2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

DFRNET-H: DYNAMIC FEATURE REFINEMENT NETWORK WITH HETEROGENEOUS KERNELS AND WEIGHTED FUSION FOR HIGHWAY MONITORING

Xu Liu, Wei Han†, Siren Batu, Peng Zhang, Kai Liu and Ming Ma*

Abstract— **Highway vehicle detection remains challenging due to scale variation, motion blur, and frequent occlusions. While YOLO-based detectors meet real-time demands, their feature extraction and fusion remain limited in complex traffic scenes. To address this, we propose DFRNet-H (Dynamic Feature Refinement Network), which integrates three lightweight modules: CSP-MSPA enhances small-object representation with fractal partial convolution, RDA enlarges receptive fields through recursive dilated aggregation, and AFPN adaptively reweights multi-scale features for efficient fusion. On the UA-DETRAC benchmark dataset, DFRNet-H achieves a +4.4% improvement over YOLOv11-N on mAP_{50-95}, and on our self-constructed Highway Vehicle Detection (HVD) dataset it achieves a further +2.5% gain. These results demonstrate that DFRNet-H effectively balances accuracy and efficiency under complex highway scenarios.**

I. INTRODUCTION

With the rapid growth of intelligent transportation and autonomous driving, highway vehicle detection has become essential for traffic safety and efficiency. However, highway scenes pose challenges such as high vehicle speed, large scale variation, distant blurred small objects, and complex illumination or weather, making accurate real-time detection difficult.

To address these challenges, we propose DFRNet-H, a dynamic feature refinement network. It leverages cross-layer information flow and dynamic feature fusion with heterogeneous convolutions, frequency enhancement, and context-adaptive modeling, achieving robust small-object detection while maintaining lightweight efficiency.

Our main contributions are:

This work was supported in part by the Natural Science Foundation of China (No.62462049); Research Project on Strengthening the Construction of Important Ecological Security Barrier in Northern China by Higher Education Institutions in Inner Mongolia Autonomous Region under Grant No.STAQZX202321; and the 'Jiebang Guashuai' Science and Technology Project of Inner Mongolia Transportation Group.

†These authors contributed equally with the first author. *Corresponding author.

Xu Liu is with the Computer Science Faculty, Inner Mongolia University, Hohhot, China 32309160@mail.imu.edu.cn

Wei Han is with the Computer Science Faculty, Inner Mongolia University, Hohhot, China 32409210@mail.imu.edu.cn

Siren Batu is with the Computer Science Faculty, Inner Mongolia University, Hohhot, China csbatu@imu.edu.cn

Peng Zhang is with the Inner Mongolia Transportation Group Digital Logistics Technology Co., Ltd., Hohhot, China 921500621@qq.com

Liu Kai is with the Inner Mongolia Transportation Group Digital Logistics Technology Co., Ltd., Hohhot, China lk@nmgmmzy.com

Ming Ma is with the Computer Science Faculty, Inner Mongolia University, Hohhot, China csmaming@imu.edu.cn

(1) CSP Multi-Scale Partial Aggregation (CSP-MSPA): A fractal cascaded multi-scale convolution module that enhances small-object representation with high efficiency.

(2) Recursive Dilated Aggregation (RDA): A recursive dilated convolution module with weight sharing, integrating local details and global context for robust features.

(3) Adaptive Feature Pyramid Network (AFPN): A recursive cross-layer fusion framework with heterogeneous kernels, enabling adaptive frequency-domain modeling across scales.

(4) Dataset construction: We build a new highway vehicle detection dataset, Highway Vehicle Detection (HVD), covering diverse traffic conditions. This dataset supports evaluation where DFRNet-H surpasses mainstream detectors in small-object accuracy, robustness, and real-time performance.

II. RELATED WORK

A. Object Detection

Single-stage detectors dominate for their efficiency, with the YOLO series evolving rapidly: YOLOv8 introduces an anchor-free head, YOLOv9 [6] and YOLOv10 [7] redesign assignment and prediction, and YOLOv11–13 [8], [9] enhance feature fusion and multi-scale modeling. GOLD-YOLO [10] further improves localization robustness. Transformer-based detectors (e.g., DETR [2], [11], DEIM [12]) explore end-to-end paradigms, though challenges remain in convergence and small-object detection. Overall, recent trends emphasize lightweight design, multi-scale fusion, and global–local collaboration, motivating our DFRNet-H.

B. Feature Pyramid Network

Feature Pyramid Networks (FPN) [13] and their variants (e.g., PANet [14], BiFPN [15]) are widely used for multi-scale fusion. Conventional FPNs integrate semantics across layers but face two issues: (1) long propagation paths hinder gradient flow and feature utilization; (2) fixed convolutions (e.g., 3×3) lack flexibility for diverse contexts. BiFPN alleviates these with weighted bidirectional fusion, yet still relies on standard convolutions. Similarly, YOLO's necks (e.g., PAN-FPN in YOLOv5, C2f-PAN in YOLOv8) optimize fusion paths but remain limited by fixed kernels, restricting adaptive perception in highway scenarios with large scale variations.

III. METHOD

In highway vehicle detection, large scale variation, high speed, and occlusion hinder accurate recognition. Although YOLO excels in real-time detection, it struggles with distant small objects and suboptimal feature fusion. To address this, we redesign the network from three aspects: (1) a lightweight multi-scale cascaded backbone module to enhance small-object representation; (2) efficient feature aggregation via shared and dilated convolutions; and (3) an adaptive multi-branch feature pyramid with weighted fusion. These modules jointly improve accuracy and robustness while preserving efficiency.

A. CSP Multi-Scale Partial Aggregation Module

In highway scenarios, distant small objects remain challenging. YOLOv11's C3K2 structure improves large-object detection by expanding the receptive field, but deep stacking weakens small-object features. To address this, we redesign the backbone and propose the CSP Multi-Scale Partial Aggregation (CSP-MSPA), which enhances distant small-object detection while maintaining efficiency and lightweight design (Fig. 1).

The CSP-MSPA module consists of a 1×1 convolution and a Multi-Scale Partial Aggregation Block (MSPA Block), where the MSPA Block serves as the core component. Given an input feature $x \in \mathbb{R}^{C \times H \times W}$, a 3×3 convolution Conv_3 is first applied to perform initial channel-wise fusion, enhancing the global semantic representation of the input. The convolution output is then split along the channel dimension: one part is fed into the next convolution layer for deeper feature extraction, while the other part is preserved to maintain shallow information. This channel-splitting strategy is specifically designed to optimize gradient flow and minimize computational redundancy, following the Cross Stage Partial (CSP) design philosophy. Unlike traditional feature reuse mechanisms that propagate full feature maps at each stage—which would lead to a substantial increase in parameters and FLOPs without introducing significantly new information—our approach ensures that each convolutional branch focuses on a unique subset of features. This effectively prevents the accumulation of redundant information across scales while keeping the model lightweight.

The feature extraction process proceeds progressively across multiple scales. First, the input x passes through a 3×3 convolution, producing $conv1\_out$, which is split into two parts: $conv1\_out\_1$, fed into the next convolutional layer for deeper feature extraction, and $conv1\_out\_2$, retained as shallow features for later fusion. Next, $conv1\_out\_1$ is processed by a 5×5 group convolution Conv_5, yielding $conv2\_out$. This output is similarly divided into $conv2\_out\_1$ and $conv2\_out\_2$, where the former continues for deeper processing while the latter preserves mid-level information. Finally, $conv2\_out\_1$ is passed through a 7×7 group convolution Conv_7, producing the deep feature $conv3\_out$ and forming a progressive extraction pathway from global to local scales.

After multi-scale feature extraction, the deep features $conv3\_out$ are concatenated with the preserved features $conv2\_out\_2$ and $conv1\_out\_2$ along the channel dimension. A 1×1 convolution Conv_1 is then applied for channel integration, producing an aggregated feature f. Finally, f is fused with the input x via residual addition:

$$f = \text{Conv}_{1\times1}\big(\text{Concat}[conv3\_out,\ conv2\_out\_2,\ conv1\_out\_2]\big). \tag{1}$$

It is worth noting that each convolution operates only on a portion of the input channels. Group convolution ensures independence across scales while reducing computational redundancy, and the channel split-and-concatenate strategy guarantees effective fusion of shallow and deep features.

Subsequently, a channel attention mechanism is introduced. Using the sigmoid activation $\sigma(\cdot)$, channel weights w are generated and multiplied with f to adaptively enhance salient information:

$$\hat{f} = f \otimes \sigma(f), \tag{2}$$

where $\otimes$ denotes channel-wise multiplication. Specifically, the $\times$ and $+$ symbols in the MSPA Block represent the element-wise multiplication for attention weighting and the residual addition for feature fusion, respectively. Finally, a residual connection adds the refined features $\hat{f}$ to the input x, yielding the output:

$$y = \hat{f} + x. \tag{3}$$

This design enables progressive modeling of global-to-local multi-scale features while leveraging attention to capture channel importance and suppress redundancy. By integrating the CSP structure with multi-scale group convolutions, CSP-MSPA reduces computational cost, preserves feature independence, and enriches scale representation. As a result, the module achieves a balance between multi-scale feature learning and efficiency, improving the perception of small, distant, and complex-scene targets.

B. Recursive Dilated Aggregation Module

The SPPF (Spatial Pyramid Pooling–Fast [17]) module fuses multi-scale features to improve recognition of objects at different sizes, using a single pooling kernel applied sequentially. However, max pooling discards informative cues and disrupts spatial continuity [18]. To address this, we redesign the structure with Recursive Dilated Aggregation (RDA) using recursive dilated convolutions and weight sharing, enhancing multi-scale object perception (Fig. 2).

The RDA module begins with a 1×1 convolution applied to the input feature map $X \in \mathbb{R}^{C_1 \times H \times W}$, which compresses the channel dimension to a hidden size

$$C_h = \frac{C_1}{2}, \tag{4}$$

thereby reducing computational cost and alleviating redundant features. The compressed feature serves as the initial element in the feature list Y for recursive processing.

Subsequently, the RDA module adopts a recursive dilated convolution structure. Specifically, let the dilation rate list

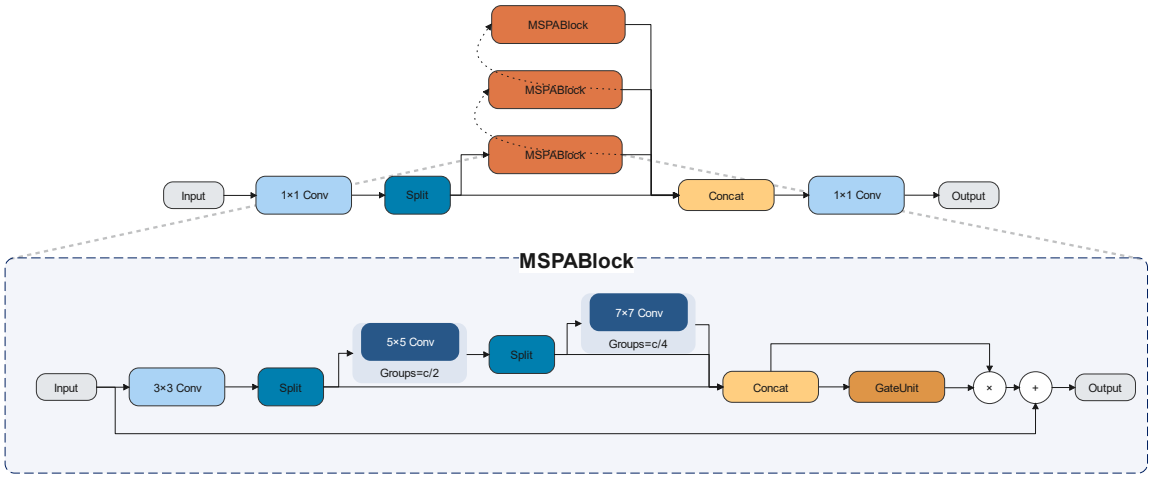


Fig. 1: CSP Multi-Scale Partial Aggregation

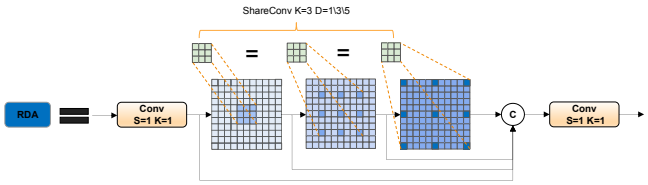


Fig. 2: Recursive Dilated Aggregation

be $d_i \in \{1, 3, 5\}$ and the number of recursive layers be n. At the i-th iteration, the module applies the same 3×3 convolution kernel to the previous feature y_{i-1} with dilation d_i, formulated as:

$$y_i = \text{ShareConv}(y_{i-1}; w, d = d_i, p = p_i), \tag{5}$$

where the stride is set to 1, w denotes the shared convolutional parameters, and the padding size is $p_i = d \cdot (3 - 1) + 1/2$. The receptive field is expanded by varying the dilation rate while keeping the convolution kernel fixed. This ensures that features extracted at different dilation rates remain aligned in the same feature space, maintaining semantic consistency across scales. Moreover, parameter sharing avoids the linear growth of parameters with depth, significantly reducing computation and model complexity compared with traditional ASPP modules.

Since the same convolutional kernel repeatedly "observes" the input at different scales, the model captures hierarchical representations ranging from local details to global context while preserving consistent discriminative patterns. Unlike parallel multi-branch convolutions that often yield inconsistent targets, the recursive and weight-sharing strategy enhances the compactness and composability of feature representations. Through recursive iterations, each layer not only leverages the representations of the previous one but also incorporates receptive fields of different scales, forming a progressive feature extraction strategy from local to global. This design enables the module to expand the receptive field while avoiding inconsistencies across branches.

After multi-scale feature extraction, the outputs of all recursive convolutions are concatenated along the channel dimension, followed by a 1×1 convolution for dimensionality expansion and integration:

$$Y_{\text{out}} \in \mathbb{R}^{C_2 \times H \times W}. \tag{6}$$

As defined in Eq. 6, the resulting feature representation matches the target output channels. In this way, the RDA module achieves efficient multi-scale feature aggregation within a lightweight design, balancing fine-grained local structures with high-level semantic context.

The proposed design offers several advantages. First, recursive dilated convolutions capture multi-scale features by leveraging varying dilation rates, which strengthens the network's ability to detect both small and distant objects. Second, by sharing convolutional kernels across recursive iterations, the module remains lightweight and significantly reduces parameter count and computational cost, making it more efficient than conventional ASPP structures. Third, the recursive formulation enforces feature compactness and consistency across layers, thereby avoiding conflicts among parallel branches. Finally, with appropriate padding and recursive propagation, the module preserves the spatial resolution of the input, ensuring that structural information is retained throughout the feature extraction process.

In summary, this study replaces the C3K2 module in YOLOv11 with the CSP-MSPA module and substitutes the SPPF module with the RDA module, thereby constructing a robust backbone network tailored for feature extraction under complex highway environments. The architecture of the backbone network is illustrated in Fig. 3.

C. Adaptive Feature Pyramid Network

Feature pyramids have been widely adopted for multi-scale feature fusion, as single-scale feature maps cannot simultane-

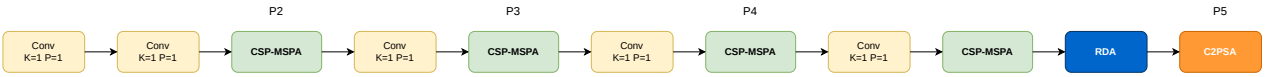


Fig. 3: Overall architecture of the proposed backbone network, where the C3K2 module in YOLOv11 is replaced with CSP-MSPA and the SPPF module is replaced with RDA.

ously represent both small and large objects. Shallow features provide high spatial resolution but lack semantic abstraction, while deep features are semantically rich yet suffer from spatial detail loss. By cross-layer fusion, high-resolution spatial details and high-level semantic information can be preserved within the same feature hierarchy, constructing an approximately scale-invariant joint representation that significantly enhances robustness to extreme scale variations.

BiFPN extends traditional feature pyramids by introducing weighted bidirectional fusion, where a learnable weighting mechanism enables more efficient top-down and bottom-up information flow. However, BiFPN primarily optimizes fusion paths, while feature processing at individual nodes still relies on conventional convolution operations, which remain insufficient for high-speed scenarios with large scale discrepancies among targets.

To address these limitations, we design an Adaptive Feature Pyramid Network (AFPN) to replace the PAN structure in YOLOv11's neck. Inspired by BiFPN, our method enhances cross-scale interactions in the neck, mitigating the insufficient feature utilization observed in conventional YOLO architectures. Specifically, our design incorporates three types of information flows: top-down, bottom-up, and cross-scale lateral fusion. This multi-path interaction effectively bridges the semantic–spatial gap across layers, enabling efficient cross-level feature compensation and enhancement. The overall architecture is shown in Fig. 4.

Top-down pathway: Starting from high-level semantic features, global contextual information is progressively propagated to shallower layers through up-sampling and alignment, injecting semantic representations into higher-resolution feature maps.

Bottom-up pathway: In contrast, the bottom-up pathway begins with shallow high-resolution features. Through convolutional down-sampling and multi-scale aggregation, spatial details such as edges and textures are gradually transmitted to deeper layers, compensating for the fine-grained information typically lost in semantic-rich representations.

Cross-scale lateral fusion: To further enhance inter-scale connectivity, multiple lateral fusion nodes are introduced at intermediate levels, forming multi-source aggregation hubs. Dynamic weighting is applied to explicitly model the relative importance of different resolution features, improving the global consistency of fused representations.

The feature fusion strategy in FPN-based networks largely determines the overall representational capacity and detection performance. To this end, we introduce learnable fast normalized fusion weights at each fusion node, which explicitly model the relative importance of input features at different resolutions.

Specifically, suppose a fusion node receives n input features $\{x_1, x_2, \ldots, x_n\}$ with corresponding fusion weights $\{w_1, w_2, \ldots, w_n\}$. To ensure non-negativity, the weights are first activated by ReLU:

$$\hat{w}_i = \mathrm{ReLU}(w_i), \quad i = 1, 2, \ldots, n. \tag{7}$$

Next, to avoid numerical instability and to normalize relative contributions, we apply l_1-normalization:

$$\tilde{w}_i = \frac{\hat{w}_i}{\sum_{j=1}^{n} \hat{w}_j + \epsilon}, \quad \epsilon = 10^{-4}. \tag{8}$$

Finally, the fused output feature is expressed as

$$y = \sum_{i=1}^{n} \tilde{w}_i \cdot x_i. \tag{9}$$

As shown in Eq. 9, this process eliminates the computational overhead of traditional softmax operations while efficiently modeling the importance distribution of input features.

Moreover, we embed this fusion module in a residual manner along both the top-down and bottom-up pathways:

$$y_{\text{out}} = y + x_{\text{skip}}, \tag{10}$$

where x_{skip} denotes the shortcut input. As indicated in Eq. 10, this residual design preserves low computational cost while mitigating gradient vanishing, thereby enabling effective interaction between high-level semantics and low-level spatial details.

After feature fusion, we introduce the CSP-MSCB (Cross Stage Partial Multi-Scale Convolutional Block) to strengthen the representation capacity of each fusion node. This block combines the CSP structure with the Multi-Scale Convolutional Block (MSCB). MSCB captures both local texture details and global context by applying multiple convolution kernels of different sizes (e.g., 1×1, 3×3, 5×5) in parallel or stacked form, thereby enriching multi-scale representations at each node. In parallel, the CSP structure directly bypasses part of the input features to the output, enabling optimized cross-layer gradient flow. This design retains deep semantic information while preserving shallow spatial details, ensuring that fused features are sufficiently enhanced across scales.

However, due to the significant resolution differences across feature levels, traditional upsampling operations (e.g.,

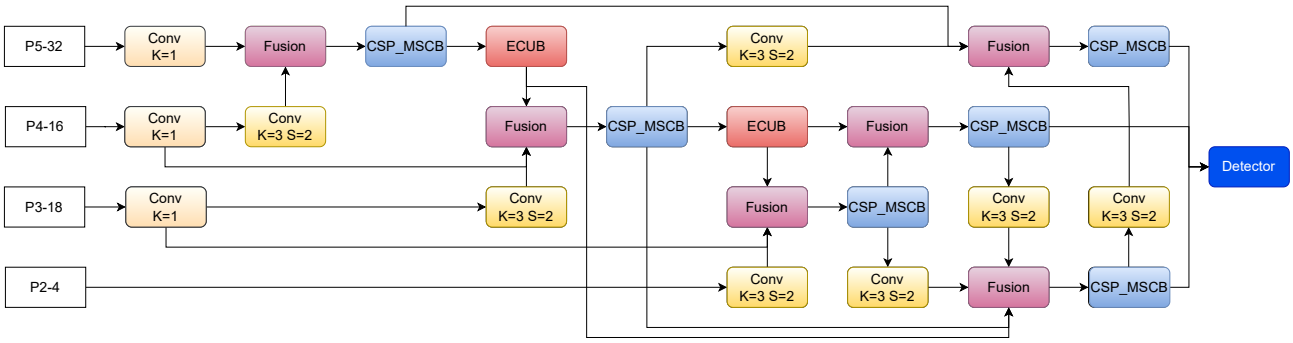


Fig. 4: Adaptive Feature Pyramid Network

nearest-neighbor or bilinear interpolation) merely enlarge spatial resolution without performing convolutional refinement or channel adjustment. Such operations result in simple interpolation, lacking feature enhancement, and may even introduce spatial distortions while increasing computational overhead. To address this issue, we incorporate the EUCB (Efficient Upsample Convolutional Block) [21] into the node enhancement process.

The EUCB first upsamples low-resolution features using bilinear interpolation, then applies depthwise separable convolution to reconstruct spatial and channel information, followed by a 1×1 convolution and channel shuffle operation to reorganize and balance feature channels. Formally, the EUCB output is defined as

$$Y = \mathrm{Conv}_{1\times1}\Big(\mathrm{Shuffle}\big(\mathrm{DWConv}\big(\mathrm{Upsample}(X)\big)\big)\Big), \quad (11)$$

where the components are defined as follows: $\mathrm{Upsample}(X)$ denotes bilinear interpolation, which enlarges low-resolution features by a factor of 2 to match high-resolution feature scales. $\mathrm{DWConv}(\cdot)$ represents depthwise separable convolution that extracts spatial information per channel while keeping the computational cost low. $\mathrm{Shuffle}(\cdot)$ indicates the channel shuffle operation, which reorders feature channels to enhance inter-scale interaction along the channel dimension. Finally, $\mathrm{Conv}_{1\times1}(\cdot)$ is a 1×1 convolution that integrates channel information and adjusts the output dimensionality.

As shown in Eq. 11, EUCB preserves the spatial structure of features while reducing computational complexity, and further improves the efficiency of fusing upsampled features with high-resolution shallow features. By integrating CSP-MSCB for multi-scale feature enhancement and BiFPN-style bidirectional fusion with EUCB, the AFPN achieves lightweight and real-time performance while enabling efficient interaction between high-level semantics and low-level spatial details.

Overall, the proposed AFPN demonstrates strong adaptability in highway scenarios: it effectively captures fine-grained details for distant small vehicles, leverages contextual modeling for large nearby trucks, and ensures efficient multi-scale feature interactions through weighted fusion. Experimental results validate that AFPN significantly improves detection accuracy across varying object scales while maintaining real-time inference speed.

IV. EXPERIMENTS

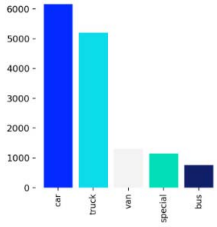
A. Dataset and Implementation Details

1) Details: All experiments are conducted on a workstation running Ubuntu 20.04.6 with the PyTorch 2.0.1 framework (CUDA 12.8) and two NVIDIA RTX 4090D GPUs (24 GB each). The models are trained from scratch without loading any pretrained weights for 300 epochs with a total batch size of 16, using a fixed random seed of 1. The optimizer is stochastic gradient descent (SGD) with momentum 0.9 and a weight decay of 5×10^{-4}. For data augmentation, we employ color jittering, random flipping, random cropping, Mosaic, and Mixup.

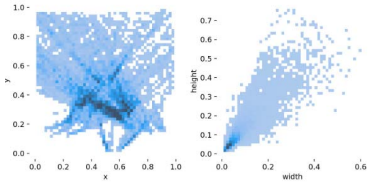
2) Dataset: We constructed a dedicated HVD dataset with authorization from the Inner Mongolia Transportation Group, using surveillance cameras mounted 6–6.5 meters above the ground. The cameras captured video streams at 25 FPS under diverse conditions, including daytime, nighttime, tunnel, rain, fog, snow, sandstorm, and strong wind. Frames were sampled every 50 frames, resulting in a total of 10,000 images at resolutions of 880×720 and 1272×720. Vehicles were categorized into five classes: *car*, *van*, *truck*, *bus*, and *spv* (special-purpose vehicle). Annotations were created manually using the open-source tool DarkLabel, and exported in VOC format. The dataset was split into training, validation, and test sets at a ratio of 7:2:1, ensuring no data leakage. As shown in Fig. 5, compared to the UA-DETRAC benchmark dataset, our dataset contains more small-scale and distant vehicles under complex scenarios, making it well suited for comprehensive highway detection evaluation. To improve model generalization, we further employ the UA-DETRAC dataset, which contains over 1.4M annotated instances from 10K+ traffic videos under diverse conditions. Frames are sampled every 30 frames, yielding 8,187 images categorized into four classes: *bus*, *car*, *van*, and *others*. Following the same 7:2:1 ratio, the dataset is divided into training, validation, and test sets. To ensure reproducibility and respect the distinct taxonomies of each source, the two datasets are utilized independently. Specifically, separate models are

(a) Vehicle Categories

(b) Representative real-world traffic scenes with diverse environments and vehicle types



(c) Distribution of Vehicle Numbers



(d) Spatial and Size Distribution of Bounding Boxes

Fig. 5: Illustration of the HVD dataset: (a) examples of vehicle categories, (b) representative real-world traffic scenes, (c) distribution of vehicle numbers, and (d) distribution of bounding box positions (left) and width/height (right).

TABLE I: Comparison of YOLOv11-N with CSP-MSPA backbone improvement on the HVD dataset.

Method	Params (M)	FLOPs (G)	mAP_{50}	mAP_{50-95}
YOLOv11-C3K2	2.6	6.4	88.0	72.5
YOLOv11-CSP-MSPA	2.6	7.6	**89.9**	**73.9**

trained and evaluated on each dataset according to their original label definitions. While the network architecture and training protocols remain identical across all experiments, no label alignment or cross-dataset joint training is performed. This approach ensures that the performance gains on each benchmark reflect the intrinsic effectiveness of the proposed architectural improvements rather than any data manipulation. Together, these two datasets provide a comprehensive basis for evaluating both the generalization ability and practical applicability of the proposed method.

B. Experimental Results

1) Module Comparison: To assess the contribution of each module, we extended the YOLOv11-N baseline with CSP-MSPA, AFPN, and RDA, and conducted experiments on the HVD dataset.

TABLE II: Comparison of YOLOv11-N with AFPN and BiFPN Neck improvements on the HVD dataset.

Method	Params (M)	FLOPs (G)	mAP_{50}	mAP_{50-95}
YOLOv11-PAN	2.6	6.4	88.0	72.5
YOLOv11-BiFPN [15]	2.7	6.6	89.1	73.3
YOLOv11-AFPN	2.1	6.6	**89.3**	**73.8**

TABLE III: Comparison of YOLOv11-N with RDA Neck improvement on the HVD dataset.

Method	Params (M)	FLOPs (G)	mAP_{50}	mAP_{50-95}
YOLOv11-SPPF	2.6	6.4	88.0	72.5
YOLOv11-RDA	2.7	6.3	**89.4**	**74.0**

As shown in Table I, integrating CSP-MSPA into the backbone improves performance to 89.9% mAP_{50} and 73.9% mAP_{50-95}, compared to 88.0% and 72.5% for the baseline. C3K2 serves as the default backbone block of YOLOv11-N, while CSP-MSPA extends it with multi-scale convolutional branches and cross-layer interactions, enhancing feature diversity and information flow. The increase in GFLOPs (from 6.4 G to 7.6 G) is not a result of redundant channel calculations but reflects a deliberate allocation of computational resources toward spatial feature modeling. In highway scenarios involving high-resolution inputs and dense, distant targets, the multi-scale kernels allow the model to capture features across varying receptive fields. This investment in spatial reasoning provides the granular long-range context necessary for accurate localization of fine-grained targets, proving more effective than simply increasing channel depth. Consequently, while maintaining a comparable parameter count (2.6 M), CSP-MSPA achieves a superior trade-off between detection accuracy and computational investment.

Table II compares AFPN and BiFPN. AFPN achieves 89.3% mAP_{50} and 73.8% mAP_{50-95}, representing gains of 1.3% and 1.8% over the baseline, while reducing the parameter count to 2.1 M. AFPN builds upon the traditional top-down FPN framework by introducing learnable normalization weights, allowing adaptive reweighting of multi-level features during fusion. AFPN achieves higher accuracy (73.8% relative to 73.3% for BiFPN) in complex highway scenarios. At the same time, the computational cost remains low (6.6 GFLOPs), highlighting its effectiveness in balancing accuracy with lightweight design.

According to Table III, RDA achieves the best performance with 89.4% mAP_{50} and 74.0% mAP_{50-95}. By combining recursive dilated convolutions with weight sharing, it enlarges the receptive field while reducing redundancy, improving contextual modeling for distant, blurred, and complex targets. With only a slight increase in complexity (2.7M, 6.3 GFLOPs), RDA proves both efficient and practical. Overall, CSP-MSPA boosts fine-grained backbone features, AFPN improves lightweight multi-scale fusion, and RDA enhances contextual robustness, together validating the effectiveness of the proposed design.

TABLE IV: Ablation study of different modules on the HVD dataset.

CSP-MSPA	RDA	AFPN	Params (M)	FLOPs (G)	mAP_{50}	mAP_{50-95}
–	–	–	2.6	6.4	88.0	72.5
✓	–	–	2.6	7.6	89.9	73.9
–	✓	–	2.7	6.3	89.4	74.0
–	–	✓	2.1	6.6	89.3	73.8
✓	✓	–	2.8	7.6	89.4	74.3
✓	–	✓	2.2	7.8	90.1	74.5
–	✓	✓	2.3	6.5	89.6	74.2
✓	✓	✓	2.3	7.8	**90.4**	**75.0**

TABLE V: Performance comparison of different detectors on the UA-DETRAC and the HVD datasets.

Method	Params (M)	FLOPs (G)	UA-DETRAC		HVD	
			AP_{50}	mAP_{50-95}	mAP_{50}	mAP_{50-95}
YOLOv8-N	3.0	8.1	77.4	62.9	89.0	73.6
YOLOv9-T [6]	2.0	7.6	80.3	64.0	89.1	73.6
YOLOv10-N [7]	2.3	6.5	80.3	64.2	88.3	72.8
YOLOv11-N	2.6	6.4	80.0	64.9	88.0	72.5
YOLOv12-N [8]	2.6	6.3	80.3	63.8	88.5	73.1
YOLOv13-N [9]	2.5	6.2	76.5	62.5	82.0	71.8
GOLD-YOLO-N [10]	5.6	12.1	80.4	63.1	90.1	72.3
RT-DETR-L [11]	32.0	103.5	75.3	57.5	82.6	65.9
DEIM-N [12]	3.8	7.2	83.5	67.0	91.0	74.8
DFRNet-H (ours)	2.3	7.8	**86.0**	**69.3**	90.4	**75.0**

2) Ablation Studies: To analyze the contribution of each proposed module, we conducted ablation studies on the YOLOv11-N baseline using the HVD dataset, as summarized in Table IV. Incorporating CSP-MSPA and RDA improved mAP_{50}/mAP_{50-95} from 88.0%/72.5% to 89.4%/74.3%, where CSP-MSPA enhances small-object representation via multi-scale fractal convolutions and RDA expands receptive fields through recursive dilated convolutions. Combining CSP-MSPA with AFPN further boosted accuracy to 90.1%/74.5%, indicating the critical role of adaptive cross-scale fusion in addressing scale variations. Similarly, RDA with AFPN achieved 89.6%/74.2%, highlighting the complementarity between contextual modeling and multi-scale fusion. The full model integrating all three modules delivered the best results, 90.4%/75.0%, with only 2.3M parameters and 7.8 GFLOPs, validating their synergy: CSP-MSPA strengthens small-object features, RDA provides robust context, and AFPN enables adaptive fusion. Overall, each module contributes consistent gains, and their joint integration significantly improves small-object detection and robustness while maintaining efficiency.

3) Model Comparison: In a comprehensive evaluation on UA-DETRAC and our HVD dataset (Table V), DFRNet-H achieves mAP_{50-95} of 69.3% and 75.0%, respectively, setting new records among lightweight models. It shows significant improvements over YOLO variants and even surpasses larger Transformer-based detectors, demonstrating the proposed modules' gains in complex highway scenarios and their favorable trade-off between efficiency and accuracy.

Beyond quantitative results, Fig. 6 provides qualitative comparisons. DFRNet-H produces more precise bounding boxes and achieves higher confidence scores on small or distant vehicles compared to YOLOv11, highlighting its advantage in challenging highway scenarios. Furthermore, the heatmap visualizations in Fig. 7 show that DFRNet-H focuses more effectively on vehicle regions while suppressing irrelevant background responses. This indicates that the proposed modules not only improve accuracy but also enhance feature interpretability and robustness.

Fig. 6: Qualitative comparison of detection results: (a) ground-truth annotations, (b) YOLOv11 predictions, and (c) DFRNet-H predictions. Compared to YOLOv11, DFRNet-H achieves more accurate localization, fewer missed detections, and better performance on small and distant vehicles.

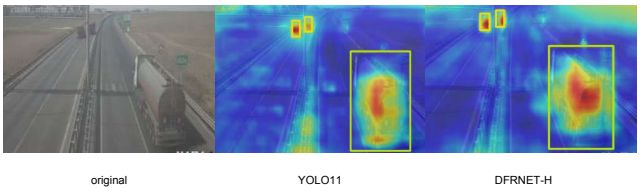


Fig. 7: Heatmap visualization of feature responses: (a) original input, (b) YOLOv11 activation map, and (c) DFRNet-H activation map. DFRNet-H exhibits stronger and more focused responses on vehicles while suppressing background noise, demonstrating its enhanced feature discrimination.

V. CONCLUSIONS

This paper presents DFRNet-H, a lightweight network for highway vehicle detection. By integrating CSP-MSPA, RDA, and AFPN, it enhances feature extraction, context modeling, and multi-scale fusion. Experiments on the HVD dataset and UA-DETRAC demonstrate superior accuracy and robustness over mainstream detectors while maintaining real-time performance. Future work will focus on optimizing DFRNet-H for edge deployment and improving robustness under diverse traffic conditions. Owing to its efficiency and accuracy, DFRNet-H is particularly suitable for deployment in intelligent transportation systems, roadside monitoring units, and in-vehicle perception modules.

REFERENCES

[1] Yongke Wei, Zimu Zeng, Tingquan He, Shanchuan Yu, Yuchuan Du, and Cong Zhao, "An adaptive vehicle detection model for traffic surveillance of highway tunnels considering luminance intensity," Sensors, vol. 24, no. 18, pp. 5912, 2024.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, "End-to-end object detection with transformers," in European conference on computer vision. Springer, 2020, pp. 213–229.
[3] Yan-Feng Lu, Jing-Wen Gao, Qian Yu, Yi Li, Yi-Sheng Lv, and Hong Qiao, "A cross-scale and illumination invariance-based model for robust object detection in traffic surveillance scenarios," IEEE transactions on intelligent transportation systems, vol. 24, no. 7, pp. 6989–6999, 2023.
[4] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," Computer Vision and Image Understanding, vol. 193, pp. 102907, 2020.
[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in European conference on computer vision. Springer, 2014, pp. 740–755.
[6] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao, "Yolov9: Learning what you want to learn using programmable gradient information," in European conference on computer vision. Springer, 2024, pp. 1–21.
[7] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, et al., "Yolov10: Real-time end-to-end object detection," Advances in Neural Information Processing Systems, vol. 37, pp. 107984–108011, 2024.
[8] Yunjie Tian, Qixiang Ye, and David Doermann, "Yolov12: Attention-centric real-time object detectors," arXiv preprint arXiv:2502.12524, 2025.
[9] Mengqi Lei, Siqi Li, Yihong Wu, Han Hu, You Zhou, Xinhu Zheng, Guiguang Ding, Shaoyi Du, Zongze Wu, and Yue Gao, "Yolov13: Real-time object detection with hypergraph-enhanced adaptive visual perception," arXiv preprint arXiv:2506.17733, 2025.
[10] Chengcheng Wang, Wei He, Ying Nie, Jianyuan Guo, Chuanjian Liu, Yunhe Wang, and Kai Han, "Gold-yolo: Efficient object detector via gather-and-distribute mechanism," Advances in Neural Information Processing Systems, vol. 36, pp. 51094–51112, 2023.
[11] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen, "Detrs beat yolos on real-time object detection," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2024, pp. 16965–16974.
[12] Shihua Huang, Zhichao Lu, Xiaodong Cun, Yongjun Yu, Xiao Zhou, and Xi Shen, "Deim: Detr with improved matching for fast convergence," in Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 15162–15171.
[13] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.
[14] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," in proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 9197–9206.
[15] Mingxing Tan, Ruoming Pang, and Quoc V Le, "Efficientdet: Scalable and efficient object detection," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10781–10790.
[16] Yijing Guo, Yixin Zeng, Fengqiang Gao, Yi Qiu, Xuqiang Zhou, Linwei Zhong, and Choujun Zhan, "Improved yolov4-csp algorithm for detection of bamboo surface sliver defects with extreme aspect ratio," Ieee Access, vol. 10, pp. 29810–29820, 2022.
[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 9, pp. 1904–1916, 2015.
[18] Fisher Yu and Vladlen Koltun, "Multi-scale context aggregation by dilated convolutions," arXiv preprint arXiv:1511.07122, 2015.
[19] Seung-Hwan Bae, "Object detection based on region decomposition and assembly," in Proceedings of the AAAI conference on artificial intelligence, 2019, pp. 8094–8101.
[20] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam, "Rethinking atrous convolution for semantic image segmentation," arXiv preprint arXiv:1706.05587, 2017.
[21] Md Mostafijur Rahman, Mustafa Munir, and Radu Marculescu, "Emcad: Efficient multi-scale convolutional attention decoding for medical image segmentation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 11769–11779.