Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Design of an Efficient Object Detection Algorithm for Traffic Accident Recognition

Yongping Zheng
Zhejiang University of Science and Technology
Dianzheng Xu
Zhejiang University of Science and Technology
Feng Li
Bingqiang Huang
Zhejiang University of Science and Technology
Haoyu Pei
Zhejiang University of Science and Technology
Zhihua Huang
Xinjian Xiang
hzxxj@sina.com

Zhejiang University of Science and Technology

Research Article

Keywords: Lightweight, ShuffleNetV2, DW_ADown, DS_SENetV2

Posted Date: July 7th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9638146/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Signal, Image and Video Processing on October 14th, 2026. See the published version at https://doi.org/10.1007/s11760-026-05760-1.

Design of an Efficient Object Detection Algorithm for Traffic Accident Recognition

Yongping Zheng[1], Dianzheng Xu[1], Feng Li[2], Bingqiang Huang[1], Haoyu Pei[1],,Zhihua Huang[2],Xinjian Xiang[1]

1School of Automation and Electrical Engineering, Zhejiang University of Science and Technology,Hangzhou 310023, China

2YONGKANG VALID TECHNOLOGY CORPORATION LIMITED, Jinhua 321300, China

Corresponding author: XINJIAN XIANG (e-mail: hzxxj@sina.com).

Abstract With the increasing traffic density and the growing complexity of urban traffic environments, existing traffic accident detection methods struggle to meet the dual demands of real-time performance and accuracy. Addressing the limitations of the YOLO11n-based models—particularly their low precision in challenging conditions such as occlusion and nighttime scenes, as well as the difficulty of deploying them efficiently on edge devices like in-vehicle terminals and roadside units—this paper proposes a lightweight improved model named SDD-YOLO11n. First, the model introduces a ShuffleNetV2 backbone to enhance inference speed and accuracy while maintaining low computational complexity. Second, a DW_ADown module is incorporated into the downsampling layers to mitigate information loss and improve generalization ability. Finally, the original model's SPPF module is replaced with DS_SENetV2 to strengthen the network's capacity to capture channel patterns and global contextual knowledge, thus achieving better feature representation. Experimental results demonstrate that compared to the original YOLO11n model, SDD-YOLO11n improves mAP@0.5 by 1.3%, reduces parameters by 69.1%, lowers FLOPs by 65.6%, and still maintains a frame rate of 70.9 FPS, indicating an excellent balance between detection accuracy and computational efficiency. Compared with mainstream models such as YOLOv8 and YOLOv9, the proposed model maintains superior performance under lightweight conditions and meets real-time detection requirements.

Index terms Lightweight; ShuffleNetV2; DW_ADown; DS_SENetV2

1 Introduction

With the annual increase in the number of motor vehicles, the demand for road traffic continues to grow, resulting in a rising frequency of traffic accidents. Traditional manual reporting methods heavily rely on the initiative and judgment of eyewitnesses or individuals involved in the incident, often leading to delayed reports, inaccurate information, or even missed reports. In some cases, injured individuals may be unconscious and unable to call for help, and accidents occurring at night or in remote areas are more likely to be overlooked. Moreover, manual reporting typically takes several minutes to reach the command center after the incident, missing the critical "golden rescue time[1]. "To improve the efficiency of traffic accident detection, this paper conducts a study on traffic accident detection techniques.

In traditional traffic accident detection research, Liu proposed an RCNN-based rollover detection algorithm that identifies rollover accidents by analyzing spatial relationships among vehicles[2]. However, its detection accuracy decreases in conditions of dense traffic flow or adverse weather. Zhu et al. developed a model based on Learning Vector Quantization (LVQ), integrating meteorological factors to enhance detection accuracy[3]. Although innovative in accounting for weather influences, this method still struggles with accuracy in scenarios involving heavy traffic occlusion.

In response to the challenges posed by adverse weather, heavy traffic occlusion, and real-time detection requirements in traffic accident detection, researchers have proposed various innovative methods in recent years. Jingru Yang proposed a rotation vehicle detection algorithm based on an

improved YOLO combined with an enhanced StrongSORT for object tracking, achieving accident vehicle detection and tracking[4-6]; however, due to large parameter and computational complexity, real-time operation is difficult.Wang Xinyu developed a YOLOv5-based vehicle collision detection algorithm that effectively identifies collisions but still faces accuracy limitations under complex weather conditions[7]. Hou Xiaoyan proposed an improved YOLOv5 model that balances detection accuracy and speed, allowing users to choose appropriate versions based on deployment needs for accident detection[8]. Chen et al. introduced an accident detection and liability determination method based on YOLOv8 and GPT-4V, yet the model still lacks lightweight efficiency and real-time performance in complex scenarios[9]. Although the system proposed by Luo et al. performs well at unsignalized intersections, it suffers from limited scene adaptability, reliance on fixed camera layouts, vulnerability in speed estimation, lack of validation on real accident scenarios, and high deployment costs, which hinder its applicability in complex environments[10]. Hong Weitian et al. combined YOLOv5 with visual odometry to judge vehicle collisions, but the system remains constrained under nighttime conditions[11]. Tang Zifeng et al. enhanced the YOLOv8 algorithm to assess accident severity, but the increased algorithmic complexity raises concerns over deployment efficiency[12].

This paper proposes a lightweight traffic accident detection algorithm based on Yolo11n to address the problems of high computational cost and false/missed detections under occlusion and heavy traffic conditions. By introducing a ShuffleNetV2 backbone, the DW_ADown downsampling module, and the DS_SENetV2 feature enhancement module, the model effectively reduces parameter size and computational complexity while improving multi-scale perception and feature representation[13]. The improved algorithm demonstrates outstanding performance on the VADV dataset, significantly reducing the computational load and parameter count while maintaining high detection accuracy.

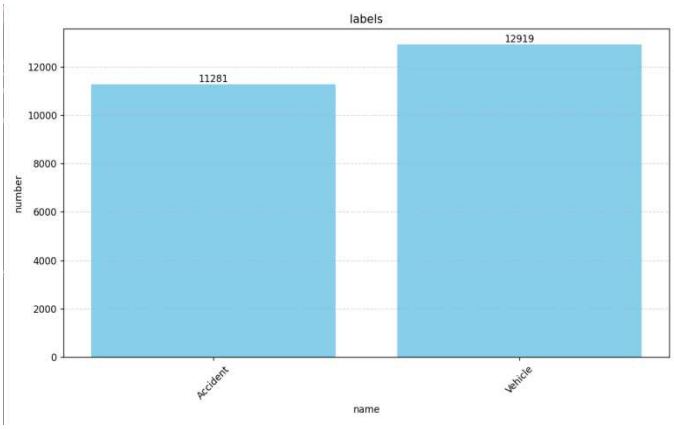


FIGURE 1. Distribution of data labels in the dataset.

2 Related works

This study utilizes the publicly available VADV dataset, which contains traffic accident images captured in various scenarios, such as urban streets and rural roads. The dataset includes images of accidents involving cars, motorcycles, and other vehicles. After a thorough screening process, 7,562 images were retained. The dataset was further augmented with additional images depicting conditions such as rain, poor lighting, and dense traffic occlusion to address the challenges posed by complex backgrounds during accident detection. These newly added images were carefully annotated.

The final merged dataset consists of 9,787 images, categorized into two classes based on content: Accident for images depicting traffic accidents and Vehicle for images showing standard driving vehicles. All images were resized to a uniform size of [640×640]. The dataset was split into training, testing, and validation sets with a 7:2:1 ratio. The statistical information for each category in the dataset is shown in Fig. 1.

3 Overview of the YOLO11n Model

The Ultralytics team released YOLO11n in 2024 as a state-of-the-art model building on previous YOLO versions[14]. Despite its strong object detection performance and enhanced flexibility, YOLO11n struggles in complex traffic environments. Challenges such as occlusion, dense traffic, and motion blur often lead to false positives and missed detections, undermining its reliability in accident detection. Moreover, its complex architecture, large parameter count, and high computational demands limit real-time deployment on edge devices like in-vehicle

terminals and roadside units, reducing its practicality in real-world applications.

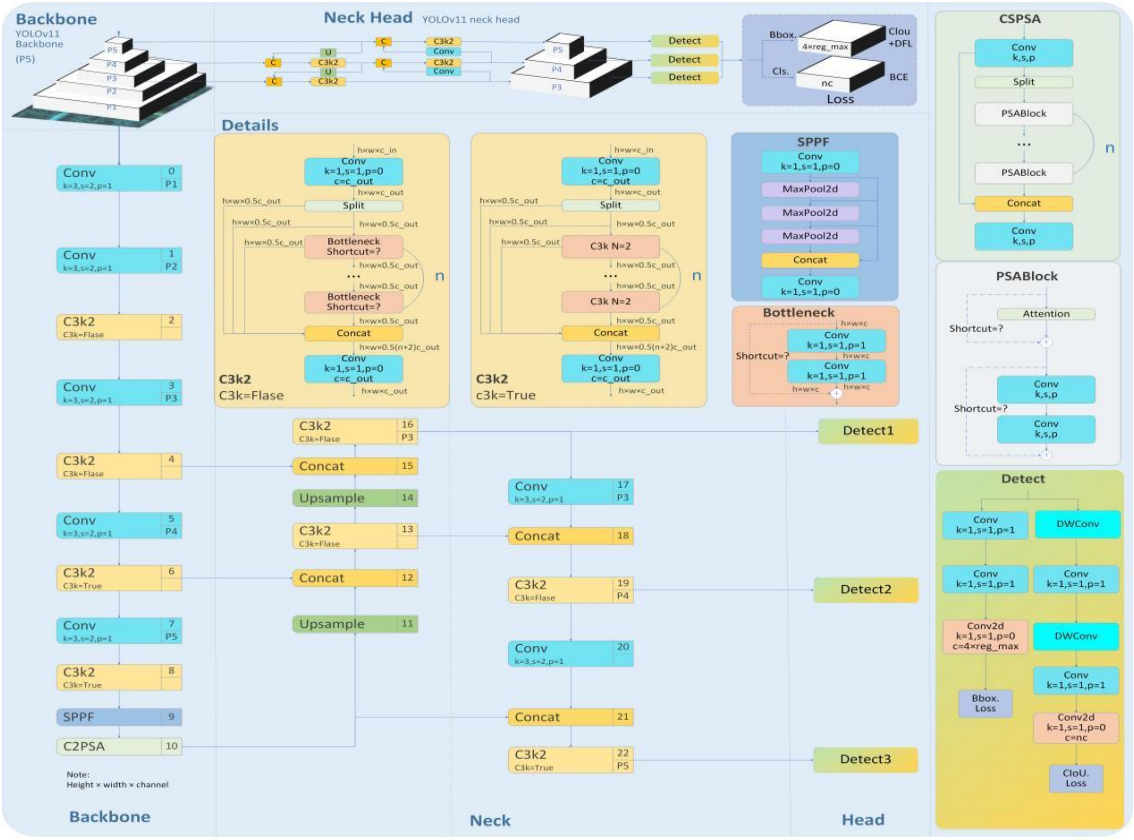


FIGURE 2. Structure of YOLOv11 network

4 Improved YOLO11n Detection Algorithm

This paper proposes an improved detection model, illustrated in Fig. 3, to address the limitations of the original YOLOv11n architecture—namely, its large parameter count, high computational cost, and poor adaptability to complex background interference.

First, ShuffleNetV2 is introduced as the backbone network to significantly improve inference speed and detection accuracy while maintaining low computational complexity[15]. Second, a novel downsampling module named DW_ADown is designed and incorporated into the downsampling stage to alleviate information loss and enhance the model's generalization ability. Finally, a lightweight DS_SENetV2 module replaces the original SPPF structure, enhancing the network's channel modeling and global feature extraction capabilities.

These modifications collectively contribute to a more efficient and robust traffic accident detection model suited for real-time deployment on edge devices.

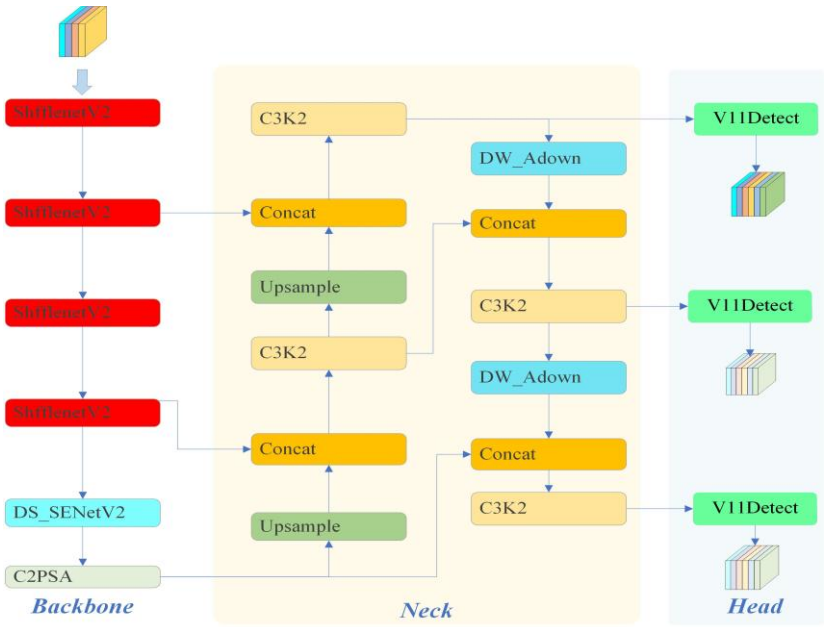


FIGURE 3. The architecture of the proposed SDD-YOLO11n network

4.1 ShuffleNetV2 Module

Considering the real-time requirements of traffic accident detection, this paper addresses the issues of the original YOLO11n backbone—namely, its structural complexity, large parameter size, and high computational resource consumption—by proposing a lightweight backbone network based on ShuffleNetV2. The goal is to significantly reduce the model's computational cost while maintaining or improving detection accuracy.

The proposed network design's core lies in introducing pointwise grouped convolution and channel shuffle mechanisms, which, through structural optimization, achieve a lightweight replacement for the complex backbone of YOLO11n.

Firstly, the improved network utilizes 1×1 pointwise grouped convolutions in the feature extraction module. The input feature maps are divided into subgroups and convolved separately[16], reducing parameter count and computational load. Compared to standard 1×1 convolution, pointwise grouped convolution maintains the same feature dimensionality while reducing computational complexity to 1/n of the original[17], where n is the number of groups. This significantly lowers the computational burden, making the network more suitable for real-time detection on mobile and edge devices.

Secondly, the channel shuffle operation is incorporated into each module to address the inter-group information blockage caused by grouped convolutions[18]. This operation rearranges the feature channels among different groups, enabling cross-group information exchange. This strategy retains the lightweight nature of grouped convolutions and enhances feature connectivity, improving the model's representation capability and detection accuracy.

Moreover, the proposed structure strictly follows Shufflenetv2's design principles, such as balanced computation, reduced memory access cost, and minimization of fragmented operations, ensuring the overall architecture is efficient and highly deployable in practical hardware environments[19].

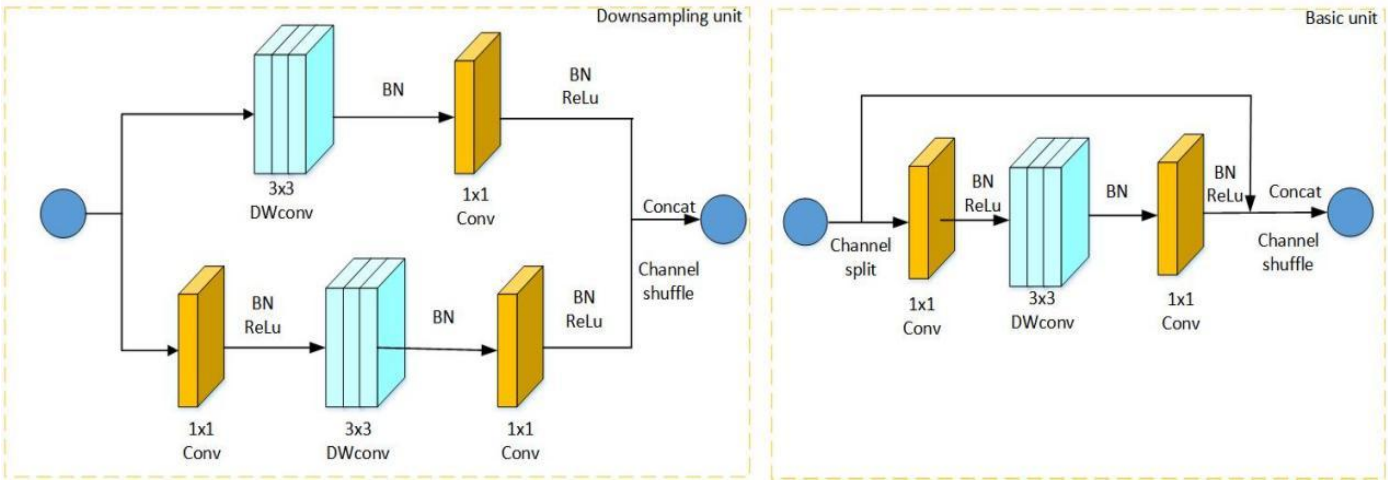


FIGURE 4. The architecture of the ShuffleNetV2 module

4.2 DW_ADown Module

The traditional convolutional downsampling layers used in the original YOLO11n model suffer from computational redundancy and feature loss. In traffic accident detection scenarios, this leads to significantly increased computational load and parameter size, imposing high hardware requirements that limit practical deployment in real-world traffic management systems.To address these issues, this paper proposes the DW_ADown module, illustrated in Fig. 4(a), which integrates the concepts of ADown and DWConv[20,21]. The key improvement is replacing the standard convolution in ADown with DWConv, substituting the original model's downsampling layer.The DWConv consists of a depthwise convolution followed by a pointwise convolution. The depthwise convolution extracts spatial features by applying independent 3×3 convolutions to each channel, while the pointwise convolution aggregates channel-wise information through 1×1 convolutions. Specifically, depthwise separable convolution first applies DWConv to each channel independently. Then, it uses a 1×1 pointwise convolution to merge all channels into the output feature map, effectively reducing computational cost and improving efficiency.

As depicted in Fig. 5, the 3×3 depthwise convolution kernel operates channel-wise with a small kernel size to preserve spatial feature extraction capability while significantly lowering computation. This design facilitates easier model convergence and mitigates overfitting risks. The 1×1 pointwise convolution mixes channel information by performing linear combinations across channels, allowing flexible control over network width and reducing computational demands. By combining these two convolution types, DWConv achieves efficient feature fusion and dimension adjustment, substantially decreasing parameter quantity.

To further mitigate information loss, the ADown downsampling module is introduced. This module significantly reduces the model's computational load while effectively balancing feature compression and information preservation[22]. The module first applies average pooling to the input image and then splits the feature map into two parts named x1 and x2. For x1, a 3×3 convolution with stride two is applied to perform downsampling and extract high-dimensional features. For x2, max pooling is first performed, followed by a 1×1 convolution for further processing[23]. Finally, the two branches are concatenated along the channel dimension, as detailed in Fig. 4(b).

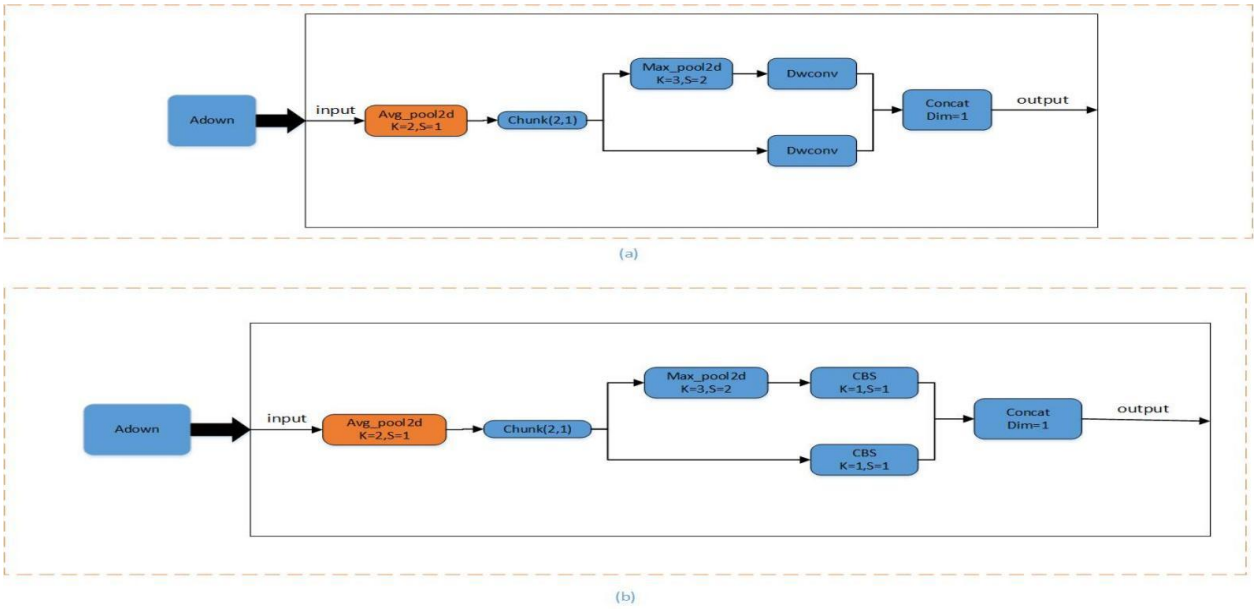


FIGURE 5. The architecture of the DS_Adown module

The Dw_adown module designed in this work integrates the above two components, significantly reducing computational complexity while preserving spatial feature extraction capabilities and minimizing information loss. This effectively alleviates the network's computational burden.

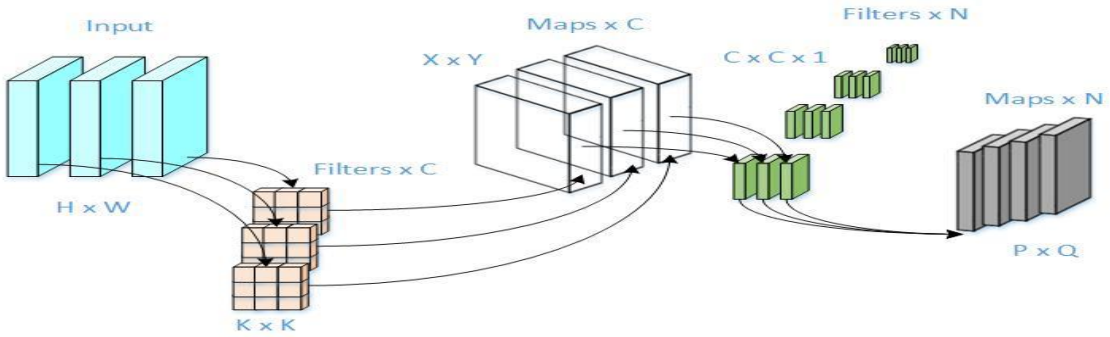


FIGURE 6. The architecture of the DWConv module

4.3 DS_SENetV2 Module

Scenarios for traffic accident detection often involve dense traffic and complex weather conditions, which can significantly disturb target feature patterns, degrade spatial localization, and reduce the overall performance of detection models. To address these challenges, this paper introduces the DS_SENetV2 module. The proposed module inherits the multi-scale feature aggregation concept from the original SPPF module, while integrating depthwise separable convolution (DWConv)[24] and the improved SENetV2 architecture[26]. This design effectively enhances the model ' s multi-scale information perception and feature representation capabilities, while significantly reducing computational complexity and parameter overhead.

As illustrated in Figure 7, the ResNeXt module adopts multi-branch convolutions to enhance feature diversity. In contrast, SENet computes channel-wise weights through global average pooling followed by fully connected layers. SENetV2 integrates the merits of both ResNeXt and SENet by embedding a multi-branch aggregation structure into the residual block to implement the squeeze-and-excitation mechanism, thereby improving feature refinement and global information integration. Based on this, the C2PSA-SENetV2 module is designed in the backbone network, which aims to enhance the model' s robustness under complex field conditions and strengthen its capability to distinguish soybean seedlings from weeds.

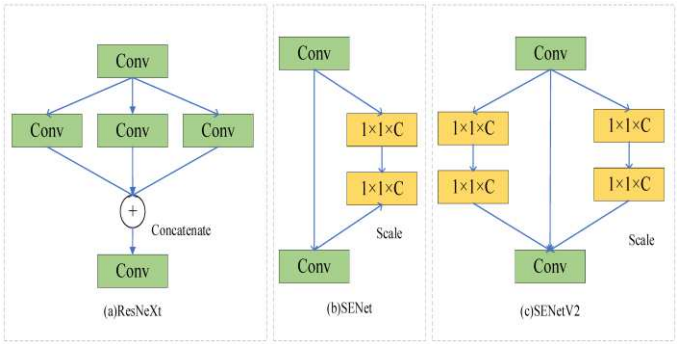


FIGURE 7. ResNeXt, SENet, SENetv2 modules.

Figure 8 illustrates the internal mechanism of the SaE module in SENetV2. First, the input features are compressed via a squeeze operation to generate compact feature descriptors, which are then fed into multi-branch fully connected layers. The resulting outputs go through an excitation stage, and the segmented features are finally concatenated to reconstruct the original feature structure. This design allows the network to capture richer and more diverse feature patterns from the input, effectively models inter-channel dependencies during feature transformation, and significantly improves the network's overall feature representation capability.

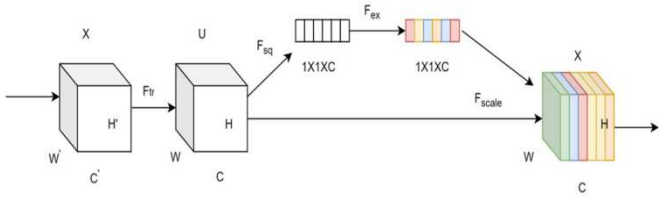


FIGURE 8. Structure of the SE module.

The Squeeze-and-Excitation module is a lightweight channel attention mechanism designed to adaptively recalibrate convolutional feature responses by explicitly modeling the interdependencies between different feature channels. As illustrated in Figure 8, the module comprises four core operations: feature transformation (F_{tr}), squeeze (F_{sq}), excitation (F_{ex}), and feature scaling (F_{scale}).

First, the input feature map $X \in R^{H'*W'*C'}$ undergoes a standard convolution operation to produce the intermediate feature map $U \in R^{H'*W'*C'}$. This transformation process, denoted as F_{tr}, is formulated as:

$$U_C = V_C * X = \sum_{s=1}^{C'} V_C^s * X^s \qquad (1)$$

where V represents the set of learnable convolution kernels, V_C denotes the parameters of the $c-th$ filter, V_C^S refers to the 2D spatial kernel corresponding to the $s-th$ input channel, X^S is the feature map of the $s-th$ input channel. Subsequently, the squeeze operation F_{sq} compresses the spatial dimensions of U to aggregate global contextual information for each individual channel. Specifically, global average pooling is applied across the $H \times W$ spatial dimensions, transforming the feature map U_C into a compact channel descriptor vector $Z \in R^{1X1XC}$. The mathematical formulation is as follows:

$$Z_C = F_{Sq}(U_C) = \frac{1}{H*W}\sum_{i=1}^{H}\sum_{j=1}^{W} u_C(i,j) \qquad (2)$$

Where $u_C(i,j)$ denotes the feature value at spatial position (i,j) in the $c-th$ channel of U.

Following the squeeze step, the excitation operation F_{ex} is performed to capture channel-wise dependencies and generate adaptive channel weights. The vector Z is fed into two consecutive fully connected layers: the first layer reduces the dimensionality of Z and applies a ReLU activation to introduce non-linearity, while the second layer restores the original dimensionality and employs a sigmoid activation function to output normalized attention weights. Finally, the learned weights are applied to the original feature map U via the scaling operation F_{scale}, yielding the recalibrated feature map, where each channel is adaptively emphasized or suppressed according to its task-specific importance.

Next, the module's core employs a stacked depthwise separable convolution (DWConv) structure, effectively separating spatial and channel convolutions. Additionally, the module draws from the SENetV2 concept to deeply couple the SE attention mechanism with multi-scale feature extraction. This fusion enables the model to have enhanced channel sensitivity and information filtering during feature aggregation, strengthening responses to key features while suppressing background noise and redundant information. As a result, the discriminative power and robustness of feature representation are improved.

Unlike the original SPPF module, which relies solely on pooling operations to build the feature pyramid, the proposed DS_SENetV2 module combines DWConv, SE, and SENetV2 mechanisms. This fusion achieves a triple optimization effect: selective feature enhancement, reduced computational cost, and improved scale awareness.

5 Experimental Results and Analysis

5.1 Experimental Environment

Table 1 shows the experimental environment and configuration used in this study.

Table 1. Experimental Environment and Configuration Details

Item	Experimental Environment
System	Ubuntu 18.04.6 LTS
CPU	Intel Xeon(R) Silver 4210 CPU @ 2.20GHz × 40
GPU	NVIDIA GeForce RTX 2080 Ti/PCIe/SSE2
Programming	Python 3.8
Software	Anaconda、Pycharm2021
Deep Learning Framework	Pytorch 2.3.0+cu121
GPU Acceleration Library	CUDA 12.1

5.2 Evaluation Metrics

To comprehensively evaluate the performance of the proposed model, five evaluation metrics are adopted in this study: frames per second (FPS), mean average precision at IoU threshold 0.5 (mAP@0.5), model weight size, number of parameters, and computational complexity. Among them, the calculation formulas for mean average precision (mAP@0.5) are shown in equations (3) to (6).

$$P = \frac{TP}{TP + FP} \quad (3)$$

$$R = \frac{TP}{TP + FN} \quad (4)$$

$$AP = \int_0^1 P(R)dR \quad (5)$$

$$mAP = \sum_{i=1}^{M} \frac{AP_i}{M} \quad (6)$$

Where TP denotes true positives, i.e., the number of samples correctly predicted as positive by the model; FP denotes false positives, i.e., the number of samples incorrectly predicted as positive; FN denotes false negatives, i.e., the number of samples that are positive but predicted as negative by the model; P(R) denotes precision at different recall rates; M denotes the number of classes; and AP_i denotes the average precision of the i-th class.

5.3 Ablation Study

Multiple groups of ablation experiments were conducted to evaluate the detection effectiveness of each module in the SDD-YOLO11n model. The ablation results are shown in Table 3.

According to the comparison of the ablation experiments, replacing the backbone of the YOLO11n model with ShuffleNetV2 significantly reduces the number of parameters and computational complexity. Specifically, the parameters, computation, and model weight size decreased by 64.4%, 51%, and 53%, respectively, while the model's mAP@0.5 increased by 1.4%. This indicates that lightweight backbone replacement maintains model accuracy and substantially improves detection performance. After integrating the DS_SENetV2 module, the mAP@0.5 increased by 1.2% without significant changes in parameters or computation, demonstrating that DS_SENetV2 enhances the model's scale awareness and feature selectivity. When the DW_ADown downsampling module was replaced, both parameters and computation decreased, while precision remained stable, and mAP@0.5 improved by 1.5%. This shows that the DW_ADown downsampling module

effectively retains spatial feature extraction capabilities and reduces information loss while significantly lowering computational burden.

By comparing experiments ① and ⑤, it can be seen that while parameters and computation decreased by 67.2% and 57.8%, respectively, the model's mAP@0.5 increased by 1.4%. Comparing experiments ② and ⑤ reveals that after replacing the backbone with ShuffleNetV2, parameters, and computation drastically dropped by 67.3% and 58.4%, respectively, with mAP@0.5 remaining unchanged. Comparing experiments ① and ⑥ shows that after integrating all three modules, the model's mAP@0.5 increased by 1.3%, while parameters, computation, and weight size decreased by 69.1%, 65.6%, and 63.5%, respectively. Additionally, as shown in Fig. 8, SDD_YOLO11n demonstrates better training stability than YOLO11n.

In summary, by replacing the backbone with ShuffleNetV2, integrating DS_SENetV2, and replacing the DW_ADown downsampling module, the computational cost and parameter count can be significantly reduced while improving traffic accident detection accuracy. Although the improved model experiences a slight decrease in frame rate, it still meets real-time detection requirements.

Table 3. Ablation Experiment Results

Group	ShuffleNetV2	DS_SENetV2	DW_Adown	mAP0.5	Size/MB	Params/M	FPS	GFLOS
①	×	×	×	0.811	5.24	2.59	75	6.4
②	×	√	×	0.823	5.28	2.6	87.2	6.5
③	×	×	√	0.826	4.3	2.1	84.3	5.2
④	√	×	×	0.827	2.06	0.92	71	3.1
⑤	√	√	×	0.825	2.0	0.85	58.7	2.8
⑥	√	√	√	0.824	1.91	0.80	70.9	2.2

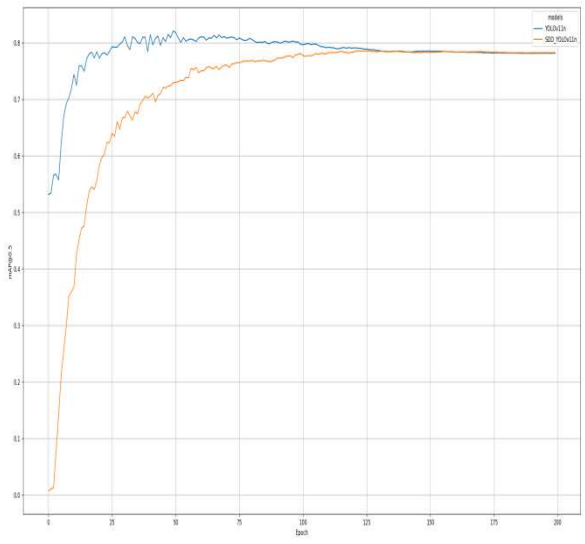


FIGURE 8. Comparison of mAP@0.5 before and after improvement.

5.4 Comparison Experiments

To verify the effectiveness of the SDD-YOLO11n algorithm, multiple comparison experiments were conducted under the same dataset, hardware, experimental environment, framework, and training strategies. The comparison results are shown in Table 4.

The lightweight improved model SDD-YOLO11n, designed based on the original YOLO11n network structure, demonstrates significant performance advantages. Experimental results indicate that compared with the original YOLO11n model, SDD-YOLO11n achieves a 1.3% improvement in mAP@0.5. Meanwhile, the model parameters decrease from 5.24M to 1.91M, FLOPs reduce from 6.4G to 2.2G, and the inference frame rate slightly decreases to 70.9 FPS. Overall, the model achieves a better balance between accuracy and efficiency.

Compared with the widely used classical detection model SSD, SDD-YOLO11n improves mAP@0.5 by 1.0%, reduces the parameter count by approximately 43.85M, and decreases FLOPs by 270.9G. The inference speed increases about 2.7 times (from 26.4 FPS to 70.9 FPS). These results fully validate that the proposed model maintains high accuracy and fast response under minimal computational resources, making it especially suitable for embedded systems with strict real-time and deployment efficiency requirements.

SDD-YOLO11n also shows significant advantages compared to multiple other models: Compared to YOLOv8n, it achieves a slight improvement in mAP@0.5

while significantly reducing parameter count and computational cost, greatly enhancing deployability on resource-constrained devices. Compared to larger models like YOLOv8s and YOLOv9c, SDD-YOLO11n attains higher accuracy with over 90% model compression. Even when compared with the YOLO11s model from the same series, SDD-YOLO11n achieves comparable or better detection accuracy with less than 10% of the parameter size, fully demonstrating its outstanding lightweight design and efficient inference performance, highlighting its great engineering application value.

Table 4 Comparison of Detection Performance Among Different Models

Model Name	mAP0.5	Size/MB	Params/M	FLOPs/G	FPS
SSD	0.814	45.76	23.612	273.1	26.4
YOLOv8n	0.821	5.89	3.01	8.2	120.4
YOLOv8s	0.8	21.49	11.13	28.6	93.3
YOLOv9c	0.809	49.22	25.3	102.3	63
YOLO11n	0.811	5.24	2.59	6.4	75
YOLO11s	0.811	18.31	9.41	21.3	106
SDD_YOLO11n	0.824	1.91	0.80	2.2	70.9

In summary, compared with large-scale models such as YOLOv9c, YOLOv7, and SSD, the SDD-YOLO11n significantly reduces the number of parameters and computational costs while achieving better mAP@0.5 performance, improving deployment efficiency. Compared with medium-sized models such as YOLOv8s and YOLO11s, although the frame rate is slightly lower, SDD-YOLO11n achieves a better balance between accuracy and resource consumption, offering higher precision and compression ratio. Compared with models of similar sizes, such as YOLOv8n and YOLO11n, SDD-YOLO11n further improves detection accuracy while maintaining a lightweight design. SDD-YOLO11n demonstrates excellent overall performance in accuracy, efficiency, and model complexity, making it especially suitable for real-time detection scenarios under resource constraints. As shown in Fig. 9, the proposed model's training curve exhibits more minor fluctuations than others, indicating a more stable training process, reduced overfitting, and stronger generalization capability.

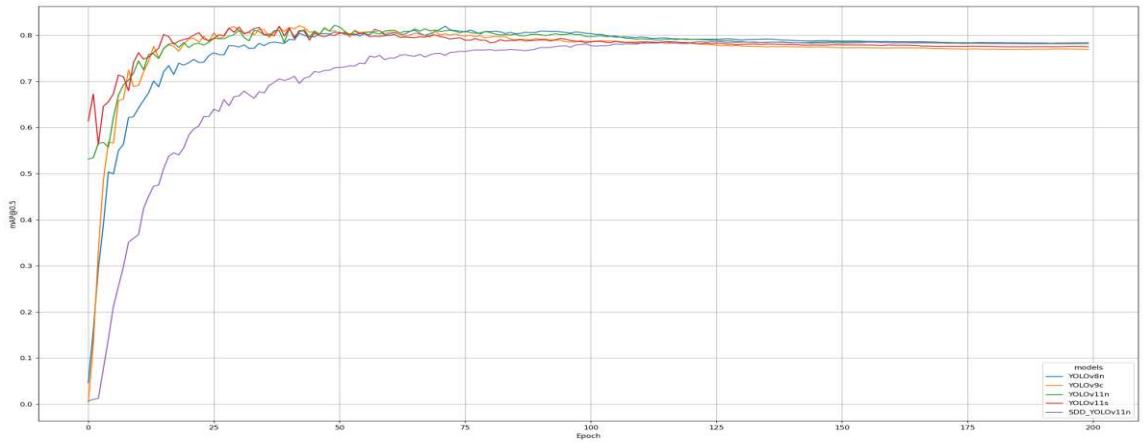


FIGURE 9. Comparison of mAP@0.5 curves among different models during training

5.5 Detection Performance Comparison and Analysis

To verify the proposed algorithm's practical detection performance, a random image was selected from the dataset for testing. The tested models include SDD_YOLO11n, YOLO11n, YOLO11s, YOLOv8n, YOLOv8s, and YOLOv9c. The detection results are shown in Fig. 10.

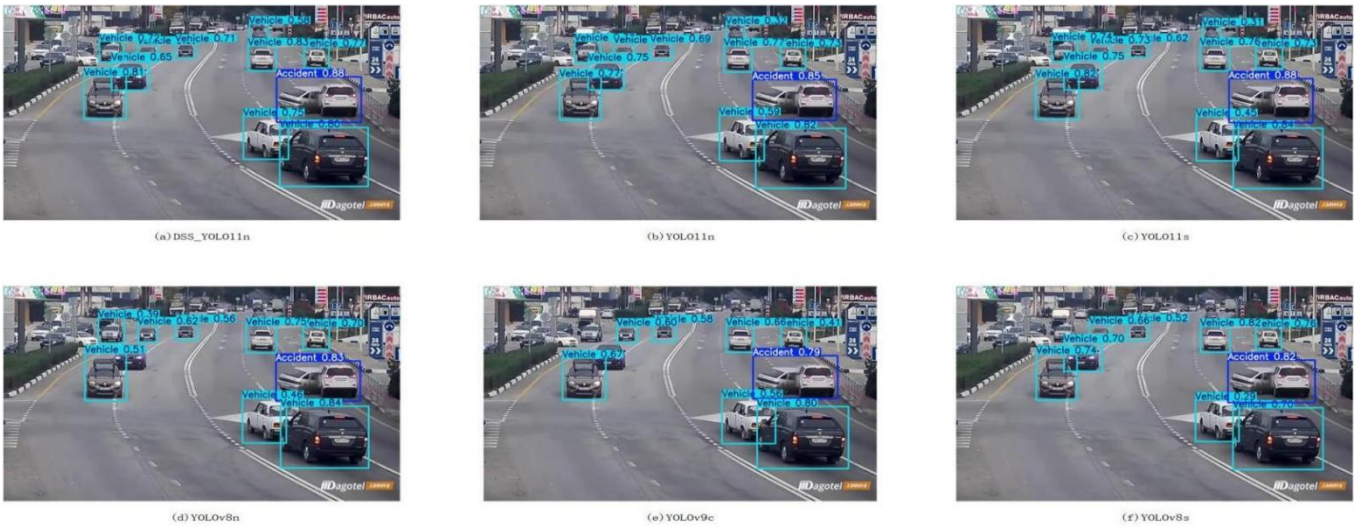


FIGURE 10. Comparison of detection results on the test image

As shown in the comparison results of Fig. 10 (a), (b), and (c), the proposed SDD_YOLO11n achieves higher confidence scores and does not exhibit missed detections or false positives for vehicles on the road. Comparing Fig. 10 (a) with (d), (e), and (f), it can be seen that SDD_YOLO11n has higher confidence than YOLOv8s, YOLOv8n, and YOLOv9c. Meanwhile, YOLOv8n misses two standard moving vehicles, YOLOv8s misses three, and YOLOv9c misses two. These results demonstrate that SDD_YOLO11n achieves the best detection performance while maintaining a lightweight design.

6 Conclusion

This paper proposes a lightweight Yolo11n algorithm for traffic accident emergency response scenarios requiring faster detection speeds, higher accuracy, and better adaptability to complex environments.

By introducing the ShuffleNetV2 module to replace the original backbone, the model significantly reduces the number of parameters while enabling cross-group information exchange, thereby improving detection accuracy.

The DW_Adown downsampling module replaces the traditional convolutional downsampling layer used in the original YOLO11n model. It effectively preserves spatial feature extraction and reduces information loss while significantly lowering computational cost and network burden.

The designed DS_SENetV2 module achieves a triple optimization effect of enhanced feature selection, reduced computational cost, and improved scale awareness.

Comparison experiments show that the improved SDD_YOLO11 model achieves better detection performance while reducing model parameters, computation, and weight size by 69.1%, 65.6%, and 63.5%, respectively, making it more suitable for deployment on embedded devices.

However, the proposed model still has some limitations. For example, the dataset does not include a detailed categorization of traffic accident types. Future work will focus on deploying the network on embedded devices for further study and refining the dataset by more detailed classification of accident types.

Founding

This work received no external funding.

Data availability statement

The original VADV dataset is publicly available at https://universe.roboflow.com. The filtered, augmented, and re-annotated dataset generated in this study is not publicly available due to institutional restrictions but is available from the corresponding author upon reasonable request.

References

[1] Karthikeyan, M., Manesh, V., Krishna, L. S., Vijay, B., Vishwabharan, R., & Prabhu, E. (2021). IoT based accident detection and response time optimization. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 358 – 363). IEEE.https://doi.org/10.1109/ICCMC51019.2021.9418272

[2] Liu, J. (2019). A rollover accident detection algorithm based on RCNN. Computer Era, (01), 55 – 58. https://doi.org/10.16644/j.cnki.cn33-1094/tp.2019.01.016

[3] Zhu, C., Guo, Z., & Ke, J. (2021). Advanced fuzzy-logic-based traffic incident detection algorithm. Advances in Fuzzy Systems, 2021, 1.

[4] Yang, J. (2024). Research on traffic accident detection technology based on convolutional neural networks [Master's thesis, Southwest University of Science and Technology]. CNKI. https://doi.org/10.27415/d.cnki.gxngc.2024.001041

[5] Liu Y, Su H, Zeng C, et al. A robust thermal infrared vehicle and pedestrian detection method in complex scenes[J]. Sensors, 2021, 21(4): 1240.

[6] Terven, J.; Córdova-Esparza, D.-M.; Romero-González, J.-A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Mach. Learn. Knowl. Extr. 2023, 5, 1680 – 1716.

[7] Wang, X. (2023). Research on vehicle collision traffic accident detection based on deep learning [Master's thesis, People's Public Security University of China]. CNKI. https://doi.org/10.27634/d.cnki.gzrgu.2023.000305

[8] Hou, X. (2023). Research and application of fast traffic accident detection based on feature fusion [Master's thesis, South China University of Technology]. CNKI. https://doi.org/10.27151/d.cnki.ghnlu.2023.005626

[9] Chen, J., Lu, S., & Zhong, L. (2024). An Autonomous Intelligent Liability Determination Method for Minor Accidents Based on Collision Detection and Large Language Models. Applied Sciences, 14(17), 7716. https://doi.org/10.3390/app14177716

[10] Luo, Z., Bi, Y., Ye, Q., Li, Y., & Wang, S. (2025). A Novel Machine Vision-Based Collision Risk Warning Method for Unsignalized Intersections on Arterial Roads. Electronics, 14(6), 1098. https://doi.org/10.3390/electronics14061098

[11] Hong, W., Qi, T., Liu, W., et al. (2025). Research on vehicle collision detection method based on machine vision. Electronic Design Engineering, 33(03), 8 – 12. https://doi.org/10.14022/j.issn1674-6236.2025.03.002

[12] Tang, Z., Tang, Y., Pan, D., et al. (2024). Traffic accident severity detection based on UAV. Modern Electronic Technology, 47(20), 123 – 128. https://doi.org/10.16652/j.issn.1004-373x.2024.20.019

[13] Wei, C., Tan, Z., Qing, Q., Zeng, R., & Wen, G. (2023). Fast Helmet and License Plate Detection Based on Lightweight YOLOv5. Sensors, 23(9), 4335. https://doi.org/10.3390/s23094335

[14] Zhang, W., & Jiang, F. (2025). AHN-YOLO: A Lightweight Tomato Detection Method for Dense Small-Sized Features Based on YOLO Architecture. Horticulturae, 11(6), 639. https://doi.org/10.3390/horticulturae11060639

[15] Wang, Z., Lei, X., & Zhou, S. (2024). Traffic sign detection based on lightweight RT-DETR. In 2024 IEEE 2nd International Conference on Sensors, Electronics and Computer Engineering (ICSECE) (pp. 1593–1597). IEEE.https://doi.org/10.1109/ICSECE61636.2024.10729388

[16] Zhao, W., Ren, C., & Tan, A. (2024). Study on Nighttime Pedestrian Trajectory-Tracking from the Perspective of Driving Blind Spots. Electronics, 13(17), 3460. https://doi.org/10.3390/electronics13173460

[17] Li, R. (2023). YOLOv5-based traffic sign detection algorithm. In 2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI) (pp. 1162 – 1165). IEEE.https://doi.org/10.1109/ICETCI57876.2023.10176904

[18] Zou, H., Zhan, H., & Zhang, L. (2022). Neural Network Based on Multi-Scale Saliency Fusion for Traffic Signs Detection. Sustainability, 14(24), 16491. https://doi.org/10.3390/su142416491

[19] Wei, C., Tan, Z., Qing, Q., Zeng, R., & Wen, G. (2023). Fast Helmet and License Plate Detection Based on Lightweight YOLOv5. Sensors, 23(9), 4335. https://doi.org/10.3390/s23094335

[20] Fang, S., Chen, C., Li, Z., Zhou, M., & Wei, R. (2024). YOLO-ADual: A Lightweight Traffic Sign Detection Model for a Mobile Driving System. World Electric Vehicle Journal, 15(7), 323. https://doi.org/10.3390/wevj15070323

[21] Zheng, X., Qian, S., Wei, S., Zhou, S., & Hou, Y. (2023). The Combination of Transformer and You Only Look Once for Automatic Concrete Pavement Crack Detection. Applied Sciences, 13(16), 9211. https://doi.org/10.3390/app13169211

[22] Bu, L., Li, W., Zhang, H., Wang, H., Tian, Q., & Zhou, Y. (2025). FIRE-YOLOv8s: A Lightweight and Efficient Algorithm for Tunnel Fire Detection. Fire, 8(4), 125. https://doi.org/10.3390/fire8040125

[23] Jia, P., Sheng, H., & Jia, S. (2025). LPCF-YOLO: A YOLO-Based Lightweight Algorithm for Pedestrian Anomaly Detection with Parallel Cross-Fusion. Sensors, 25(9), 2752. https://doi.org/10.3390/s25092752

[24] Li, B., Meng, Q., Li, X., Wang, Z., Liu, X., & Kong, S. (2024). Enhancing YOLOv8's Performance in Complex Traffic Scenarios: Optimization Design for Handling Long-Distance Dependencies and Complex Feature Relationships. Electronics, 13(22), 4411. https://doi.org/10.3390/electronics13224411

[25] Song, J., Hu, T., Gong, Z., Zhang, Y., & Cui, M. (2024). TLDM: An Enhanced Traffic Light Detection Model Based on YOLOv5. Electronics, 13(15), 3080. https://doi.org/10.3390/electronics13153080

[26] Jin, M., & Zhang, J. (2024). Research on Microscale Vehicle Logo Detection Based on Real-Time DEtection TRansformer (RT-DETR). Sensors, 24(21), 6987. https://doi.org/10.3390/s24216987