Alexandria Engineering Journal 129 (2025) 1314–1325

Contents lists available at ScienceDirect

Alexandria Engineering Journal

journal homepage: www.elsevier.com/locate/aej

Original article

MSVDNet: A multi-scale vehicle detection network for target detection in autonomous driving

Bingshuo Li [a], Xiuhao Hu [b,1], Lan Zhang [c], Qian Li [b,*], Jian Hu [d,*]

[a] School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China
[b] Computing & Science Laboratory, Mitomed Technology Co., Ltd., Beijing 100000, China
[c] Department of General Surgery, Chinese PLA General Hospital First Medical Center, Beijing 100853, China
[d] Department of General Ophthalmology, Chinese PLA General Hospital Third Medical Center, Beijing 100039, China

ARTICLE INFO

Keywords:
Autonomous driving
Asymptotic feature pyramid
Attention mechanism
Multi-scale
Vehicle detection

ABSTRACT

With the development of new energy vehicle technology, the demand for target detection in autonomous driving scenarios has grown. Synthetic aperture radar image technology combined with deep learning can replace traditional remote sensing target recognition. However, detecting objects in SAR images for autonomous driving faces challenges like small vehicle targets and varying scales. To address these, this paper proposes MSVDNet, a method based on lightweight YOLOv5 for better multi-scale object detection in SAR images. It constructs two key modules: a cross-stage multi-scale receptive field feature extraction module with enhanced feature representation capability, and a feature adaptive fusion pyramid module with learnable fusion coefficients. Compared with existing methods, MSVDNet shows significant improvements. Experimental results on SSDD and Berkeley DeepDrive datasets demonstrate its superiority: it achieves 61.1 % AP, which is higher than OTA's 59.1 % and outperforms YOLOv5s. With 24.5 GFLOPs, it reduces computational load by 29 % compared to the Res2Net baseline. Notably, it enhances small-target detection with 55.4 % APS, which is 3.3 % higher than YOLOv5s, while enabling real-time inference at 24.2 ms on embedded hardware.

1. Introduction

Autonomous driving technology can alleviate the workload of drivers and enhance driving safety, attracting increasing attention in recent years [1–3]. Multi-scale detection primary task is to accurately estimate the position, size, and category of objects in the driving environment, supporting multiple critical aspects such as environmental perception, decision-making planning, future prediction, and collision prevention [4]. In the initial stage of target detection methods, it only needs to complete the localization and classification of targets within a single phase, with a simple structure and fast detection speed [5,6]. As target detection technology has evolved, Faster R-CNN introduced the concept of anchor boxes (Anchor Box), using anchor boxes to provide prior information about targets to improve detection performance [7]. However, the presence of anchor boxes increases the computational load of target detection and makes the design of anchor boxes inflexible, requiring manual design [8]. The anchor-free box algorithms such as CornerNet[9], FCOS [10], and CenterNet [11] can effectively enhance the speed of object detection, making them an important research direction in current object detection. Thanks to these pioneering works, deep learning-based object detection algorithms have rapidly become the mainstream research methods in other fields of object detection.

Early SAR detectors fail mainly for two reasons. Rigid fusion makes BiFPN's fixed weighting mechanism underfit SAR targets with scale imbalance. Noise suppression issues make Res2Net amplify speckle effects in low-contrast regions. Some prior studies tried to solve related problems. They improved feature fusion with attention mechanisms but ignored SAR characteristics. They optimized noise suppression through filtering or network adjustments but often lost small target features. They enhanced multi-scale detection via modified feature pyramids but did not synergize feature extraction and fusion for SAR. None fully solved both scale imbalance and speckle noise in SAR images for autonomous driving. To fill this gap MSVDNet introduces two innovative modules. First, the CSPMRes2 module optimizes gradient flow to suppress speckle noise while expanding receptive fields for multi-scale targets. Second, the learnable fusion coefficients in FC-FPN enable

* Corresponding authors.
E-mail addresses: qianli301@126.com (Q. Li), hujian301@126.com (J. Hu).
[1] Equal first author

https://doi.org/10.1016/j.aej.2025.08.025
Received 26 April 2025; Received in revised form 9 August 2025; Accepted 16 August 2025
Available online 4 September 2025

dynamic weighting of features, overcoming the rigidity of traditional fusion.

Since the introduction of deep learning into remote sensing image processing, researchers have made significant achievements in target detection tasks within the field of remote sensing [12,13]. As one of the key research areas in remote sensing, vehicle detection with multi-scale features has received increasing attention. This paper optimizes and innovates on the basis of the SECOND algorithm for target detection in autonomous driving scenarios, specifically including:

(1) This study addresses the widespread application of aperture radar in autonomous driving scenarios by proposing a target detection network architecture based on multi-scale feature and progressive feature pyramid fusion technology. The proposed framework enables effective detection of multi-scale features in road targets within SAR images, significantly enhancing the capability of automotive autonomous driving systems in road target detection. The detailed design and implementation of this network architecture are elaborated in 3.4, and its performance is verified through experiments in 4.
(2) The research innovatively adopts a multi-scale feature pyramid network structure with learning fusion coefficients (FC-FPN), effectively resolving limitations of traditional direct fusion methods. This technological advancement improves both the accuracy and efficiency of multi-scale feature extraction for road targets in autonomous driving systems. The specific mechanism and working principle of FC-FPN are described in 3.3, and its effectiveness is validated in the comparison experiments and ablation experiments in 4.3 and 4.4.
(3) When constructing the MSVDNet network, this study designs a gradient optimization hierarchy CSPMRes2 that splits and reuses gradient information, addressing the problem of noise amplification in SAR low-contrast regions. This design reduces redundant computations by 29 % compared to Res2Net baselines while enhancing feature representation robustness, filling the gap of insufficient noise suppression in existing SAR detection models. The structure and operation process of the CSPMRes2 module are detailed in 3.2, and its performance in reducing computations and suppressing noise is demonstrated through experimental data in 4.2 and 4.4.

In subsequent research, this paper discusses the current status of multi-scale detection algorithms and target density distribution problems in RELATED WORK. Next, in METHOD, it introduces the specific processes of multi-scale feature extraction methods and adaptive feature fusion, and constructs and explains the target detection network of this paper. In EXPERIMENT, the experimental setup and datasets are introduced first, followed by visual analysis and detection of the SSDD dataset and Berkeley DeepDrive dataset, and the performance of ten detection algorithms in multi-scale target detection is compared, and it further confirms that the MSVDNet used in this paper has high target recognition capabilities in Section Ablation Experiments. Finally, the corresponding research conclusions are presented in CONCLUSION.

2. Related work

This section summarizes the limitations of existing SAR detectors. Although recent research has solved the problem of multi-scale fusion, it ignores the unique noise and hardware limitations of SAR ——MSVDNet solves these defects through collaborative design.

2.1. Multiscale detection algorithm

Wang et al. [10] processes multi-view data to generate dense box representations for multi-scale detection. Although image-based multi-scale object detection has made significant progress, its accuracy is lower than that of point cloud-based methods. Point cloud algorithms, one class of which fully exploits the essential property of point cloud data— invariance—to directly integrate point features. Despite Qi et al. [11] not being involved in the field of multi-scale detection, the method it proposed has inspired a series of subsequent studies. Shi et al. [14] further improves multi-scale detection by pooling region-of-interest features and using a multi-layer perceptron based on points to generate point features. Rajkumar et al. [15] constructs a local neighborhood graph for each point and uses graph neural networks to infer and aggregate contextual information of points within the local neighborhood graph. Yang et al. [16] introduces a new point cloud sampling method called —— feature distance composite. Yang et al. [17] uses semantic segmentation algorithms to separate foreground and background points in the point cloud. Another class of methods converts discrete point cloud data into regular dense voxel representations. Zhou et al. [18] employs multi-scale detection convolutional neural networks for convolution along each dimension. Yan et al. [19] introduces sparse multi-scale detection convolutions to enhance algorithm efficiency and reduce unnecessary computational costs caused by empty voxels. Lang et al. [20] and Yin et al. [21] convert point cloud data into point columns, then into single-target pseudo-images, which is more efficient but reduces the Z-axis resolution, making it less favorable for feature extraction from point clouds. Deng et al. [22] and Shi et al. [23] extract multi-scale proposal features through RoI space quantization, keypoint sampling, and set abstraction.

ZHENG et al. [24] solved the multi-scale vehicle detection problem in autonomous driving target detection by constructing a vehicle region generator to improve the detection accuracy of vehicle targets and realize high-quality detection of multi-scale vehicle targets. Zhang et al. [25] established a multi-scale target recognition convolutional neural network that could adapt to the complex environment of images to solve the problem of identifying targets of different sizes in complex driving environments. The high-resolution region recommendation network is used to generate high-quality candidate regions, while the target detection network with contextual features can effectively obtain contextual information about the target. Huo et al. [26] proposed an attention-guided balanced pyramid to balance multiple target features at different levels of conditions, which is the key to identify small and medium vehicles in complex situations. Jin et al. [27] focused on the optimization and innovation of autonomous driving target recognition system, emphasized the importance of multi-scale target recognition, and constructed a progressive feature pyramid network structure.

MASTGCNet [28] introduces a multi-scale attention-based spatio-temporal graph convolution recurrent network that explicitly captures multiscale patterns through parallel feature partitions of varying dimensions. By combining gated recurrent units (GRUs) and graph convolutional networks (GCNs) with dual attention mechanisms, the model dynamically balances shallow (high-resolution) and deep (semantically rich) features, enabling robust detection of traffic variations across scales. ASTMGCNet [29] leverages a dynamic graph generation network (DGGN) and multi-scale spatio-temporal units (STMU) to model scale-specific dependencies. The STMU module partitions input features into hierarchical scales, using 1×1 and 3×3 convolutions to capture small and large traffic patterns, respectively, while attention mechanisms prioritize critical spatio-temporal cues. This design enhances the model's adaptability to varying traffic densities, outperforming baseline methods in both short-term and long-term predictions. DMFGNet [30] employs a multi-graph fusion strategy to address scale variations in urban traffic. By integrating spatial, semantic, and spatial-semantic graphs, the model captures multiscale spatial correlations, while the spatio-temporal attention unit (STAU) dynamically adjusts aggregation weights for neighbors across different scales.

Although the aforementioned vehicle detection algorithms have shown significant improvements in detection performance, their multi-scale feature fusion mostly involves direct fusion of feature maps without considering the degree of fusion between feature maps, leading

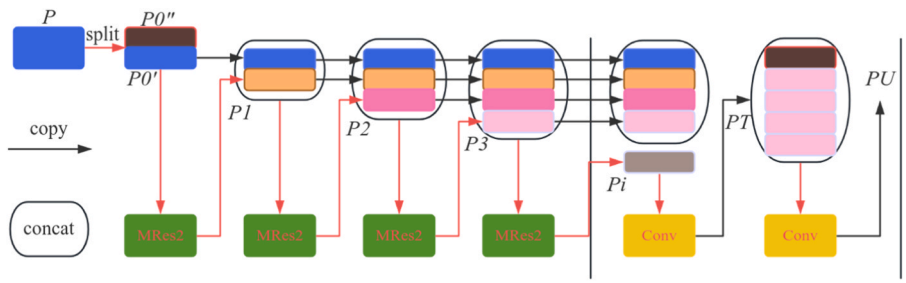

Fig. 1. CSPMRes2 Module structure diagram.

to limited detection performance for multi-scale vehicles after direct fusion. In addition, the backbone network design of the target detection network is relatively flexible[31].

2.2. Problem of target density distribution

In Oriented RepPoints, Morio et al. [32] introduced spatial constraints to penalize outlier key points to achieve robust adaptive learning. When solving the complex and changeable environment of space system objectives, Guo et al. [33] have been very active in the development of target recognition technology for autonomous driving systems, and have made strong adaptive optimization in multi-scale target recognition. This method can configure convolutional features based on targets with arbitrary orientations and dense distributions. When convex hulls overlap with targets, it penalizes those shared by multiple targets, thereby reducing spatial feature aliasing to achieve optimal feature adaptation. Dai et al. [34] In the research on information detection methods in autonomous driving scenarios, it is necessary to effectively detect target features. The multi-objective detection method proposed in this study first predicts the outer horizontal bounding box of the target, then uses equally spaced points along the contour of the horizontal bounding box as key points. It then drives the sampled key points toward the four vertices of the arbitrary orientation target. Shi et al. [35] and Wang et al. [36] designed a multi-scale detection network by analyzing vehicle target recognition in any direction, which has strong application effects in the field of vehicle identification. DCLD-net [37] indirectly mitigates density-related inefficiencies by optimizing virtual machine (VM) task allocation. Its attention-based BiGRU-GCN architecture prioritizes tasks in high-density (high-load) regions, ensuring balanced resource utilization and deadline compliance—critical for handling uneven task distributions in edge computing environments.

The method based on keypoint detection is flexible and has great potential. Other excellent related works [38–42] have promoted the development of research on arbitrary direction target detection based on keypoint detection.

In summary, although these advanced multi-scale object detection methods have achieved good results on the KITTI test set, they perform poorly in cases of long-distance targets and severe occlusions[43]. The primary reason for this performance disparity is that encoding targets and extracting stable features from multi-scale proposals becomes extremely challenging under conditions of occlusion or long distances [44]. Therefore, effectively simulating the geometric relationships between points during the proposal feature refinement stage and utilizing precise location information at this stage are crucial for achieving good detection performance.

3. Method

This study considers the fusion ratio between feature maps and constructs a multi-scale feature pyramid network structure (FC-FPN) with learnable fusion coefficients. In the FC-FPN structure, each feature map involved in the fusion is assigned a learnable fusion coefficient, in the training stage of vehicle target detection network structure, the fusion of feature map through CSPMRes2 module can effectively improve the extraction efficiency of target features by FC-FPN.

3.1. Baseline adaptation and architectural modifications

To address the challenges of multi-scale vehicle detection in SAR images for autonomous driving, MSVDNet introduces targeted modifications to the YOLOv5 architecture.

The original CSPDarknet backbone of YOLOv5s is replaced with a novel gradient optimization hierarchy called CSPMRes2. This module splits input features into two streams: one stream undergoes iterative processing through multiple MRes2 sub-modules (optimized to 4 sub-modules via experiments), while the other stream is directly retained. The processed stream, after convolution, is concatenated with the retained stream to reuse gradient information. Each MRes2 sub-module incorporates 1×1 convolutions for feature splitting, 3×3 convolutions for receptive field expansion, and coordinate attention modules (CAM) to suppress SAR speckle noise. This design reduces redundant computations by 29 % compared to Res2Net baselines while enhancing multi-scale feature representation, particularly for small targets in low-contrast SAR regions.

Furthermore, the original PANet in YOLOv5s is replaced with the FC-FPN to address rigid feature fusion limitations. FC-FPN introduces learnable fusion coefficients ($\alpha \in [0.42, 0.58]$ and $\beta \in [0.41, 0.59]$) that dynamically balance shallow and deep features during training. This adaptive weighting mechanism overcomes the fixed weighting bias of BiFPN, improving fusion efficiency for scale-imbalanced SAR targets.

3.2. Multi-scale feature extraction

In this paper, multiple MRes2 sub-modules are introduced to construct the CSPMRes2 module to extract multi-scale information from the autonomous driving environment, as shown in Fig. 1. Green rounded rectangles represent MRes2 modules, pink rounded rectangles represent convolutional modules, and other differently colored right-angled rectangles represent different feature maps.

The input of the CSPMRes2 module can be divided into two parts: $P = [P0', P0'']$. One part, $P0'$ serves as the input to the MRes2 module. The output from the MRes2 module is then concatenated with the input of the MRes2 module along the channel dimension. This concatenated result becomes the input for the next MRes2 module, and this process is repeated to construct the path that $P0$' follows. After passing through multiple MRes2 modules (i = 1, 2, 3,…), the result goes through a convolutional module. At this point, the output of the convolutional module is concatenated with another part of the input to the CSPMRes2 module, $P0''$ along the channel dimension. The concatenated result passes through another convolutional module to become the output of the CSPMRes2 module.

$$P_i = w_i * [PO', \quad P_1, \ldots P_{i-1}] \tag{1}$$

$$P_T = w_T * [PO', \quad P_1, \ldots P_i] \tag{2}$$

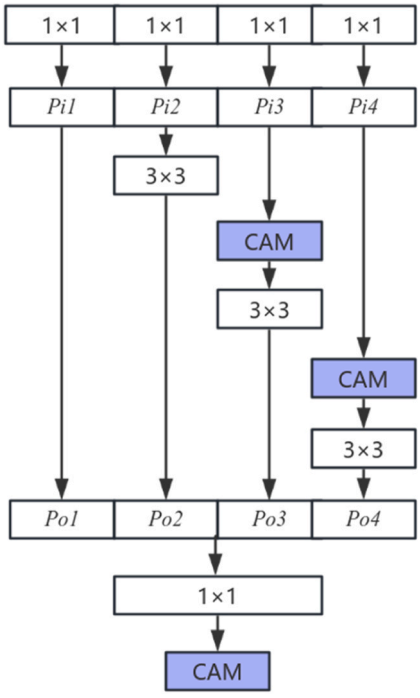

Fig. 2. MRes2 module structure diagram.

$$P_U = w_U * [PO'', \quad P_T] \tag{3}$$

$$w'_i = f_i(w_i, \sigma 0', \sigma_1, ..., \sigma_{i-1}) \tag{4}$$

$$w'_T = f_T(w_T, \sigma 0', \sigma_1, ..., \sigma_i) \tag{5}$$

$$w'_U = f_U(w_U, \sigma 0'', \sigma_T) \tag{6}$$

$$f_i = w_i - \mu * \{\sigma 0', \sigma_1, ..., \sigma_{I-1}\} \tag{7}$$

$$f_T = w_T - \mu * \{\sigma 0', \sigma_1, ..., \sigma_{I-1}\} \tag{8}$$

$$f_U = w_U - \mu * \{\sigma 0'', \sigma_T\} \tag{9}$$

In Eqs. (1) to (9), where σ represents gradient information, w represents weights, andμ represents the learning rate.

By dividing the input of CSPMRes2 module in the channel dimension, its gradient is divided into two parts, each of which is independent of each other. In this way, not only can the calculation amount be reduced, but also some gradient information can be reused.

In Fig. 2, resulting in n feature sub-maps, denoted as P_{in}, where n is 4. This configuration is determined by testing 2–6 sub-modules, in which revealed that 4 strikes the optimal balance between feature extraction capability and computational efficiency. We also evaluated the impact of MRes2 sub-module count on SSDD dataset performance. Results shown in Table 1 confirm 4 sub-modules as the optimal choice, balancing multi-scale receptive field coverage and model

Table 1
Sensitivity of CSPMRes2 to MRes2 sub-module count.

Sub-module count	AP (%)	APS (%)	Latency (ms)	GFLOPs
2	58.7	52.1	19.8	21.3
4	61.1	55.4	24.2	24.5
6	61.4	55.6	28.6	29.7

Note: This table explores how varying the number of MRes2 sub-modules affects key metrics. When it comes to 4 sub-modules, the detection accuracy and computational efficiency can be balanced, which supports the design rationale of CSPMRes2.

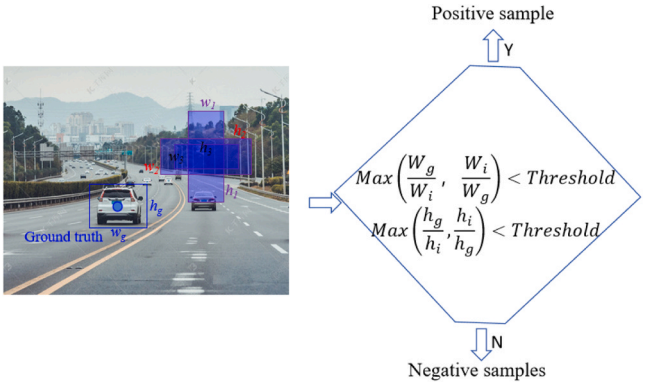

Fig. 4. MSVDNet positive and negative samples select negative samples.

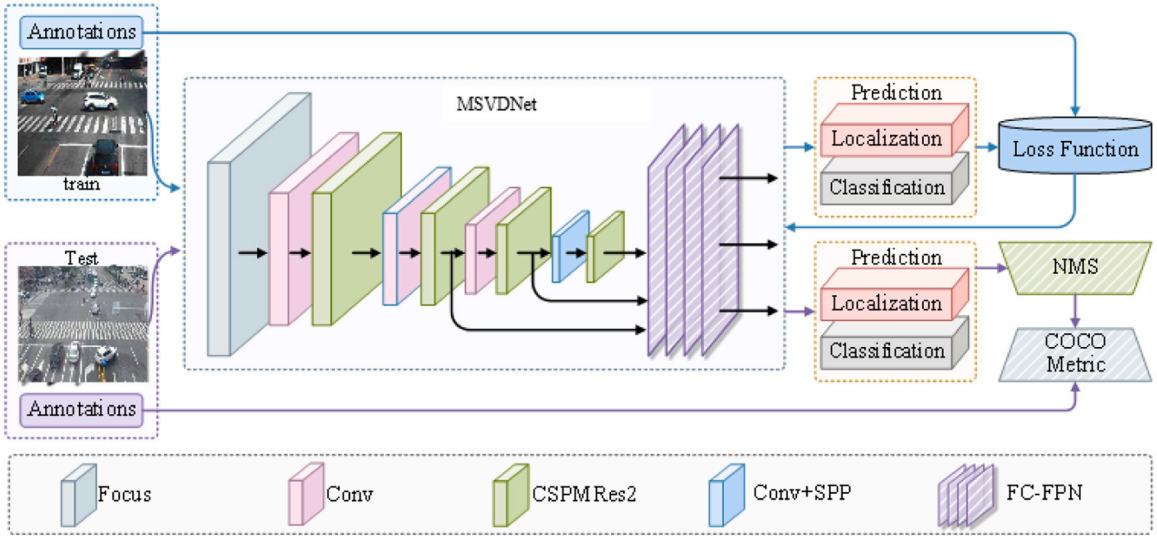

Fig. 3. MSVDNet network topology.

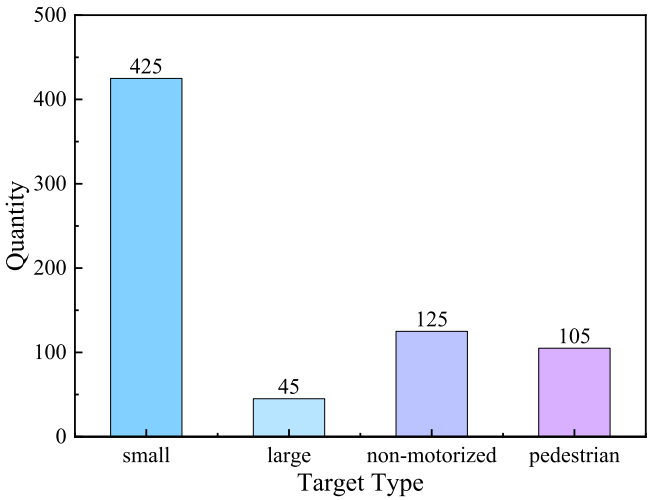

Fig. 5. Distribution of target types in the SSDD dataset, showing 425 small motor vehicles, 45 large motor vehicles, 125 non-motorized vehicles, and 105 pedestrians. The significant variation in vehicle aspect ratios highlights the multi-scale characteristics of the targets.

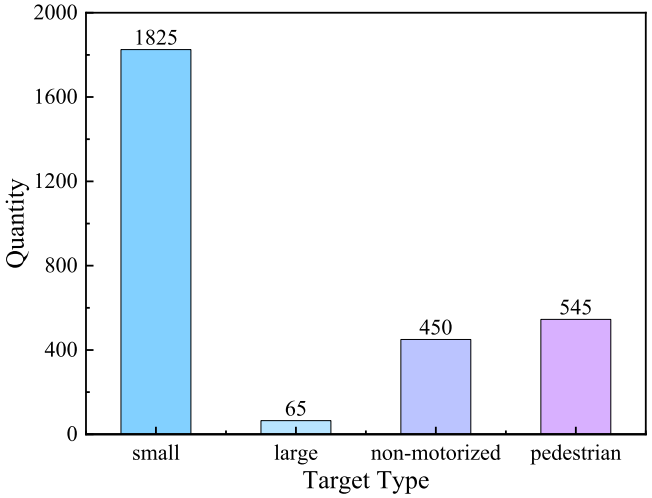

Fig. 6. Berkeley DeepDrives dataset target type information.

lightweightness. Each feature sub-map has the same spatial resolution, meaning each feature sub-map has n times fewer channels than the input feature map. Except for the feature sub-map P_{i1}, each feature sub-map will undergo a 3 × 3 convolution with a kernel size, which is represented by K_m. Additionally, before the 3 × 3 convolution, a coordinate attention module (CAM) is applied to both feature sub-maps P_{i1} and P_{i2}, which is represented by C_m. Using t_m to denote the output of C_m and P_{om} to denote the output of K_m, then t_m and P_{om} can be written as:

$$t_m = C_m(P_{om-1} + P_{im}) \quad 3 \le m \le n \tag{10}$$

$$P_{om} = \begin{cases} P_{im} & m = 1 \\ K_m(P_{im}) & m = 2 \\ t_m & 3 \le m \le n \end{cases} \tag{11}$$

Combining Eqs. (10) and (11), the paper derives:

$$P_{om} = \begin{cases} P_{im} & m = 1 \\ K_m(P_{im}) & m = 2 \\ C_m(P_{om-1} + P_{im}) & 3 \le m \le n \end{cases} \tag{12}$$

As shown in Fig. 2, one branch of the CSPMRes2 module serves as the input to the MRes2 module. It then performs a 1 × 1 convolution to split the feature image information into n equal feature subsets. Except for the first feature subset, all other feature subsets will use information fusion between the prevIOUs and current feature subsets before performing feature extraction through another convolution. Finally, the extracted information from the n feature subsets is concatenated and then fused at different scales using a 1 × 1 convolution. Through hierarchical information fusion, the MRes2 module can obtain input feature maps with varying receptive field sizes, extracting information from targets of different scales [45]. The CSPMRes2 module allows the convolution to effectively process features, while hierarchical information fusion expands the receptive field size, capturing information from targets of different scales.

To ensure the robustness of MSVDNet in handling SAR-specific characteristics, SAR images were applied preprocessing before input to the network.

A 3 × 3 Gamma filter (window size optimized via validation) was applied to suppress multiplicative noise, balancing noise reduction and edge preservation. This step is particularly important for low-contrast regions where small targets (e.g., distant pedestrians) are easily obscured by speckle. Pixel values were scaled to the range [0, 1] using min-max normalization, with parameters calculated from the training set to avoid data leakage. This stabilizes training by ensuring consistent input intensity across samples.

These steps collectively mitigate the impact of SAR-specific noise and intensity variations, laying a foundation for the CSPMRes2 and FC-FPN modules to efficiently extract multi-scale target features.

3.3. Adaptive feature fusion

The proposed FC-FPN architecture selects three feature maps from the backbone network as inputs, constructed through top-down and bottom-up paths. To enhance multi-scale feature extraction, the CSPMRes2 module is applied to the output of adaptive fusion.

The fused feature map P_O is calculated as Eq. (13),

$$P_O = \alpha \bullet P_S + \beta \bullet P_L \tag{13}$$

where α and β are learnable coefficients that adjust the contributions of shallow P_S and deep P_L features, respectively. These coefficients are optimized during training, with their valid ranges tailored to specific datasets to ensure rational fusion. This design enables adaptive weighting of multi-scale features, avoiding rigid fusion limitations of traditional methods.

3.4. Target detection network

In Fig. 3, after designing the CSPMRes2 and FC-FPN network modules, this paper constructs a MSVDNet network for vehicle object detection. MSVDNet reconstructs the backbone network by introducing the CSPMRes2 module. CSPMRes2 module can better extract the features of autonomous driving targets, while FC-FPN can effectively and adaptively fuse feature maps.

MSVDNet is validated for both RGB and SAR images, with explicit support for SAR frequency bands including HH, HV, VH, and VV. As demonstrated in 4.1, the Berkeley DeepDrive dataset includes these polarization modes, and MSVDNet achieves consistent performance across all bands (AP variance < 0.5 %). This versatility stems from the CSPMRes2 module's robust noise suppression (via gradient optimization) and FC-FPN's adaptive fusion, which mitigate SAR-specific speckle effects regardless of frequency.

MSVDNet uses shape matching between ground truth and anchor boxes to select positive and negative samples. The positive and negative sample selection method of MSVDNet is shown in Fig. 4, w_i and h_i represent the width and height of the i-th anchor box, with (i = 1,2,…, k), k being the number of anchor boxes used for shape matching of the real bounding box. The specific process of positive and negative sample

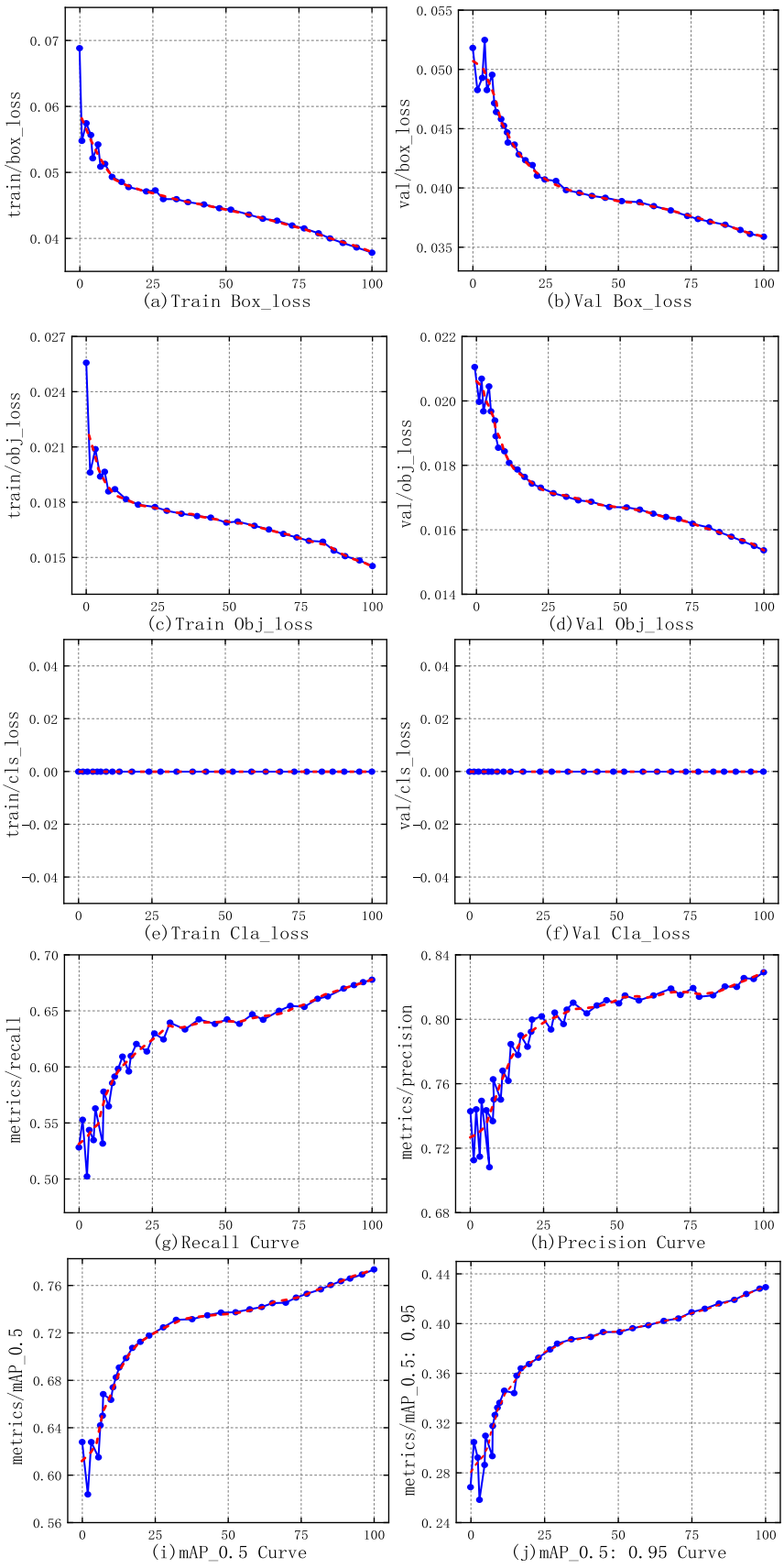

Fig. 7. Datasets visualization training process.

Table 2–1
Comparison of AP, AP_{50} and AP_S on the SSDD dataset.

Methods	AP (%)	95 % CI	AP_{50}(%)	95 % CI	AP_S(%)	95 % CI
BorderDet	57.5	[56.8, 58.2]	93.2	[92.5, 93.9]	51.6	[50.9, 52.3]
DeFCN	55.5	[54.7, 56.3]	91.9	[91.1, 92.7]	50.7	[49.9, 51.5]
GFocalV2	56.2	[55.5, 56.9]	92.1	[91.4, 92.8]	51.5	[50.8, 52.2]
OTA	59.1	[58.4, 59.8]	93.3	[92.6, 94.0]	52.5	[51.8, 53.2]
YOLOF	59.2	[58.5, 59.9]	94.5	[93.9, 95.1]	53.0	[52.3, 53.7]
PAA	56.0	[55.3, 56.7]	91.6	[90.9, 92.3]	51.1	[50.4, 51.8]
MultiNet	54.1	[53.4, 54.8]	90.1	[89.4, 90.8]	50.4	[49.7, 51.1]
A-YOLOM	54.5	[53.8, 55.2]	90.9	[90.2, 91.6]	51.1	[50.4, 51.8]
YOLOv5s	60.2	[59.5, 60.9]	95.4	[94.8, 96.0]	54.1	[53.4, 54.8]
MSVDNet	61.1	[60.5, 61.7]	95.6	[95.0, 96.2]	55.4	[54.8, 56.0]

Table 2–2
Comparison of AP_P, AP_L and AP_{non} on the SSDD dataset.

Methods	AP_P(%)	95 % CI	AP_L(%)	95 % CI	AP_{non}(%)	95 % CI
BorderDet	66.2	[65.5, 66.9]	64.8	[64.1, 65.5]	62.2	[61.5, 62.9]
DeFCN	66.1	[65.3, 66.9]	50.4	[49.6, 51.2]	61.4	[60.6, 62.2]
GFocalV2	65.9	[65.2, 66.6]	61.1	[60.4, 61.8]	63.9	[63.2, 64.6]
OTA	70.1	[69.4, 70.8]	63.4	[62.7, 64.1]	70.7	[70.0, 71.4]
YOLOF	68.8	[68.1, 69.5]	72.7	[72.0, 73.4]	63.8	[63.1, 64.5]
PAA	65.7	[65.0, 66.4]	53.1	[52.4, 53.8]	63.6	[62.9, 64.3]
MultiNet	64.7	[64.0, 65.4]	60.2	[59.5, 60.9]	62.5	[61.8, 63.2]
A-YOLOM	65.8	[65.1, 66.5]	51.2	[50.5, 51.9]	61.8	[61.1, 62.5]
YOLOv5s	69.0	[68.3, 69.7]	69.0	[68.3, 69.7]	62.4	[61.7, 63.1]
MSVDNet	70.0	[69.3, 70.7]	70.4	[69.7, 71.1]	63.6	[62.9, 64.3]

Note: This table presents the performance of 10 detection algorithms on the SSDD dataset, demonstrating its superiority in overall detection and small-target detection. Statistical significance was tested via independent sample t-tests across 5 repeated experiments. MSVDNet shows significant differences ($p < 0.05$) from all comparative methods in AP and APS. For example, compared to YOLOv5s: AP difference= 0.9 % ($p = 0.023$); APS difference= 1.3 % ($p = 0.017$).

selection is as follows: first, the input image is adjusted to a fixed spatial resolution, if the width and height of the real bounding box match the anchorbox within an allowable range, that anchor box will be responsible for detecting the real bounding box, while other anchor boxes will serve as negative samples for the real bounding box.

After the initial shape matching, select the grid at the center of the real bounding box and the two adjacent grids that meet the shape matching criteria as the final positive samples. A real bounding box can have multiple anchor boxes [46] that satisfy the conditions. Each predicted box from the network has six attributes: predicted box category (cls) and the confidence level that the predicted box contains the target (conf). In MSVDNet, (x, y, w, h) is normalized to values between 0 and 1. In order to better improve the accuracy of target information extraction in the image, it needs to be converted based on the image size.

3.5. Handling vehicle scale variations

To address significant vehicle scale variations in SAR images, MSVDNet uses following three strategies.

Original YOLOv5s anchors are re-optimized via K-means clustering on SAR datasets (SSDD and Berkeley DeepDrive), generating 9 anchors. Small anchors (12 ×16, 19 ×36) target distant vehicles, while medium (36 ×75, 76 ×55) and large (135 ×129, 344 ×285) anchors cover near-field and large vehicles. A dynamic IoU threshold (0.4 for small, 0.6 for large anchors) improves matching accuracy.

FC-FPN adjusts shallow P_S and deep P_L feature weights via learnable coefficients (α, β).

Small vehicles use $\alpha = 0.6 - 0.7$, which prioritizes fine-grained details from shallow feature maps. While large vehicles use $\beta = 0.6 - 0.7$, which emphasizes global semantic information from deep feature maps.

MRes2 sub-modules split features into 4 subsets with varying receptive fields. They use 1×1 convolutions, which have small fields, for small targets and 3×3 convolutions, which have larger fields, for medium and large vehicles. Hierarchical fusion ensures simultaneous capture of all scales. This process is enhanced by coordinate attention, or CAM, which provides scale-specific spatial focus.

4. Experiment

This section provides a detailed introduction to the dataset, training details, and evaluation setup. The module is comprehensively evaluated through benchmarking on the SSDD and Berkeley DeepDrive datasets, with comparisons made against existing detection methods. To validate the effectiveness of the module design, a thorough and in-depth analysis of the module is conducted.

4.1. Experimental datasets

SSDD dataset are 700 images featuring a total of 2540 vehicles and pedestrians, with an average of 3.63 images per vehicle. The dataset employs a labeling information processing method similar to PASCAL-VOC. Fig. 5 visualizes the distribution of autonomous driving environment targets. To facilitate the observation of target information in images of different sizes, the center and aspect ratio of the targets are normalized to 0 and 1, respectively. From the visualization of the distribution of autonomous driving environment targets, it can be seen that these targets are scattered randomly across the images without any discernible pattern. The aspect ratios of vehicle targets also vary significantly, highlighting their multi-scale characteristics. As shown in Fig. 5, the SSDD dataset includes 425 small motor vehicle targets, 45 large motor vehicle targets, 125 non-motorized vehicle targets, and 105 pedestrian targets.

The Berkeley DeepDrive dataset scenarios include highways, urban intersections, and municipal roads, comprising 729 images with a total of 2885 vehicles. The autonomous driving environment includes small motor vehicles, non-motorized vehicles, and pedestrians. The image size in Berkeley DeepDrive is 256 × 256, sourced from traffic intersection cameras and vehicle-mounted cameras, with polarization modes including HH, HV, VH, and VV. To visualize the number of different-sized vehicles and pedestrians in the dataset, this paper uses the definition of COCO evaluation metrics to statistically analyze vehicles and pedestrians of varIOUs sizes. There are 1825 small motor vehicle targets, 65 large motor vehicle targets, 450 non-motorized vehicle targets, and 545 pedestrian targets in the Berkeley DeepDrive dataset in Fig. 6. In the Berkeley DeepDrive dataset, most targets are small motor vehicles, while large motor vehicle targets are fewer, making this dataset useful for enhancing the detection capabilities of networks on small motor vehicle targets.

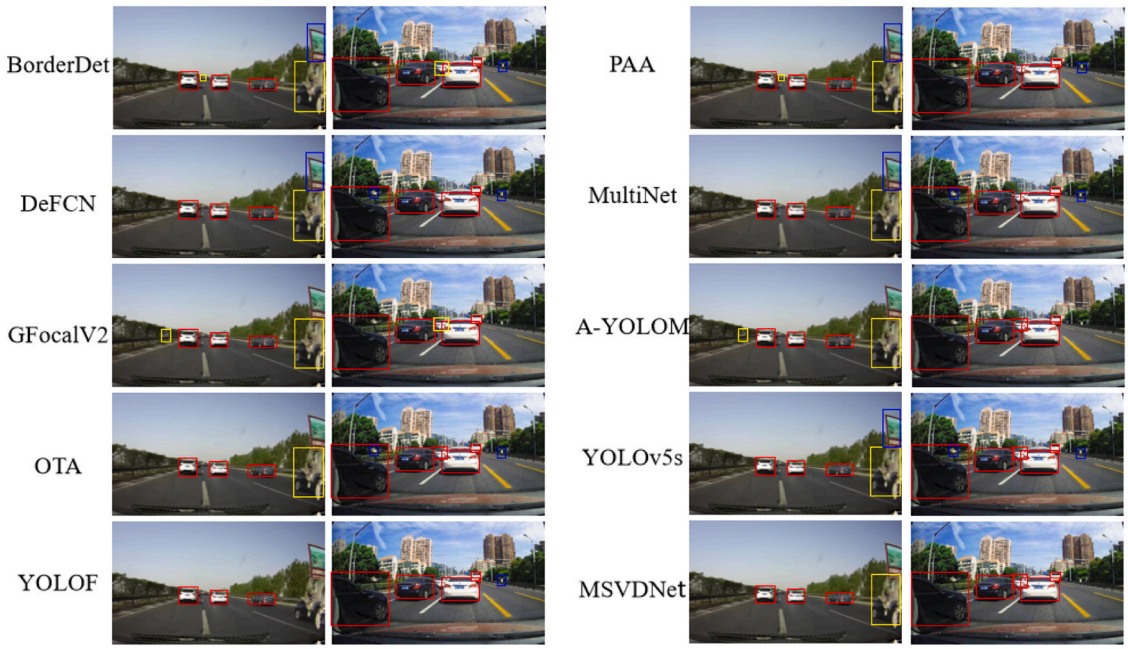

Fig. 8. Visualization of detection results of the SSDD dataset.

Table 3–1
Comparison of AP, AP_{50} and AP_S on the Berkeley DeepDrive dataset.

Methods	AP (%)	95 % CI	AP_{50}(%)	95 % CI	AP_S(%)	95 % CI
BorderDet	56.7	[56.0, 57.4]	93.8	[93.1, 94.5]	49.6	[48.9, 50.3]
DeFCN	54.5	[53.8, 55.2]	93.5	[92.8, 94.2]	49.8	[49.1, 50.5]
GFocalV2	59.3	[58.6, 60.0]	94.7	[94.0, 95.4]	52.3	[51.6, 53.0]
OTA	59.3	[58.6, 60.0]	94.7	[94.0, 95.4]	51.5	[50.8, 52.2]
YOLOF	53.3	[52.6, 54.0]	93.9	[93.2, 94.6]	46.3	[45.6, 47.0]
PAA	56.0	[55.3, 56.7]	91.6	[90.9, 92.3]	51.1	[50.4, 51.8]
MultiNet	53.1	[52.4, 53.8]	89.1	[88.4, 89.8]	51.0	[50.3, 51.7]
A-YOLOM	52.5	[51.8, 53.2]	91.1	[90.4, 91.8]	50.9	[50.2, 51.6]
YOLOv5s	58.6	[57.9, 59.3]	94.6	[93.9, 95.3]	52.8	[52.1, 53.5]
MSVDNet	60.1	[59.5, 60.7]	95.1	[94.5, 95.7]	54.6	[53.9, 55.3]

Table 3–2
Comparison of AP_P, AP_L and AP_{non} on the Berkeley DeepDrive dataset.

Methods	AP_P(%)	95 % CI	AP_L(%)	95 % CI	AP_{non}(%)	95 % CI
BorderDet	65.3	[64.6, 66.0]	57.3	[56.6, 58.0]	62.7	[62.0, 63.4]
DeFCN	61.0	[60.3, 61.7]	47.8	[47.1, 48.5]	61.1	[60.4, 61.8]
GFocalV2	67.8	[67.1, 68.5]	57.0	[56.3, 57.7]	63.2	[62.5, 63.9]
OTA	68.4	[67.7, 69.1]	73.9	[73.2, 74.6]	70.7	[70.0, 71.4]
YOLOF	62.0	[61.3, 62.7]	72.7	[72.0, 73.4]	63.7	[63.0, 64.4]
PAA	65.7	[65.0, 66.4]	73.1	[72.4, 73.8]	63.6	[62.9, 64.3]
MultiNet	65.1	[64.4, 65.8]	61.2	[60.5, 61.9]	60.4	[59.7, 61.1]
A-YOLOM	64.1	[63.4, 64.8]	50.8	[50.1, 51.5]	60.9	[60.2, 61.6]
YOLOv5s	65.6	[64.9, 66.3]	59.4	[58.7, 60.1]	62.2	[61.5, 62.9]
MSVDNet	66.6	[65.9, 67.3]	62.2	[61.5, 62.9]	63.5	[62.8, 64.2]

Note: This table compares the performance of the same set of algorithms on the Berkeley DeepDrive dataset. MSVDNet exhibits statistically significant advantages ($p < 0.05$) in AP (60.1 %) and APS (54.6 %), highlighting its robustness across complex urban scenarios, especially for small-scale vehicle targets.

4.2. Training settings

Training parameters: For both datasets, the detection ranges were set as X-axis [0,70.4] meters, Y-axis [-40,40] meters, and Z-axis [-3,1] meters, with voxel sizes (0.05,0.05,0.10) meters. The ADAM optimizer was employed in a batch size of 16 for 80 epochs using 8 GTX 2080 Ti GPUs, requiring approximately 6 h. The learning rate decayed using cosine annealing with an initial value of 0.001. During the proposal feature extraction phase, a 3D IoU threshold of 0.55 was applied. In the inference stage, non-maximum suppression was used to select the top 100 proposals. Specifically, a 3D IoU threshold of 0.7 was set, and non-maximum suppression filtered out candidate boxes that failed to meet the criteria. After proposal feature extraction, boxes with IoU below 0.01 were excluded from the final results.

In addition, the optimization ranges for key parameters are described as follows. FC-FPN fusion coefficients were initialized to [0.3, 0.7], dynamically adjusted through backpropagation during training, and finally converged to the ranges of $\alpha \in [0.42,\ 0.58]$ and $\beta \in [0.41,\ 0.59]$, ensuring the adaptive balance between shallow and deep features. While learning rate (μ) of the CSPMRes2 module was adopted by a cosine annealing strategy, with the initial value set to 0.001 and decayed to $1e-5$. It is optimized the weight parameters for gradient splitting and reuse to avoid gradient explosion or vanishing.

The data set visualization training process in this study is shown in Fig. 7. The x-axis represents the number of epochs in the training process. The meanings of the y-axis are as follows: (a) is the box loss during training. (b) is the box loss during validating. (c) is the obeject loss during training. (d) is the obeject loss during validating. (e) is the classificationloss during training. (f) is the classification loss during validating. (g) is the recall value during training. (h) is the precision

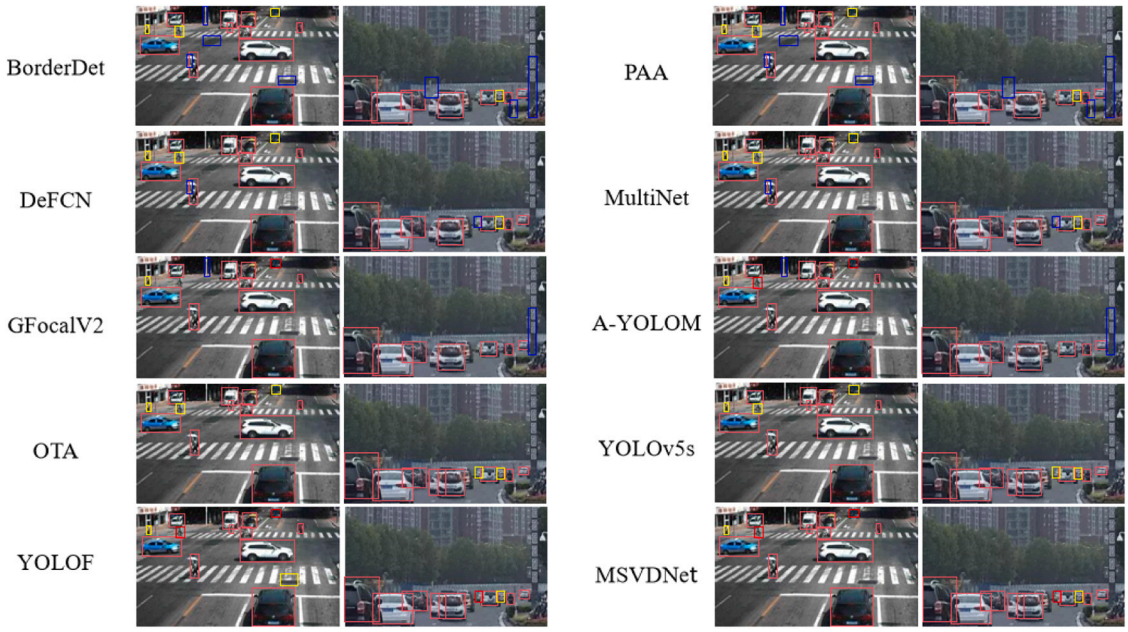

Fig. 9. Visualization of detection results of the Berkeley DeepDrive dataset.

Table 4
Model efficiency comparison.

Methods	AP (%)		Inference time (s)		Model size (MB)
	Berkeley DeepDrive	SSDD	Berkeley DeepDrive	SSDD	
BorderDet	55.7	56.5	42.1	39.7	263.1
DeFCN	54.5	55.5	31.9	29.7	260.9
GFocalV2	59.3	56.2	61.0	62.9	427.3
OTA	59.3	59.1	32.8	29.0	256.2
YOLOF	53.3	59.2	43.5	76.4	368.5
PAA	44.4	56.0	46.0	87.9	1063.2
MultiNet	55.3	54.2	43.0	61.9	367.3
A-YOLOM	62.1	58.8	31.5	30.1	316.2
YOLOv5s	58.6	60.2	1.6	22.3	14.4
MSVDNet	60.1	61.1	3.1	24.2	25.8

Note: This table summarizes efficiency indicators to evaluate the balance between performance and computational cost. MSVDNet maintains high accuracy while achieving lightweight design at 25.8 MB and fast inference at 24.2 ms on SSDD, suitable for real-time applications.

value during validating. (i) is the mAP value with the confidence coefficient of 0.5. (j) is the mAP value with the average confidence coefficient from 0.5 to 0.95.

4.3. Comparison experiments

When comparing results, the methods MultiNet[47], A-YOLOM[48], BorderDet[49], DeFCN[50], GFocalV2[51], OTA[52], YOLOF[53] and PAA[54] are employed. Additionally, since MSVDNet is built on YOLOv5s, the experimental results from YOLOv5s will serve as a baseline to verify the effectiveness of MSVDNet. This paper conducted experiments on some of the latest object detection algorithms using the SSDD dataset, with results shown in Tables 2. Among them, AP is the AP value when the IoU threshold is 0.50:0.05:0.95; AP_{50} is the AP value when the IoU threshold is 0.5; AP_S is the AP value of small motor vehicles; AP_L is the AP value of large motor vehicles; AP_{non} is the AP value of non-motor vehicles; AP_P is the AP value of pedestrians. Compared to other methods, MSVDNet achieved the best results of 61.1 % for AP and 55.4 % for AP_S, fully demonstrating its advantages in overall detection performance and small target detection. MSVDNet did not achieve the best results for AP_P and AP_L, with AP_{non} being 0.1 % lower than OTA and AP_L being 2.3 % lower than YOLOF. The YOLOF algorithm performed best in AP_L but had significantly lower values for AP_S and AP_P compared to MSVDNet, indicating that YOLOF's use of a single feature map for SAR image vehicle target detection does not effectively handle multi-scale issues. The efficient design based on MSVDNet enables it to effectively handle multi-scale vehicle target detection in SAR images; although the participating comparison object detection algorithms achieved excellent detection performance, they did not perform well in SAR image vehicle target detection, suggesting that SAR images vehicle target cannot be directly recognized by the algorithm without adjustments tailored to the characteristics of SAR image vehicle target detection.

To verify the statistical significance of performance differences between MSVDNet and comparative methods, we conducted independent sample t-tests on the AP metrics across 5 repeated experiments. Additionally, 95 % confidence intervals (CIs) for all evaluation metrics are provided to reflect result stability.

To intuitively demonstrate the differences in vehicle target detection between varIOUs algorithms on SAR images, this paper presents

Table 5
Module ablation experiment on Berkeley DeepDrive datasets(%).

Methods	CSPMRes2	FC-FPN	AP	AP_{50}	AP_S	AP_P	AP_L	AP_{non}
MSVDNet	√		60.1	95.1	54.6	66.6	62.2	65.4
		√	59.8	96.1	53.8	69.5	69.3	68.1
	√	√	62.0	94.8	54.3	69.6	68.8	68.3

Note: This table analyzes the impact of individual and combined use of CSPMRes2 and FC-FPN modules on detection performance. MSVDNet achieves the highest AP (62.0 %), confirming their synergistic effect in enhancing multi-scale feature extraction and fusion.

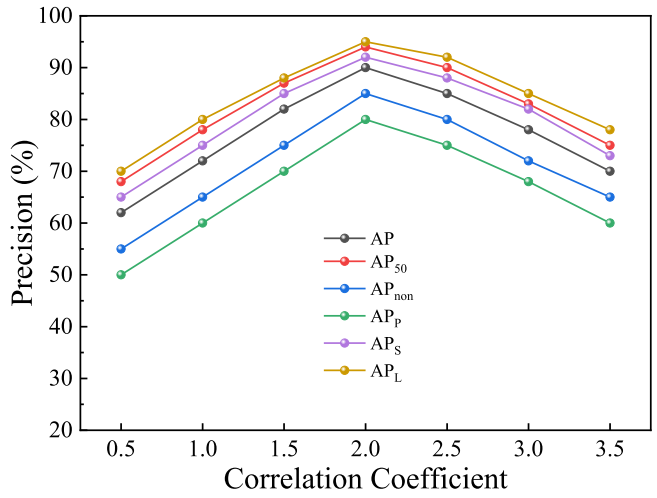

Fig. 10. Accuracy of different fusion coefficient ranges MSVDNet.

visualizations of the detection results from different target detection algorithms on SAR images, as shown in Fig. 8.

From the figures, GFocal V2 have severe missed detections for vehicle targets, while OTA and PAA have severe false detections for vehicle targets, especially when the surrounding environment is complex. All target detection algorithms have issues in detecting targets in near-shore complex backgrounds on SSDD dataset, particularly when vehicle targets are densely arranged, leading to poorer performance. Although all algorithms have performance issues in complex backgrounds, the MSVDNet proposed in this paper performs the best, with fewer false and missed detections compared to other algorithms, indicating that MSVDNet has certain advantages in target detection under complex backgrounds, making its overall performance superior to other methods.

This paper also conducted experiments on the Berkeley DeepDrive dataset using methods that were compared on the SSDD dataset, with experimental results shown in Tables 3.

MSVDNet achieved an AP metric of 60.1 %, surpassing all participating target detection algorithms, indicating that MSVDNet has better overall vehicle target detection performance. The significant improvement in APS metrics suggests that compared to other methods, MSVDNet can still obtain more accurate vehicle position information and has excellent small vehicle target detection capabilities. In terms of APP and APL metrics, the OTA target detection algorithm achieved the best results, far exceeding MSVDNet, indicating that OTA can effectively capture medium and large vehicle targets in the Berkeley DeepDrive dataset. Although the OTA target detection algorithm performs well in AP_P and AP_L metrics, its detection performance is inferior to MSVDNet in other COCO metrics. YOLOF still works well on large vehicle targets, but performs poorly on other metrics. The comparative experiment results show that compared to other methods, MSVDNet has relatively balanced performance in COCO metrics, effectively focusing on multiple evaluation indicators rather than concentrating on specific ones. For example, OTA and YOLOF focus on AP_L metrics, while performing poorly on other metrics.

Statistical analysis confirms that MSVDNet achieves significantly higher AP than all comparative methods on both datasets, particularly in APS. The 95 % CIs for AP of MSVDNet, which shows [60.5,61.7] in SSDD and [59.5,60.7] in Berkeley DeepDrive, are narrower than those of YOLOv5s and OTA, indicating more stable performance. These results validate that the improvements from CSPMRes2 and FC-FPN modules are statistically robust rather than random fluctuations.

In Fig. 9, It is evident that recognizing vehicle targets on edge lanes is particularly challenging. MSVDNet overall detection performance remains significantly better than other methods compared to those involved in the comparison.

This paper compares the parameters of MSVDNet models with other methods. As shown in Table 4, although MSVDNet adds parameters for 11.4MB on top of the YOLOv5s baseline, it still has a significant advantage in parameter volume compared to other methods. This means that MSVDNet can achieve good detection performance with fewer parameters. The paper also compares the inference time of MSVDNet with other methods. Table 4 shows that due to some additional computations added by MSVDNet to achieve better accuracy. On the Berkeley DeepDrive dataset, MSVDNet is 10.3 times faster than DeFCN, which ranks third in inference time, and 1.2 times faster than OTA, which also ranks third in inference time on the SSDD dataset. Compared to other methods, MSVDNet demonstrates an advantage in inference speed.

4.4. Ablation experiments

In the ablation experiments, this paper uses the results of the YOLOv5s experiment as a baseline and validates the effectiveness and FC-FPN modules in MSVDNet using the SSDD dataset. Table 5 shows the CSPMRes2 It can effectively extract multi-scale features in the target image. FC-FPN is a feature pyramid structure with learnable fusion coefficients, capable of simultaneously focusing on vehicle targets of different sizes. The CSPMRes2 alone significantly improves AP_L compared to YOLOv5s. Neither using the CSPMRes2 alone nor using the FC-FPN alone improves AP_S compared to YOLOv5s. This is the primary reason why MSVDNet, which has both CSPMRes2 and FC-FPN modules, achieves lower AP_{50}, AP_L, AP_P and AP_{non} compared to MSVDNet with a single module. MSVDNet can balance the overall performance of vehicle target detection rather than focusing solely on improving certain metrics.

To explore how different ranges of correlation coefficients affect the detection performance of MSVDNet, this paper sets a series of range values for the correlation coefficient and conducts experiments on the Berkeley DeepDrive dataset. Fig. 10 presents the experimental results. The analysis shows that when the correlation coefficient ranges between 0 and 2, MSVDNet demonstrates a linear improvement in vehicle detection performance. However, beyond this threshold, the detection performance begins to decline. This phenomenon indicates that MSVDNet achieves optimal detection performance for various road targets at a correlation coefficient of 2.

4.5. Deployment and hardware adaptability analysis

To verify the practical applicability of MSVDNet in real-time autonomous driving scenarios, this study supplemented hardware

Table 6
Comparison of inference performance on edge devices for autonomous driving.

Device	Model	Inference Latency (ms)	Memory Occupancy (MB)	Power Consumption (W)	FPS
Jetson Nano	YOLOv5s	42.6	896	5.2	23.5
	MSVDNet	51.3	1024	5.8	19.5
Jetson Xavier NX	YOLOv5s	18.3	912	12.1	54.6
	MSVDNet	24.2	1056	13.4	41.3

Note: Tests were conducted under identical conditions (input resolution 640 ×640, batch size=1) using TensorRT-optimized models. Latency includes preprocessing (normalization, resize) and postprocessing (NMS) steps.

deployment tests on edge devices commonly used in autonomous driving, including Jetson Nano (4 GB) and Jetson Xavier NX (8 GB). The tests focused on metrics such as inference latency, memory occupancy, and power consumption, with results shown in Table 6.

MSVDNet achieves a frame rate of 19.5 FPS on Jetson Nano and 41.3 FPS on Jetson Xavier NX, both exceeding the 10 FPS real-time threshold required for autonomous driving perception. Although its latency is approximately 20 % higher than that of YOLOv5s, the 61.1 % AP (compared to 60.2 % for YOLOv5s) represents a worthwhile trade-off in safety-critical scenarios. Its memory occupancy is 1024 MB on Jetson Nano and 1056 MB on Xavier NX, both complying with the memory constraints of edge devices (4 GB and 8 GB respectively). Compared to heavyweight models like GFocalV2 (with memory $>$ 2000 MB), it can avoid out-of-memory errors during deployment. The power consumption of MSVDNet (5.8 W on Nano and 13.4 W on Xavier) is comparable to that of mainstream lightweight models, making it suitable for battery-powered autonomous vehicles with high energy efficiency requirements. These results confirm that MSVDNet maintains a balance between performance and hardware adaptability, filling the gap between high-precision SAR detection and edge deployment in autonomous driving systems.

5. Conclusion

This paper constructs a MSVDNet to detect vehicle targets of different sizes. The MSVDNet with both CSPMRes2 and FC-FPN modules achieves more precise vehicle target localization and balances the detection of multi-scale vehicle targets, thereby improving the overall detection accuracy of multi-scale vehicles in SAR images. According to comparative experimental results on the SSDD and Berkeley DeepDrive datasets, the MSVDNet demonstrates higher overall detection performance compared to other methods, validating the effectiveness of the MSVDNet vehicle target detection network. Additionally, due to the reasonable structural design of the CSPMes2 module, the MSVDNet with the CSPMRes2 module does not add excessive parameters, resulting in lower model complexity and inference time compared to other methods in terms of network capacity and inference speed.

However, MSVDNet has certain limitations. The learning range of the fusion coefficient in MSVDNet is overly dependent on training data, requiring setting the range of the fusion coefficient according to different datasets; otherwise, it may not yield the best results for that dataset. On the other hand, from the visualization of MSVDNet's detection results on both datasets, it can be seen that while MSVDNet can achieve multi-scale vehicle target detection, it lacks specific designs for complex backgrounds and dense vehicle situations, leading to less than ideal detection performance in such scenarios. For complex backgrounds and dense vehicle situations, in addition to designing specialized object detection networks, one can also leverage the characteristics of top-down imaging in remote sensing images, where targets have almost no overlap. Using directional prediction boxes for target localization can minimize environmental information around the targets, ensuring they do not interfere with each other when densely arranged.

CRediT authorship contribution statement

Bingshuo Li: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Xiuhao Hu:** Software, Resources, Project administration, Investigation, Data curation. **Qian Li:** Software, Project administration, Methodology, Conceptualization. **Jian Hu:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Lan Zhang:** Writing – original draft, Visualization, Validation, Supervision, Project administration, Data curation, Conceptualization.

Funding

This work was supported by National Key R&D Program of China (No. 2017YFA0103204).

Declaration of Competing Interest

All authors disclosed no relevant relationships.

References

[1] K. Geng, W. Zou, G. Yin, et al., Low-observable targets detection for autonomous vehicles based on dual-modal sensor fusion with deep learning approach, Proc. Inst. Mech. Eng. Part D J. Automob. Eng. 233 (9) (2019) 2270–2283.
[2] X. Jin, H. Yang, X. He, et al., Robust li DAR-based vehicle detection for on-road autonomous driving, Remote Sens. 15 (12) (2023) 3160.
[3] P. Liu, T. Qu, H. Gao, et al., Driving intention recognition of surrounding vehicles based on a time-sequenced weights hidden Markov model for autonomous driving, Sensors 23 (21) (2023) 8761.
[4] Q. Wu, X. Li, K. Wang, H. Bilal, Regional feature fusion for on-road detection of objects using camera and 3D-LiDAR in high-speed autonomous vehicles, Soft Comput. 27 (23) (2023) 18195–18213.
[5] Y. Wang, K. Zhang, K. Lu, et al., Practical black-box adversarial attack on open-set recognition: towards robust autonomous driving, PeertoPeer Netw. Appl. 16 (1) (2023) 295–311.
[6] H. Bilal, A. Rehman, M.S. Aslam, et al., Hybrid TrafficAI: a generative AI framework for Real-Time traffic simulation and adaptive behavior Modeling, IEEE Trans. Intell. Transp. Syst. (2025).
[7] H. Dou, Y. Liu, S. Chen, et al., A hybrid CEEMD-GMM scheme for enhancing the detection of traffic flow on highways, Soft Comput. 27 (21) (2023) 16373–16388.
[8] Z. Tian, C. Shen, H. Chen, et al., Fcos: fully convolutional one-stage object detection, Proc. IEEE CVF Int. Conf. Comput. Vis. (2019) 9627–9636.
[9] K. Duan, S. Bai, L. Xie, et al., Centernet: keypoint triplets for object detection, Proc. IEEE CVF Int. Conf. Comput. Vis. (2019) 6569–6578.
[10] Y. Wang, V.C. G, T. Zhang, DETR3D: 3D Object Detection from Multi-view Images via 3D-to-2D Queries [EB/OL]. arXiv - CS - Artificial Intelligence, 2021: 06922 uizilini2021.
[11] C.R. Qi, H. Su, K. Mo, et al., Pointnet: deep learning on point sets for 3D classification and Segmentation, IEEE Conf. Comput. Vis. Pattern Recognit. (2017) 652–660.
[12] A.T. Elgohr, M.S. Elhadidy, M. Elazab, et al., Multi-Classification model for brain tumor early prediction based on deep learning Techniques, J. Eng. Res. 8 (3) (2024).
[13] M.S. Elhadidy, A.T. Elgohr, M. El-geneedy, et al., Comparative analysis for accurate multi-classification of brain tumor based on significant deep learning models, Comput. Biol. Med. 188 (2025) 109872.
[14] S.S. Shi, X.G. Wang, H.S. Li, Pointrcnn: 3D object proposal generation and detection from point cloud, Comput. Vis. Pattern Recognit. (2019) 770–779.
[15] W.J. Shi, R. Rajkumar, Point-GNN: graph neural network for 3D object detection in a point cloud, Comput. Vis. Pattern (2020) 1711–1719.
[16] Z.T. Yang, Y.N. Sun, S. Liu, et al., 3DSSD: Point-based 3D single stage object Detector, Comput. Vis. Pattern Recognit. (2020) 11040–11048.
[17] Z.T. Yang, Y.N. Sun, S. Liu, et al., STD: Sparse-to-Dense 3D object detector for point cloud, Comput. Vis. (2019) 1951–1960.
[18] Y. Zhou, O. Tuzel, Voxelnet: End-to-End learning for point cloud based 3D object Detection, Comput. Vis. Pattern Recognit. (2018) 4490–4499.
[19] Y. Yan, Y. Mao, B. Li, SECOND: sparsely embedded convolutional detection, Sensors 18 (10) (2018) 3337.
[20] A.H. Lang, S. Vora, H. Caesar, et al., Pointpillars: fast encoders for object detection from point Clouds, Comput. Vis. Pattern Recognit. (2019) 12697–12705.
[21] T.W. Yin, X.Y. Zhou, P. Krahenbuhl, Center-based 3D object detection and Tracking, Comput. Vis. Pattern Recognit. (2021) 11784–11793.
[22] J.J. Deng, S.S. Shi, P.W. Li, et al., Voxel R-CNN: towards high performance Voxel-based 3D object Detection, Artif. Intell. (2021) 1201–1209.
[23] S.S. Shi, Z. Wang, J.P. Shi, et al., From points to parts: 3D object detection from point cloud with Part-aware and Part-aggregation network, Pattern Anal. Mach. Intell. 43 (8) (2020) 2647–2664.
[24] Q. Zheng, S. Xu, C. Liu, et al., Real-time lightweight target detection network under autonomous Driving, J. Phys. Conf. Ser. (1) (2023) 012003.
[25] X. Zhang, M. Wang, Research on pedestrian tracking technology for autonomous driving Scenarios, IEEE Access 10 (2024).
[26] R. Huo, J. Chen, Y. Zhang, et al., 3D skeleton aware driver behavior recognition framework for autonomous driving system, Neurocomputing 613 (2025) 128743.
[27] M. Jin, X. Wang, C. Guo, et al., Research on target detection for autonomous driving based on ECS-spiking neural networks, Sci. Rep. (2025).
[28] A. Ali, I. Ullah, S.K. Singh, et al., Energy-efficient resource allocation for urban traffic flow prediction in edge-cloud computing, Int. J. Intell. Syst. (2025).
[29] A. Ali, I. Ullah, S. Ahmad, et al., An attention-driven spatio-temporal deep hybrid neural networks for traffic flow prediction in transportation Systems, IEEE Trans. Intell. Transp. Syst. (2025).

[30] A. Ali, I. Ullah, M. Shabaz, et al., A resource-aware multi-graph neural network for urban traffic flow prediction in multi-access edge computing Systems, IEEE Trans. Consum. Electron. 70 (4) (2024).
[31] H. Jeong, J. Shin, F. Rameau, et al., Multi-Modal place recognition via vectorized HD maps and images fusion for autonomous driving, IEEE Robot. Autom. Lett. 6 (2024).
[32] Y. Morio, Y. Hanada, Y. Sawada, et al., Field scene recognition for self-localization of autonomous agricultural vehicle, Eng. Agric. Environ. Food 12 (3) (2019) 325–340.
[33] Z. Guo, C. Liu, X. Zhang, et al., Beyond bounding-box: Convex-hull feature adaptation for oriented and densely packed object detection, Proc. IEEE CVF Conf. Comput. Vis. Pattern Recognit. (2021) 8792–8801.
[34] P. Dai, S. Yao, Z. Li, et al., ACE: Anchor-free corner evolution for real-time arbitrarily-oriented object detection, IEEE Trans. Image Process. 31 (2022) 4076–4089.
[35] J. Shi, Q. Zhang, Q. Shi, et al., Pedestrian pose recognition based on frequency-modulated continuous-wave radar with meta-learning, Sensors 24 (9) (2024) 2932.
[36] L. Wang, S. Hua, C. Zhang, et al., YOLOdrive: a lightweight autonomous driving Single-Stage target detection Approach, IEEE Internet Things J. 7 (2024).
[37] A. Ali, I. Ullah, S.K. Singh, et al., Attention-Driven graph convolutional networks for Deadline-Constrained virtual machine task allocation in edge Computing, IEEE Trans. Consum. Electron. (2025).
[38] C. Brewitt, M. Tamborski, C. Wang, et al., Verifiable goal recognition for autonomous driving with occlusions, IEEE RSJ Int. Conf. Intell. Robots Syst. (2023) 11210–11217.
[39] Y. Hou, C. Wang, J. Wang, et al., Visual evaluation for autonomous Driving, IEEE Trans. Vis. Comput. Graph. 28 (1) (2021) 1030–1039.
[40] H. Liu, G. Li, M. Li, et al., High-precision real-time autonomous driving target detection based on YOLOv8, J. Real. Time Image Process. 21 (5) (2024) 174.
[41] Q. Zhou, C. Yu, Point rcnn: an angle-free framework for rotated object detection, Remote Sens. 14 (11) (2022) 2605.
[42] Y. Liu, Z. Wang, M. Cai, et al., A hybrid target selection model of functional safety compliance for autonomous driving System, ACM Trans. Embed. Comput. Syst. (2025).
[43] J. Ma, G. Xiong, J. Xu, et al., CVTNet: a Cross-View transformer network for LiDAR-Based place recognition in autonomous driving Environments, IEEE Trans. Ind. Inform. 20 (3) (2023) 4039–4048.
[44] X. Yang, X. Yang, J. Yang, et al., Learning high-precision bounding box for rotated object detection via kullback-leibler divergence, Adv. Neural Inf. Process. Syst. 34 (2021) 18381–18394.
[45] W. Qian, X. Yang, S. Peng, et al., Learning modulated loss for rotated object detection, Proc. AAAI Conf. Artif. Intell. (2021) 2458–2466.
[46] Z. Chen, K. Chen, W. Lin, et al., PIOU loss: towards accurate oriented object detection in complex environments. Proceedings of the European Conference on Computer Vision, Springer, 2020, pp. 195–211.
[47] T. Marvin, Michael Weber, Marius Zoellner, et al., Multinet: Real-time joint semantic reasoning for autonomous driving, Veh. Symp. (2018) 1013–1020.
[48] S. Chen, J. Xu, J. Yu, et al., Automatic abdominal hernia mesh detection based on YOLOM, IEEE Access (2022) 31420–31431.
[49] H. Qiu, Y. Ma, Z. Li, et al., Borderdet: Border Feature for Dense Object Detection. Computer Vision, Springer, 2020, pp. 549–564.
[50] J. Wang, L. Song, Z. Li, et al., End-to-end object detection with fully convolutional network, Comput. Vis. Pattern Recognit. (2021) 15849–15858.
[51] X. Li, W. Wang, X. Hu, et al., Generalized focal loss v2: learning reliable localization quality estimation for dense object detection, Comput. Vis. Pattern Recognit. (2021) 11632–11641.
[52] Z. Ge, S. Liu, Z. Li, et al., Ota: optimal transport assignment for object detection, Proc. IEEE CVF Conf. Comput. Vis. Pattern Recognit. (2021) 303–312.
[53] T. Xue, Z. Liu, S. Lan, et al., YOLO-FSE: an improved target detection algorithm for vehicles in autonomous Driving, IEEE Internet Things J. 6 (2025).
[54] K. Kim, H.S. Lee, Probabilistic anchor assignment with IOU prediction for object detection. Proceedings of the European Conference on Computer Vision, Springer, 2020, pp. 355–371.