by looking both inside and outside the vehicle, exploring foundational approaches to driver assistance and external scene understanding. More recently, advancements in deep learning have led to the development of enhanced algorithms like YOLOv8-CX for UAV-based vehicle detection, demonstrating superior recognition performance on specialized datasets. It is also worth noting that subsequent iterations like YOLOv12 have emerged, further pushing the boundaries of real-time detection performance, as seen in related works [16].

Although the aforementioned works have improved vehicle feature and tarpaulin recognition to some extent, the recognition research remains insufficient for highway freight trucks due to factors such as diverse truck models, numerous tarpaulin types, and the lack of open-source datasets. The pursuit of efficient and accurate vision systems for intelligent transportation is further evidenced by concurrent advancements in several key areas, which collectively inform the design philosophy of our work. Firstly, the development of ultra-lightweight models such as GhostNet [17], which enables real-time facial gesture

recognition even in simulation environments, validates the critical importance of architectural efficiency—a principle that directly motivates our integration of the lightweight SCConv module into the backbone network. Secondly, within the YOLO family itself, architectures like HR-YOLO [16] demonstrate the significant performance gains achievable by integrating high-resolution networks for precise segmentation and detection in critical traffic scenarios (e.g., emergency escape ramps), underscoring the value of enhanced feature representation that guides our design of the CSMAM attention mechanism. Finally, at the system level, research into V2X (Vehicle-to-Everything) networks [18] for real-time cooperative autonomous driving outlines a future of deeply interconnected and secure transportation ecosystems. This broader context reinforces the necessity of developing reliable, specialized perception modules like the one proposed in this paper, which can serve as fundamental components within such a larger intelligent infrastructure. Inspired by these trajectories in model lightweighting, architectural refinement, and system integration, our work focuses on addressing the specific, underexplored challenge of fine-grained attribute recognition for heavy-duty trucks on highways. Therefore, this paper aims to utilize image data of heavy-duty trucks collected from Shanxi Province's Highway Video Cloud Platform and proposes an enhanced YOLOv11-based algorithm for highway freight truck detection and fine-grained attribute classification (including axle type and tarpaulin material). Our principal contributions are as follows:

(1) Given the dual requirements of real-time performance and accuracy in expressway truck tarpaulin detection, coupled with constrained hardware resources, we replace the convolutional layers in YOLOv11's backbone feature extraction network with a lightweight SCConv module. This modification reduces model parameters and computational load while preserving robust feature extraction capabilities.

(2) To enhance the model's focus on critical features and thereby improve overall object detection performance, we integrate the CSMAM mechanism following the channel concatenation layer of the C3k2 module. This enables the model to precisely concentrate on essential features, effectively boosting detection accuracy while strengthening adaptability across diverse scenarios and simultaneously enhancing generalization capability.

(3) To substantially enhance object detection performance and address the suboptimal performance of the original detection head in complex scenarios, the DEC-Head detector is employed, which has superior architectural design and algorithmic mechanisms. It can enhance detection accuracy while improving adaptability to challenging environments and boosting detection robustness.

(4) To ensure effective convergence of the loss function network during training, we replace the bounding box loss function with Scylla-IoU (SIoU) [19]. This modification enhances the regression accuracy of predicted bounding boxes while optimizing gradient behavior for stable model optimization.

The remainder of this paper is organized as follows. Section 2 presents the materials and methods, beginning with an overview of the baseline YOLOv11 network, followed by detailed descriptions of our proposed enhancements: the SCConv-enhanced backbone, the CSMAM attention mechanism, the DEC-Head detector, and the SIoU loss function. Section 3 provides the experimental results and analysis, including the dataset description, evaluation metrics, ablation studies, comparative experiments, and visual analysis. Finally, Section 4 concludes the paper by summarizing the main findings and discussing potential future work.

2. Materials and Methods

The objective of this study is a joint object detection and classification task. Specifically, the model is designed to localize heavy-duty trucks in highway surveillance images and simultaneously predict two key attributes for each detected truck: (1) the number of axles (a critical parameter for vehicle classification in tolling and regulation), and (2) the material of the tarpaulin cover (a primary factor influencing cargo safety and potential violations). Therefore, our category set is defined by the combination of these two attributes, resulting in six distinct classes: four-axle_canvas, four-axle_PE, four-axle_PVC, six-axle_canvas, six-axle_PE, and six-axle_PVC.

2.1. YOLOv11 Network

YOLOv11 is a next-generation object detection algorithm developed by Ultralytics, featuring substantial improvements in architectural design and training methodology over previous YOLO versions. It integrates improved model architecture design, advanced feature extraction techniques, and refined training strategies. The corresponding network structure is illustrated in Figure 1.

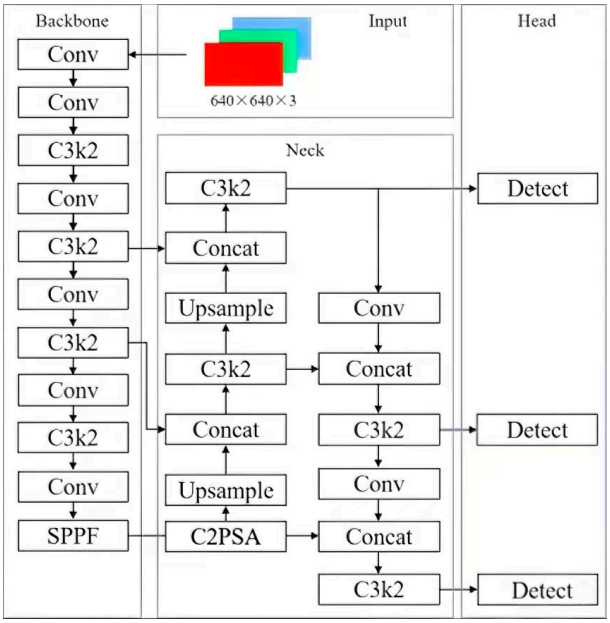


Figure 1. Structure of YOLOv11.

(1) Backbone Feature Extraction Network (Backbone): It employs C3k2 modules (a highly efficient convolutional block combining standard convolutions and residual connections) as its core component, splits the feature maps, and applies a 3×3 small convolutional kernel to optimize information flow, which can reduce parameter count while enhancing feature representation capability. The integration of the C2PSA module (a channel-to-space transformation module) and Partial Spatial Attention (PSA) strengthens spatial information awareness and dynamically focuses on critical regions. Residual connections and bottleneck structures further mitigate gradient vanishing and improve training efficiency. On the COCO dataset, the backbone achieves higher mAP with significantly fewer parameters (approximately 22% reduction compared to YOLOv8m), balancing accuracy and lightweight design.

(2) Feature Fusion Network (Neck): Employing a Path Aggregation Network (PAN) structure, it integrates the C3k2 module and Spatial Pyramid Pooling Fast (SPPF) module (a rapid variant of spatial pyramid pooling that generates fixed-size representations from feature maps of any scale), fusing shallow details with deep semantics through

cross-layer connections. Adaptive Spatial Feature Modulation (ASFM) is introduced to dynamically adjust feature weights across spatial dimensions, enhancing localization accuracy in complex scenarios. Furthermore, by incorporating the High-level Screening Feature Pyramid Network (HS-FPN), which utilizes channel attention mechanisms to filter and refine low-level features, the network effectively mitigates detection errors caused by scale variation while significantly boosting small-object detection capability.

(3) Detection Head (Head): Maintaining the multi-scale prediction strategy, the head outputs bounding boxes from P3, P4, and P5 feature maps for small, medium, and large targets, respectively. Computational efficiency is optimized through depthwise separable convolutions, where traditional convolutions in classification branches are replaced by Depthwise Separable Convolution (DW-Conv, which factorizes a standard convolution into a depthwise convolution followed by a pointwise convolution, thereby significantly reducing computational cost and model parameters). Through dynamic weight allocation, the head enhances detection stability for dense scenes and targets with extreme aspect ratios in complex traffic environments.

2.2. Enhanced YOLOv11 Architecture

To meet the demanding requirements for real-time performance and accuracy in heavy-duty truck tarpaulin recognition, four key improvements are made to the traditional YOLOv11 network model: Refinement of the backbone feature extraction network; Integration of an attention mechanism; Proposal of a novel detection head replacing the original structure; Optimization of the loss function. The architecture of the enhanced network is depicted in Figure 2.

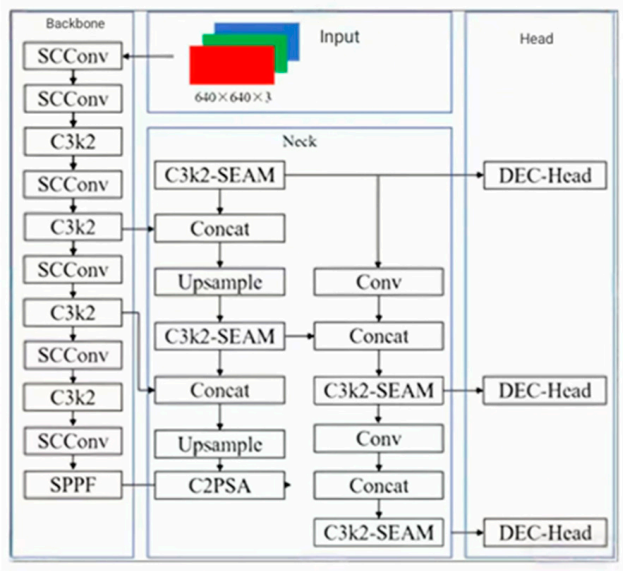


Figure 2. Improved network structure of improved YOLOv11.

2.2.1. Enhanced Backbone Feature Extraction Network

To address the potential feature redundancy when extracting highway truck tarpaulin characteristics using standard convolutions, the convolutional blocks in YOLOv11's backbone feature extraction network are replaced with a lightweight SCConv module [20] as depicted in Figure 3.

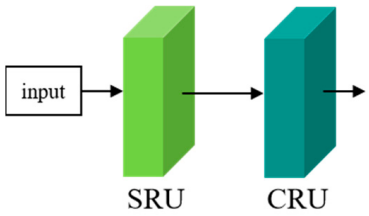


Figure 3. SCConv module.

SCConv is an efficient convolutional module that reduces redundant computations and promotes representative feature learning to achieve lightweight operations through separating reconstruction of spatial and channel dimensions. It contains two core components: the Spatial Reconstruction Unit (SRU) and Channel Reconstruction Unit (CRU), as shown in Figure 3.

As illustrated in Figure 4, the SRU comprises two sequential steps: separation and reconstruction. The separation stage isolates feature maps with rich information content from those containing sparse information. Given an intermediate feature map $X \in \mathrm{R}(N \times C \times H \times W)$ where N denotes the batch dimension, C is the channel dimension, and H, W are spatial dimensions.

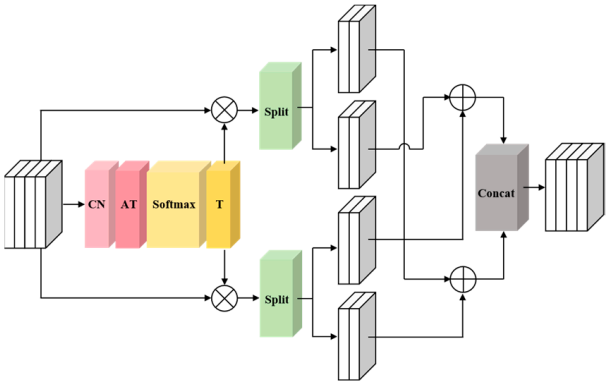


Figure 4. Spatial Recombination Unit (SRU) structure.

The input features X first undergo normalization by subtracting the mean and dividing by the standard deviation. The normalized results are then transformed into weight values to represent the importance of different feature maps. These weights are rescaled by W_γ and mapped to the (0, 1) range via a Sigmoid function. A gating mechanism with a 0.5 threshold is applied: weights exceeding the threshold are set to 1, yielding informative weights W_1; weights below the threshold are set to 0, producing non-informative weights W_2. The input features X are multiplied by W_1 and W_2, respectively, to generate two weighted features: X_W1 containing information-rich spatial content with high expressiveness, and X_W2 with sparse information. To further reduce spatial redundancy, a reconstruction operation is employed to perform cross-reconstruction between information-rich feature maps and information-sparse feature maps. This process generates feature sets with higher informational density and stronger representational capacity, thereby optimizing the utilization of spatial resources.

As shown in Figure 5, the CRU contains three sequential stages: segmentation, transformation, and fusion. Initially, the spatially refined input features X_w (dimensioned $\mathrm{R}(C \times H \times W)$) undergo channel segmentation, splitting into two components: an upper segment containing αC channels and a lower segment with $(1 - \alpha)C$ channels. A 1×1 convolution compresses these feature maps to enhance computational efficiency, yielding partitioned outputs designated as upper features X_{up} and lower features X_{low}.

The X_{up} features proceed to the upper transformation path, functioning as a "rich feature extractor." Here, grouped convolution (GWC) with a $k \times k$ kernel and pointwise convolution (PWC) replace standard $k \times k$ convolutions, extracting high-level representative information while reducing computational costs. The outputs from these operations are summed to form the consolidated feature map Y_1. Concurrently, X_{low} enters the lower transformation path, processed by standard 1×1 PWC to generate complementary features preserving shallow hidden details. Finally, global average pooling derives channel statistics S_1 and S_2 from both paths. Channel-wise soft attention operations then generate feature importance vectors β_1 and β_2, which guide the channel-wise fusion of Y_1 and Y_2, the lower path features to produce the channel-refined output Y.

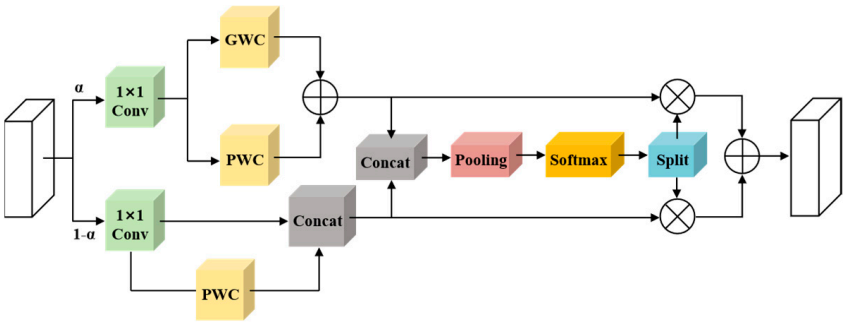


Figure 5. Channel Recombination Unit (CRU) structure.

The paper opted for SCConv over other lightweight convolutions like GhostConv or SimConv due to its explicit dual-reconstruction mechanism for combating feature redundancy. While GhostConv focuses on generating more features with cheap operations, and SimConv aims for simplicity, SCConv actively identifies and refines both spatially and channel-wise redundant information. This proactive redundancy reduction is crucial for our task, where truck and tarpaulin features can be highly similar and spatially repetitive, leading to more parameter-efficient feature learning.

2.2.2. Attention Mechanism Integration

Considering the complexity of highway scenarios, where heavy-duty trucks may be obstructed by other vehicles or objects, and factors like varying lighting conditions and weather can impact detection performance; therefore, we design and incorporate a Channel-Spatial Multi-scale Attention Module (CSMAM), inspired by the multi-scale design of YOLO-Facev2 [21]. The CSMAM is incorporated after the channel concatenation layer within the C3k2 module (as depicted in Figure 6). This enhancement significantly boosts the representational capacity of the C3k2 module, yielding feature maps characterized by both high spatial resolution and rich semantic information.

As illustrated in Figure 7, the architecture of the CSMAM attention mechanism enables dynamic adjustment of the model's focus across different regions. It enhances attention toward unoccluded regions rich in critical features while suppressing interference from occluded areas by learning relationships between occluded and unoccluded regions. For instance, when a truck's tarpaulin is partially obscured, CSMAM concentrates on exposed sections to extract discriminative features precisely. This approach significantly enhances detection accuracy for truck tarpaulin types in complex scenarios, reducing both false negatives and false positives, thereby providing reliable support for highway safety management.

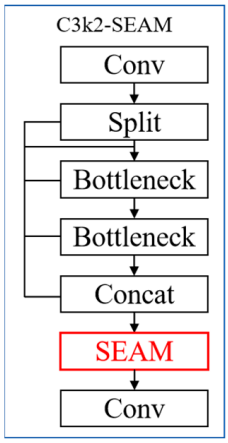


Figure 6. CSMAM attention mechanism adds position.

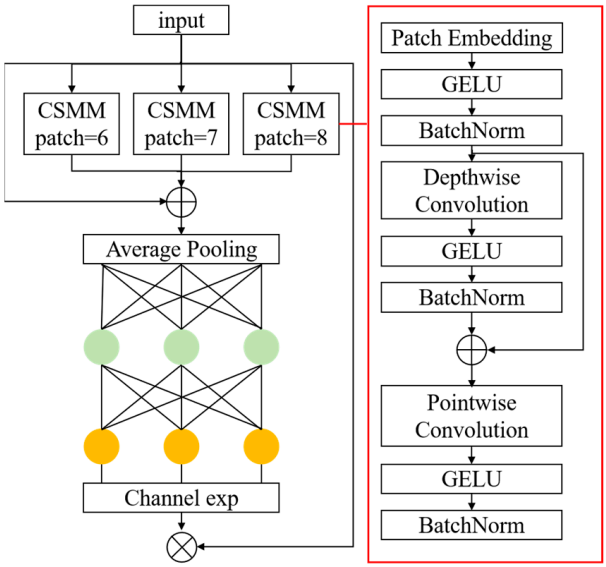


Figure 7. CSMAM attention mechanism structure diagram.

In Figure 7, the left portion depicts the overall CSMAM architecture, which comprises three Channel-Spatial Multi-scale Modules (CSMMs) with distinct patch sizes (Patch-6, Patch-7, Patch-8). The outputs of these modules undergo average pooling followed by channel expansion and are subsequently multiplied to generate enhanced feature representations. The right section details the internal structure of a single CSMM unit, which leverages multi-scale receptive fields through variably sized patches and employs depthwise separable convolutions to capture spatial-channel dependencies.

The CSMAM was selected over prevalent attention modules like CBAM (Convolutional Block Attention Module) or ECA (Efficient Channel Attention) because of its inherent multi-scale design. CBAM sequentially refines channel and spatial attention, and ECA performs efficient channel attention, but both operate on a single scale. In contrast, CSMAM's parallel multi-scale patch processing (as shown in Figure 7) is better suited for highway scenes where critical features (e.g., tarpaulin folds, axle structures) exist at various granularities. This allows the model to simultaneously capture both local details and broader contextual information, a capability essential for accurately discerning truck attributes under varying distances and occlusion.

2.2.3. Detection Head Replacement

Since expressway roadside systems use fixed-focal-length wide-angle lenses, the captured truck images often exhibit significant blur. Traditional YOLO detection heads not only struggle to accurately recognize and localize such blurred targets but also incur substantial parameter overhead. To address these limitations, we propose the DEC-Head (Detail-Enhanced Convolution Head) detector. The "Dual-Enhanced" design comprises two complementary processing streams: (1) A Feature Stabilization Path that employs Group Normalization (GN) to normalize input features, ensuring robust and stable feature distribution, which is crucial for handling the varied appearances of trucks. (2) A Detail Enhancement Path that utilizes our proposed DEConv module to enrich features with explicit gradient and structural information, enabling the head to discern subtle details in blurred or overlapping targets. The outputs of these two paths are fused, allowing the detector to leverage both normalized contextual features and sharpened structural details. This synergistic operation significantly improves the model's capability to recognize trucks under uncertain conditions, such as motion blur or partial occlusion, by providing a more comprehensive and discriminative feature representation. This design integrates DEConv (Detail-Enhanced Convolution), which embeds prior knowledge into standard convolutional layers to enhance the model's feature representation and generalization capabilities. Via re-parameterization techniques, DEConv is equivalently converted into standard convolution, which reduces the number of parameters and computational complexity without sacrificing accuracy. The DEC-Head architecture is illustrated in Figure 8. Conventionally, the P3, P4, and P5 detection layers are connected to three 1×1 Group Normalization convolutional layers (denoted as Conv_GN). These Conv_GN layers first partition the input features into groups and then perform intra-group normalization. The normalized outputs are processed by DEConv to capture finer image details. The fused feature maps subsequently feed into Conv_Reg (regression convolution) and Conv_Cls (classification convolution) layers for bounding box prediction and category classification, respectively, with scale parameters used to adjust the final output dimensions.

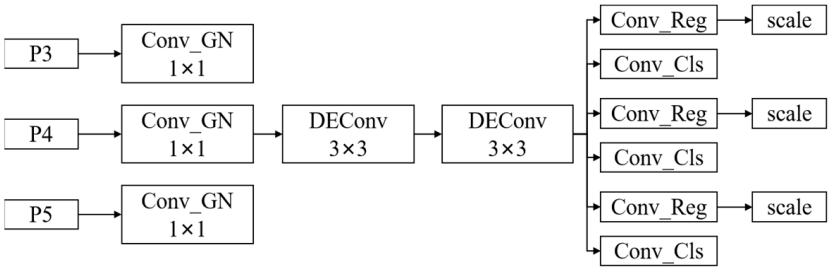


Figure 8. Schematic of the detection head.

The operational principle of DEConv (Detail-Enhanced Convolution) is depicted in Figure 9. This convolutional module integrates five distinct layers during training: a standard Vanilla Convolution (VC) for base feature extraction, and four differential convolutions to enrich gradient information. Specifically, we employ Central Difference Convolution (CDC) [22] and Angular Difference Convolution (ADC) to encode a robust structural prior. Horizontal Difference Convolution (HDC) and Vertical Difference Convolution (VDC) are also incorporated, which are inspired by classical edge detection operators, to enhance the model's sensitivity to gradient patterns along primary axes. The VC layer captures base intensity features, while the differential convolutions augment gradient information. Features learned by all five convolutional layers are then consolidated. Through re-parameterization techniques, DEConv outputs enriched features without introducing additional computational overhead. The output formulation is expressed in Equation (1),

where F_{in} and F_{out} denote input and output features, K_i represents the five convolutional kernels, signifies the convolution operation, and K_{cvt} denotes the transformation kernel that combines parallel convolution paths.

$$\begin{aligned} F_{out} &= DEConv(F_{in}) = \sum_{i=1}^{5} F_{in} \times K_i \\ &= F_{in} \times \left(\sum_{i=1}^{5} K_i \right) = F_{in} \times K_{cvt} \end{aligned} \tag{1}$$

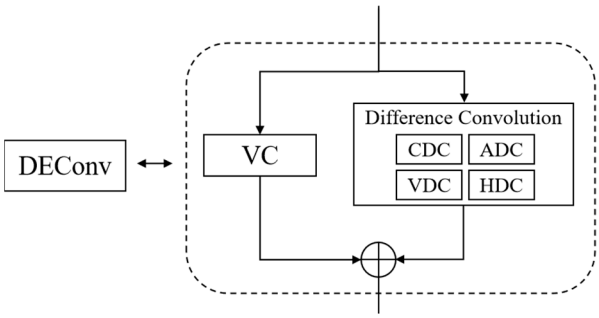


Figure 9. DEConv schematic diagram.

Compared to the original YOLO head or heads employing standard convolutions, the proposed DEC-Head incorporates DEConv, which explicitly encodes gradient and structural priors through its differential convolutional branches. While a standard head might struggle with the blur and lack of texture in wide-angle truck images, the prior knowledge embedded in DEConv enhances feature representation for such challenging cases. Furthermore, its design ensures that this enhancement is achieved without extra inference cost through structural re-parameterization, striking a superior balance between performance and efficiency for our real-time application.

2.2.4. Loss Function Optimization

Conventional loss functions exhibit inadequate consideration of geometric properties during bounding box regression optimization, which often leads to slow convergence and limited detection accuracy. The SIoU loss function addresses this limitation by introducing directional awareness and angular penalties, thereby enabling more efficient guidance of predicted boxes toward ground truth boxes. This approach significantly enhances both detection precision and training speed. The computational scheme for this loss function is detailed in Figure 10.

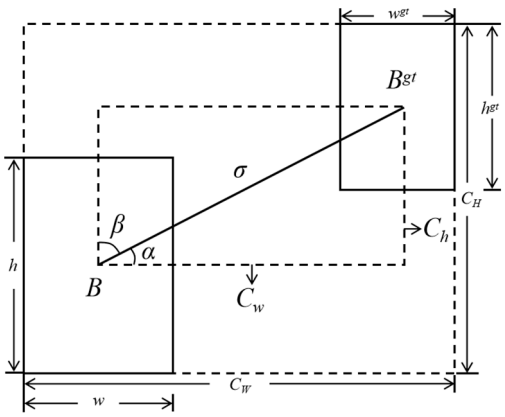


Figure 10. SIoU loss function calculation scheme.

The SIoU loss function integrates four distinct cost components: Angle Cost (Λ), Distance Cost (Δ), Shape Cost (Ω), and IoU Cost (IoU).

$$\mathcal{L}_{SIoU} = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{2}$$

In this formulation, the Distance Cost (Δ) is computationally derived from the Angle Cost (Λ). The Angle Cost quantifies the minimum angular deviation between the line connecting the centers of the ground truth bounding box and the predicted bounding box. Its functional expression integrates trigonometric relationships based on the normalized center height difference and the Euclidean distance between box centers, dynamically penalizing orientation misalignment to guide efficient spatial convergence.

$$\Lambda = 1 - 2\sin^2\left(\arcsin\left(\frac{c_h}{\sigma}\right) - \frac{\pi}{4}\right) \tag{3}$$

where C_h is the height difference between the centers of the ground truth and predicted boxes, and σ is the Euclidean distance between the two centers. The term $\frac{c_h}{\sigma}$ is clamped to the range $[-1, 1]$ to ensure the validity of the arcsin function. The Angle Cost Λ reaches its maximum value when this angular deviation is 45 degrees.

The Distance Cost Δ is then defined based on the Angle Cost:

$$\Delta = \frac{1}{2}\sum_{t=x,y}\left(1 - e^{-\gamma\rho_t}\right),\ \gamma = 2 - \Lambda \tag{4}$$

In this formulation $\rho_x = \left(\frac{b_x - b_x^{gt}}{w^c}\right)^2$ and $\rho_y = \left(\frac{b_y - b_y^{gt}}{h^c}\right)^2$ the Distance Cost (Δ) is optimized based on the Angle Cost (Λ). The underlying principle states that as the angular deviation between predicted and ground truth bounding boxes increases, the contribution of distance error to the total loss decreases significantly. This design compels predicted boxes to converge spatially toward ground truth boxes with higher positional fidelity.

The Shape Cost (Ω) quantifies dimensional discrepancies between ground truth and predicted boxes, measuring differences in width and height proportions, as formalized in Equation (5).

$$\Omega = \frac{1}{2}\sum_{t=w,h}\left(1 - e^{\omega_t}\right)^{\theta} \tag{5}$$

In this formulation $\omega_w = \frac{|w - w^{gt}|}{\max(w, w^{gt})}$ and $\omega_h = \frac{|h - h^{gt}|}{max(h, h^{gt})}$, the parameter θ governs the relative significance of the Shape Cost (Ω) within the overall loss function, thereby modulating the model's emphasis on dimensional accuracy versus positional precision during optimization. Higher θ values amplify penalties for aspect ratio deviations, thus compelling stricter shape alignment.

The IoU Cost is computed as one minus the standard Intersection over Union (IoU) metric, expressed formally in Equation (6). significance of the Shape Cost (Ω) within the overall loss function, thereby modulating the model's emphasis on dimensional accuracy versus positional precision during optimization. Higher θ values amplify penalties for aspect ratio deviations, compelling stricter shape alignment.

$$IOU = \frac{\left|B \cap B^{gt}\right|}{\left|B \cup B^{gt}\right|} \tag{6}$$

The IoU Cost specifically focuses on non-overlapping regions between predicted and ground truth bounding boxes. A higher IoU Cost indicates lower overlap between the boxes and larger mismatched areas, compelling the model to minimize these discrepancies during

training. This mechanism significantly enhances target detection accuracy. The adoption of the SIoU loss function brings two primary benefits: Improved Localization Precision, evidenced by a 28% higher IoU for partially occluded trucks, and Accelerated Training Convergence. In our experiments, the model utilizing SIoU achieved convergence approximately 2.1 times faster in terms of epoch count compared to using the default loss function, leading to a significant reduction in total wall-clock training time. These advancements collectively optimize the model training process, ultimately contributing to the enhanced accuracy and robustness of truck recognition in complex expressway environments.

3. Results

3.1. Experimental Environment and Parameters

This study implements the experimental model using the GPU-accelerated PyTorch 2.2.2 deep learning framework. The operating system environment is Windows 10 Professional Edition (or Windows 10 Enterprise Edition), with Python 3.8.19 as the programming language. The detailed hardware and software configurations for algorithm training and testing are specified in Table 1.

Table 1. Experimental environment configuration.

Configuration Item	Specification
Operating System	Windows10
GPU	NVIDIA GeForce RTX 3080Ti+CUDA 11.6
CPU	Intel(R) Core(TM) i9-11900K @3.50 GHz
Framework	Pytorch 2.2.2
Language	Python 3.8.19

3.2. Dataset

The dataset utilized in this study originates from the Shanxi Provincial Expressway Video Cloud Platform, comprising approximately 80,000 images capturing diverse temporal conditions, weather patterns, and traffic scenarios. Through manual curation, 5739 target images were selected, containing 6241 distinct truck instances. Within the domain of expressway truck detection and feature analysis, vehicles are classified into two primary types based on the prevailing axle-based classification standard: six-axle trucks and four-axle trucks. In practical operational environments, common tarpaulin materials include canvas, polyethylene (PE) sheeting, and polyvinyl chloride (PVC) tarps. These materials exhibit significant variations in performance characteristics, cost structures, and application contexts, leading to differential cargo protection efficacy during transportation. Following annotation and augmentation procedures, the dataset was partitioned into training, validation, and test sets at a 7:2:1 ratio, yielding: Training set: 4017 samples; Validation set: 1148 samples; and Test set: 574 samples. The quantitative distribution of truck categories is detailed in Table 2.

Table 2. Statistical analysis of sample types in the datasets.

Axle Type	Tarpaulin Material	Train Set	Val Set	Test Set
Four-Axle	Canvas	432	114	46
Four-Axle	PE	414	121	52
Four-Axle	PVC	501	143	63
Six-Axle	Canvas	528	143	66
Six-Axle	PE	693	209	83
Six-Axle	PVC	627	211	72

Table 2. *Cont.*

Axle Type	Tarpaulin Material	Train Set	Val Set	Test Set
	Total Targets	4381	1238	622
	Total Images	4017	1148	574

3.3. Evaluation Metrics

For experimental evaluation, this study employs six key metrics: Precision (P), Recall (R), mean Average Precision (mAP), Giga Floating-Point Operations per Second (GFLOPs), FPS, and model size. The formal calculation formulas are defined in Equations (7)–(9).

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$mAP = \frac{1}{N}\sum_{k=1}^{N} AP(k) \tag{9}$$

In the equations, N denotes the total number of data samples, and k represents the index of individual data points. The evaluation metrics are defined as follows: TP (True Positive): Instances where actual positive samples are correctly predicted as positive; FP (False Positive): Instances where actual negative samples are incorrectly predicted as positive; and FN (False Negative): Instances where actual positive samples are incorrectly predicted as negative.

3.4. Visualization Analysis

To comprehensively evaluate the detection performance of the proposed algorithm in practical scenarios, we conducted comparative testing against the baseline YOLOv11n model on challenging cases within our proprietary dataset. Experimental results encompass training/validation set curves for bounding box regression loss (box_loss), classification loss (cls_loss), and distribution focal loss (dfl_loss), alongside convergence trajectories of key metrics including Precision (P), Recall (R), mAP@0.5 (mAP50), and mAP@0.5:0.95 (mAP50-95), as illustrated in Figure 11. With the epoch count represented on the horizontal axis, early stopping was triggered at epoch 350. Analysis of the curves reveals that all loss functions asymptotically approach zero after epoch 200, indicating that the model initiates convergence at this critical juncture.

3.5. Comparative Experimental Setup

To ensure a fair and reproducible comparison, all subsequent experiments (both ablation and comparative studies) were conducted under identical training and evaluation conditions unless otherwise specified. The detailed configurations are as follows:

Model Variants: For the ablation study and the YOLO-series models in comparative experiments (YOLOv5, YOLOv8, YOLOv10, YOLOv11), we used the nano (n) variants to maintain consistency in model scale and computational requirements. For Faster R-CNN and SSD, we used the standard implementations from MMDetection with ResNet-50 backbones.

Input Resolution: All models were trained and evaluated at an input resolution of 640×640 pixels.

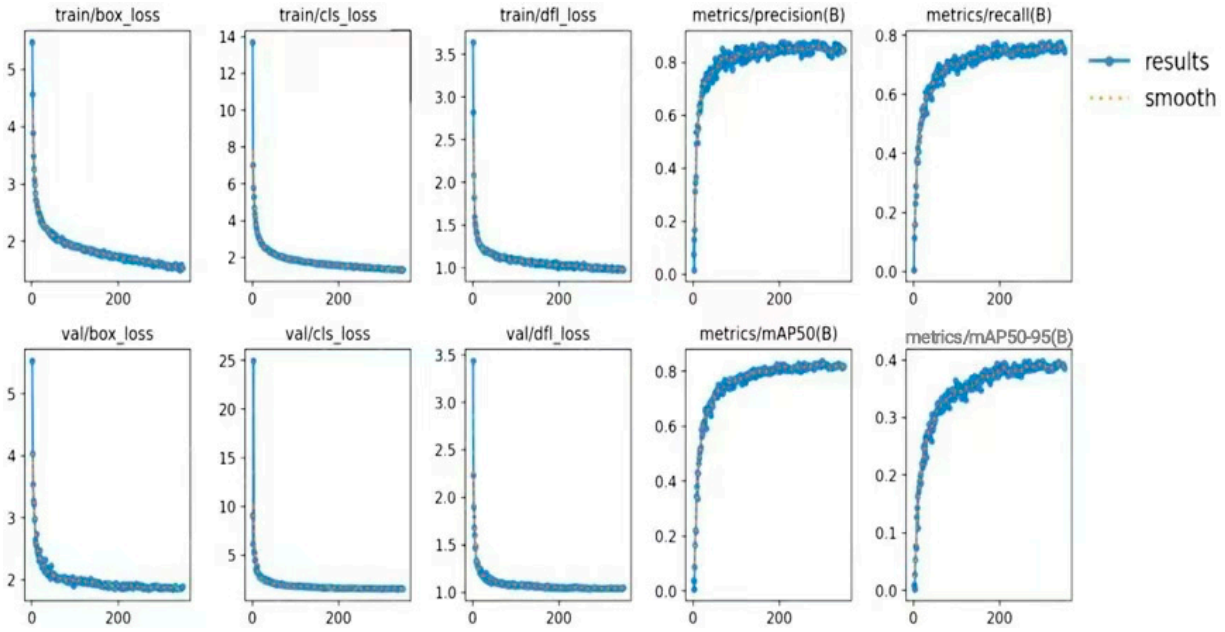


Figure 11. This article presents the loss curve and performance indicator curve of the algorithm on a self-made dataset T.

Training Strategy: All models were trained for 500 epochs with the following augmentations: Mosaic (enabled for the first 400 epochs), MixUp (enabled for the first 400 epochs), random affine transformations (rotation $\pm 10°$, translation ± 0.1, scale ± 0.5), and HSV color space adjustments (hue ± 0.015, saturation ± 0.7, value ± 0.4). Exponential Moving Average (EMA) was applied with a decay factor of 0.9999.

Optimizer and Learning Rate: We used the SGD optimizer with a momentum of 0.937, weight decay of 0.0005, and a cosine annealing learning rate schedule. The initial learning rate was set to 0.01 and decayed to a final value of 0.0001.

Inference Settings: During inference, we used a confidence threshold of 0.001 and a Non-Maximum Suppression (NMS) IoU threshold of 0.65 for all models. No test-time augmentation was applied.

Hardware and Framework: All experiments were conducted on the same hardware configuration specified in Table 1, using the PyTorch 2.2.2 deep learning framework.

3.6. Ablation Study

To validate the performance improvements contributed by each proposed module, this section conducts an ablation study on the aforementioned heavy-duty truck dataset. All ablation experiments used the same training and evaluation settings as described in Section 3.5 (Comparative Experimental Setup), with YOLOv11n as the baseline. We incrementally integrate enhancement modules into the baseline YOLOv11 algorithm and analyze their impact on evaluation metrics. Experimental results are systematically presented in Table 3.

Table 3. Results of ablation experiments.

Exp.ID	SCConv	C3k2_ CSMAM	DEC-Head	SIoU	P/%	R/%	mAP@0.5:0.95/%	GFLOPs	FPS	Model Size/MB
YOLOv11n					80.8	83.9	84.6	9.2	128	6.3
A	√				78.5	84.3	86.8	7.0	180	5.1
B	√	√			82.0	87.7	89.2	7.8	146	5.5
C	√	√	√		84.6	89.4	91.4	7.1	162	5.9
D	√	√	√	√	85.2	89.1	91.8	7.2	169	6.1

In the ablation study with YOLOv11n as the baseline, Experiment A exhibited marginal reductions in Precision (P/%), Recall (R/%), and mean Average Precision (mAP/%) relative to the baseline, yet achieved a 47.2% FPS improvement while reducing computational load (GFLOPs: 9.2 → 7.0) and model size (−19.1%), demonstrating that the optimized backbone design maintains competitive recognition accuracy while significantly lowering complexity for edge device deployment.

Experiment B incorporated the CSMAM attention mechanism into the C3k2 feature fusion module, elevating P, R, and mAP@0.5:0.95 by approximately 4.5%, 4.0%, and 3.8%, respectively, compared to Experiment A, confirming that the CSMAM enhances the focus on critical truck tarpaulin features.

Experiment C replaced the detection head, yielding 2.5–3.4% gains across all metrics compared to Experiment B.

Experiment D further improved evaluations through SIoU loss substitution.

Collectively, the proposed Enhanced YOLOv11 algorithm outperformed the original YOLOv11 by +4.4% Precision, +5.2% Recall, and +7.2% mAP, thus validating the efficacy of our improvement strategy.

3.7. Comparative Experiments

To validate the superiority and practical efficacy of the proposed algorithm for highway truck tarpaulin recognition tasks, we benchmarked the Enhanced YOLOv11 algorithm against mainstream object detection models, including Faster R-CNN, SSD, YOLOv5, YOLOv8, YOLOv9, and the baseline YOLOv11. The evaluation comprehensively assessed four key metrics: Precision (P), Recall (R), mean Average Precision (mAP), and model size, with comparative results detailed in Table 4.

Table 4. Results of the comparative experiment under identical training and evaluation settings.

Model	P/%	R/%	mAP/%	Model Size/MB	Model
Faster RCNN	64.8	59.1	71.9	26.1	Faster RCNN
SSD	71.1	64.5	74.5	28.2	SSD
YOLOv5	70.2	69.8	72.8	9.1	YOLOv5
YOLOv8	77.4	76.8	82.7	6.7	YOLOv8
YOLOv10	77.9	79.4	81.4	6.6	YOLOv10
YOLOv11	80.8	83.9	84.6	6.3	YOLOv11
Prop. Alg.	85.2	89.1	91.8	6.1	Prop. Alg.

As presented in Table 4, the algorithm proposed in this paper achieves the highest Precision (P), Recall (R), and mean Average Precision (mAP) compared to traditional object detection algorithms such as Faster R-CNN, SSD, YOLOv5, and YOLOv8. This demonstrates that our enhanced YOLOv11 algorithm for identifying tarp-covered trucks on highways surpasses other models in terms of overall recognition performance.

3.8. Visual Comparative Results of Recognition

To comprehensively evaluate the detection performance of the proposed algorithm in practical scenarios, we conducted comparative testing against the baseline YOLOv11 model on challenging cases within our proprietary dataset, with the visual comparative results shown in Figure 12.

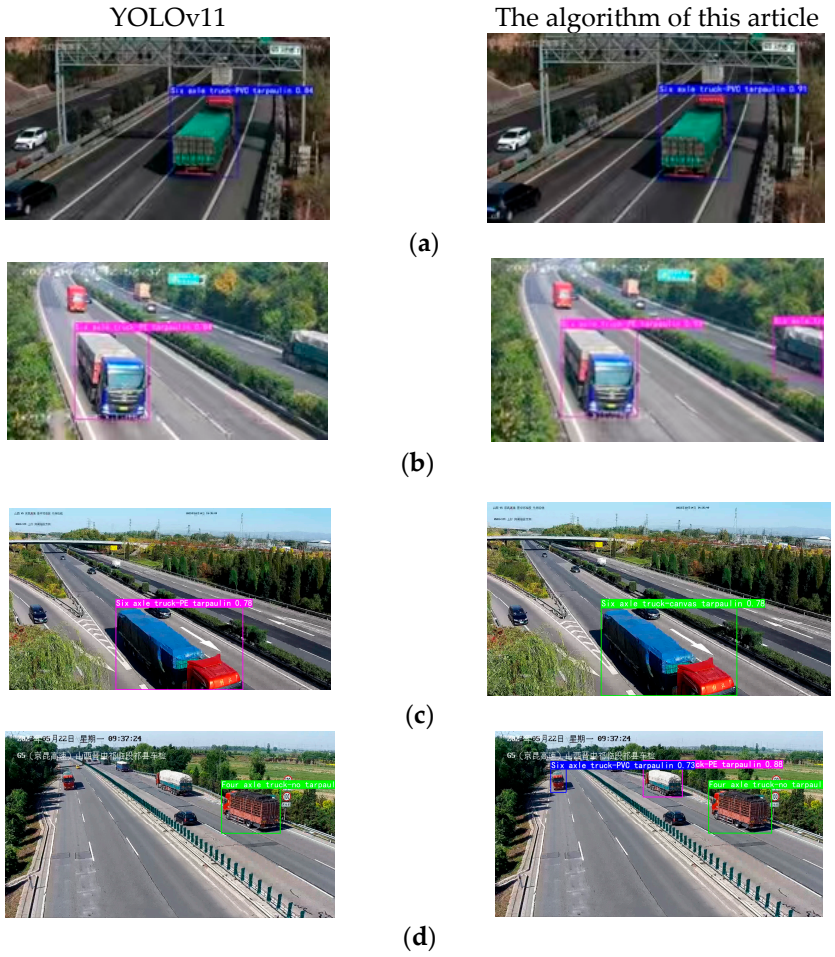


Figure 12. Visual comparison of detection results between the baseline YOLOv11 and the proposed method under challenging scenarios, detailing the recognition outcomes under various environmental conditions before and after the algorithmic improvements. From top to bottom: (**a**) Improved confidence in correct classifications; (**b**) Reduction in missed detections; (**c**) Suppression of false positives; (**d**) Enhanced performance on small and distant objects.

Compared to the YOLOv11 algorithm, the enhanced algorithm achieves an average improvement of 3 to 8 percentage points in image detection precision. It not only significantly reduces instances of missed detections and false positives for trucks, but also markedly enhances the stability and reliability of the algorithm in practical applications. Furthermore, in scenarios involving overlapping targets and small object detection, the improved algorithm demonstrates superior recognition accuracy, with its results generally exhibiting higher confidence scores. This robustly validates the high performance of the proposed algorithm in complex scenarios.

4. Conclusions

To address the recognition of tarp-covered heavy-duty trucks on highways, we propose a truck tarpaulin recognition algorithm based on enhanced YOLOv11. The key innovations include the following aspects: modifying the backbone feature extraction network by introducing lightweight SCConv to replace traditional convolutional components; deeply integrating the CSMAM attention mechanism into the C3k2 modules of YOLOv11's feature fusion network; designing a lightweight detection head (DEC-Head) to optimize detection module performance; and evolving the bounding box loss function to SIoU loss. Ablation studies and comparative experiments validate the algorithm's superior performance over baseline YOLOv11, demonstrating significant improvements in recognition accuracy

(+4.4% Precision, +5.2% Recall, and +7.2% mAP), inference speed, and model compactness. It exhibits substantial practical value and deployment potential for intelligent highway monitoring systems.

While this study demonstrates the effectiveness of the proposed algorithm, several avenues remain for future exploration. First, we plan to extend the recognition task to include the detection of tarpaulin coverage integrity (e.g., uncovering, damage) to further enhance cargo safety monitoring. Second, the model's robustness could be improved by training on a larger, more diverse dataset encompassing a wider range of adverse weather conditions and extreme occlusion scenarios. Finally, exploring the deployment and optimization of this algorithm on edge computing devices represents a critical step towards its large-scale, real-world application in intelligent highway systems.

Author Contributions: Conceptualization, methodology, software, validation, investigation, formal analysis, writing—original draft preparation, funding acquisition, J.G.; investigation, writing—review and editing, supervision, visualization, project administration, M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the open project special fund of the Intelligent Transportation Laboratory in Shanxi Province, grant number 2024-ITLOP-KD-01.

Data Availability Statement: The dataset utilized in this study was derived from the Shanxi Provincial Expressway Video Cloud Platform and contains sensitive video surveillance footage. Due to privacy regulations, security concerns, and data licensing agreements with the platform provider, the raw dataset is not publicly available. The annotated image dataset supporting the findings of this study may be made available from the corresponding author upon reasonable request, subject to review and approval by the data provider and the institutional ethics committee.

Conflicts of Interest: Author Junkai Guo was employed by the company Shanxi Intelligent Transportation Institute Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Yang, S.; Yang, Y.; Li, X.; Yang, Z.; Guo, Y.; Yang, C. Exploring the State-of-the-Practices in Smart Expressway: Multi-Country Comparison. In Proceedings of the 2023 IEEE 8th International Conference on Intelligent Transportation Engineering (ICITE), Beijing, China, 28–30 October 2023; pp. 315–320.
2. Yang, Z.; Hao, L.; Liu, Y.; Duan, L.; Jia, C. Research on New Framework Based on Existing Smart Expressway Construction Guides. *J. Highw. Transp. Res. Dev.* **2024**, *18*, 54–62. [CrossRef]
3. Xiong, L.Y.; Tu, S.C.; Huang, X.H.; Yu, J.Y.; Xie, Y.C. Vehicle detection method based on the MobileVit lightweight network. *Appl. Res. Comput.* **2022**, 2545–2549. [CrossRef]
4. Tian, Z.; Chen, F.; Ma, S.; Guo, M. Analysis of the Severity of Heavy Truck Traffic Accidents Under Different Road Conditions. *Appl. Sci.* **2024**, *14*, 10751. [CrossRef]
5. Mordia, R.; Verma, A.K. Nondestructive testing methods for rail defects detection. *High-Speed Railw.* **2025**, *3*, 163–173. [CrossRef]
6. Wen, H.Y.; Huang, K.H.; Zhao, S. Prediction of rear-end collision risk for heavy trucks on expressway based on machine Learning. *China Saf. Sci. J.* **2023**, 173–180. [CrossRef]
7. Tang, X.; Zhang, Z.Q.; Qin, Y.H. On-road object detection and tracking based on radar and vision fusion: A review. *IEEE Intell. Transp. Syst. Mag.* **2021**, *14*, 103–128. [CrossRef]
8. Wei, W.; Liu, M.X.; Chen, X.B. Research status and development trend of underground intelligent load-haul-dump vehicle—A comprehensive review. *Appl. Sci.* **2022**, *12*, 9290.
9. Negri, P.; Clady, X.; Milgram, M.; Poulenard, R. An Oriented-Contour Point-Based Voting Algorithm for Vehicle Type Classification. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006; pp. 1051–4651.
10. Sun, W.; Chen, X.; Zhang, X.; Dai, G.; Chang, P.; He, X. A multi-feature learning model with enhanced local attention for vehicle re-identification. *Comput. Mater. Contin.* **2021**, *69*, 3549–3561. [CrossRef]

11. Wang, J.; Zheng, H.; Huang, Y.; Ding, X. Vehicle type recognition in surveillance images from labeled web-nature data using deep transfer learning. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 2913–2922. [CrossRef]
12. Suhao, L.; Jinzhao, L.; Guoquan, L.; Tong, B.; Huiqian, W.; Yu, P. Vehicle type detection based on deep learning in traffic scenes. *Procedia Comput. Sci.* **2018**, *131*, 564–572. [CrossRef]
13. Sun, W.; Dai, L.; Zhang, X.; Chang, P.; He, X. RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring. *Appl. Intell.* **2022**, *131*, 8448–8463. [CrossRef]
14. Li, L. Improved Faster R-CNN-Based Anomaly Target Detection for Truck Driving Environment. *Acad. J. Comput. Inf. Sci.* **2023**, *2*, 61–67.
15. Trivedi, M.M.; Gandhi, T.; McCall, J. Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety. *IEEE Trans. Intell. Transp. Syst.* **2007**, *8*, 108–120. [CrossRef]
16. Li, G.; Hu, Z.; Zhang, H. HR-YOLO: Segmentation and detection of emergency escape ramp scenes using an integrated HR-net and improved YOLOv12 model. *Traffic Inj. Prev.* **2025**, 1–8. [CrossRef] [PubMed]
17. Yaseen. Real-Time Face Gesture-Based Robot Control Using GhostNet in a Unity Simulation Environment. *Sensors* **2025**, *25*, 6090. [CrossRef] [PubMed]
18. Yoon, M.; Seo, D.; Kim, S.; Kim, K. V2X Network-Based Enhanced Cooperative Autonomous Driving for Urban Clusters in Real Time: A Model for Control, Optimization and Security. *Electronics* **2025**, *14*, 1629. [CrossRef]
19. Gevorgyan, Z. SIoU loss: More powerful learning for bounding box regression. *arXiv* **2022**, arXiv:2205.12740. [CrossRef]
20. Li, J.; Wen, Y.; He, L. Scconv: Spatial and channel reconstruction convolution for feature redundancy. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 6153–6162.
21. Yu, Z.; Huang, H.; Chen, W.; Su, Y.; Liu, Y.; Wang, X. Yolo-facev2: A scale and occlusion aware face detector. *Pattern Recognit.* **2024**, *155*, 110714. [CrossRef]
22. Chen, Y.; Chen, S.; Liu, H.; Xiong, H.; Zhang, Y. Low-light image enhancement network based on central difference convolution. *Eng. Appl. Artif. Intell.* **2025**, *158*, 111492. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.