Journal of Real-Time Image Processing (2026) 23:25
https://doi.org/10.1007/s11554-025-01812-w

RESEARCH

HAS-DETR: a real-time lightweight traffic sign detection model

Junhao Dong[1] · **Hongxiang Liao**[1] · **Xiaohui Ji**[1]

Received: 14 September 2025 / Accepted: 16 November 2025

Abstract

Real-time traffic sign detection is critical for autonomous driving and intelligent transportation system safety. A key challenge in this field is balancing real-time performance enhancement, model lightweighting, and detection accuracy. To address this, we propose HAS-DETR, a real-time lightweight traffic sign detection model based on RT-DETR, with novel improvements to the backbone network, feature fusion mechanism, and small-target detection capability. Specifically, we design a High-Frequency Enhanced CSP Backbone (HFCSP-Backbone) to resolve the issues of insufficient high-frequency detail extraction and excessive parameters in traditional backbones; we introduce an Attention Scale Sequence Fusion (ASSF) module to dynamically model contextual correlations across multi-scale features for better feature representation; and we add a Small-Target Enhancement (STE) detection head embedded in the Transformer decoder (via a S2 small-scale feature layer) to mitigate small-target missed detection. Experiments on the TT100K dataset show that HAS-DETR reduces parameters by 58.99% and computational load by 20.49%, while improving detection performance: mAP@0.5 increases from 84.8 (RT-DETR-R18 baseline) to 86.6%, and mAP@0.5:0.95 from 63.1 to 66.4%, with precision and recall reaching 90% and 81.7%, respectively. Compared with existing methods, HAS-DETR achieves a superior balance between lightweighting and accuracy, offering an efficient solution for real-time traffic sign detection in complex scenarios.

Keywords Small-target detection · Traffic sign detection · RT-DETR · Lightweight

1 Introduction

Traffic sign detection serves as a core module in intelligent transportation systems and autonomous driving technologies, with a critical role in safeguarding road safety and enhancing traffic operational efficiency [1–4]. On-board detection systems in intelligent transportation and autonomous driving collect and analyze road images in real time via on-board cameras, and must rapidly identify traffic sign information [5]. However, existing models generally exhibit high computational complexity, large parameter counts, and low inference speeds. A key challenge in this field lies in improving real-time performance and reducing both model complexity and parameter counts while maintaining detection accuracy. Although lightweight research has focused on network architecture optimization, model compression, and hardware acceleration, three scenario-specific challenges remain for traffic sign detection: enhancing the capture of fine-grained details (e.g., textures of small traffic signs) [6], improving real-time processing speed to meet the millisecond-level response requirement of on-board systems, and reducing computational consumption to adapt to resource-constrained embedded devices in vehicles.

With the advancement of deep learning, end-to-end models, typified by DETR (Detection Transformer) [7], have garnered attention owing to their advantages of anchor-free design and global modeling capabilities. While foundational models like Deformable DETR [8] focused on improving attention efficiency and convergence speed, and Efficient DETR [9] explored lightweighting through pruning and distillation, applying them to specific real-time traffic sign tasks remains challenging. RT-DETR (Real-Time DETR) [10], the latest improved version, has demonstrated outstanding performance on general object detection datasets (e.g., COCO [11]), with accuracy surpassing that of mainstream

✉ Xiaohui Ji
xhji@cugb.edu.cn

Junhao Dong
jhdong@email.cugb.edu.cn

Hongxiang Liao
hxliao@email.cugb.edu.cn

[1] School of Artificial Intelligence, China University of Geosciences, Beijing 100083, China

Published online: 10 December 2025

models including YOLOv5, YOLOv7, and YOLOv8 [12], while avoiding the target deletion issue caused by Non-Maximum Suppression (NMS) in traditional models. However, when applied to traffic sign detection, RT-DETR still exhibits limitations in small-target detection scenarios: detailed features of traffic signs (typically smaller than 32×32 pixels) are easily overlooked, leading to degraded detection performance; meanwhile, the high computational complexity of the Transformer [13] architecture limits real-time performance, making it difficult to meet the requirements of embedded devices.

To address the above issues, we propose a real-time lightweight traffic sign detection model, HAS-DETR (High-Frequency Enhanced CSP Backbone, Attention Scale Sequence Fusion, and Small-Target Enhancement-based DETR), based on the RT-DETR framework. The name "HAS-DETR" is derived from the initials of its three core improved modules–High-Frequency Enhanced CSP Backbone (HFCSP-Backbone), Attention Scale Sequence Fusion (ASSF), and Small-Target Enhancement (STE)–combined with the "DETR" component of the baseline model RT-DETR. By optimizing the backbone's feature extraction, enhancing multi-scale feature fusion efficiency, and supplementing small-target detection branches, HAS-DETR improves small-target accuracy and inference speed while reducing computational load and parameters, offering a more effective solution for resource-constrained intelligent transportation scenarios.

Our novel contributions, designed to address these gaps, are threefold:

(1) To resolve traditional backbones' limitations (insufficient high-frequency detail extraction and high computational redundancy) and optimize feature extraction, we propose a High-Frequency Enhanced CSP Backbone (HFCSP-Backbone), with its core being the High-Frequency Enhanced CSP (HFCSP) module. This module integrates a high-frequency enhanced residual structure and CSP strategy: the former strengthens extraction of texture/edge-related high-frequency details to make up for traditional backbones' detail loss, while the latter reduces computational redundancy without compromising feature expression, balancing multi-scale feature extraction efficiency and lightweighting.
(2) For multi-scale feature fusion, we design the Attention Scale Sequence Fusion (ASSF) module. It dynamically models contextual correlations across different-scale features, adaptively adjusts feature weights by focusing on key scales' semantic guidance, and effectively integrates global semantics and local details to provide more targeted feature input for subsequent detection heads.
(3) To enhance small-target detection, a Small-Target Enhancement (STE) head is embedded in the Transformer decoder. By introducing the S2 small-scale feature layer to supplement small targets' high-resolution details, and combining with the decoder's global contextual modeling capability, it significantly improves the model's perception and localization accuracy for small traffic signs.

2 Related work

Object detection, a key task in computer vision, aims to identify objects in images and determine their positions. In the field of traffic sign detection, traditional methods primarily rely on manually extracted features (e.g., color and shape). These methods are not only computationally complex and inefficient for detection tasks but also lack robustness and adaptability in variable environments [14]. In recent years, with the advancement of deep learning, detection models based on this technology have significantly outperformed traditional methods in both accuracy and efficiency due to their capacity to autonomously learn and extract features, thus becoming the mainstream approach in this field [15]. Deep learning-based object detection architectures are mainly categorized into two types: detectors based on Convolutional Neural Networks (CNNs) and emerging architectures based on Transformers.

The evolution of CNN-based detectors reflects the ongoing pursuit of balancing detection speed and accuracy. Early two-stage detectors, such as the OverFeat model proposed by Sermanet et al. [16], were the first to integrate recognition, localization, and detection into a single framework. However, their fixed sliding windows resulted in insufficient bounding box localization accuracy. Subsequently, algorithms in the R-CNN [17] series gradually improved localization accuracy by introducing Region Proposal Networks (RPN) and Spatial Pyramid Pooling (SPPNet). Fast R-CNN [18] optimized computational efficiency by extracting features from the entire image; Faster R-CNN [19] further integrated the detection process using RPN to generate detection boxes directly. The primary trade-off of these two-stage methods is their high localization accuracy at the cost of high computational complexity and low inference speed. To tackle this issue, single-stage detectors emerged. The YOLO [12] model proposed by Redmon et al. treats detection as a regression problem, achieving extremely fast detection speed but sacrificing some accuracy, particularly on small objects. As an improvement, the SSD [20] model proposed by Liu et al. combines the regression concept of YOLO with the anchor box mechanism of Faster R-CNN, achieving a better balance between speed and accuracy.

Despite the considerable success of CNN-based detectors, most rely on manually designed post-processing components (e.g., Non-Maximum Suppression, NMS) for filtering redundant detection boxes. In complex scenarios with dense targets or severe occlusions, the NMS mechanism may erroneously suppress detection boxes containing targets, thereby affecting detection stability and accuracy—a problem that is particularly prominent in complex road environments. To fully address this issue, Transformer-based detectors (DETR) [7] were proposed. As a pioneering end-to-end, single-stage framework, these detectors eliminate manual components such as NMS and construct a true end-to-end detection framework, greatly simplifying the algorithm workflow. This simplified pipeline is their main advantage, but the trade-off in the original DETR was slow convergence and high computational cost. Following DETR, researchers have proposed a series of improved models. For instance, Deformable DETR [8] improves convergence speed and efficiency by introducing deformable attention modules; Anchor DETR [21] integrates anchor box prior knowledge into the framework to achieve better localization performance.

To apply the advantages of the Transformer architecture to real-time detection tasks, researchers have focused on model lightweighting and efficiency improvement. Efficient DETR [9] significantly reduces computational and memory overhead through techniques such as knowledge distillation and pruning. The RT-DETR [10] model proposed by Zhao et al. is specifically designed for real-time object detection, introducing lightweight Transformer modules and significantly improving detection accuracy while avoiding problems caused by NMS. However, despite its overall excellent performance, prior analyses suggest that RT-DETR's performance on small targets is suboptimal, as its feature fusion structure may not provide sufficient high-resolution detail for accurately detecting small-scale objects. Furthermore, the large number of parameters in RT-DETR makes it difficult to deploy on edge computing devices, failing to fully meet the needs of real-time road traffic sign detection. In summary, the current research focus, spanning from specific traffic sign tasks to broader visual domains like specialized recognition using attention-based transformers [22], is on how to further reduce model parameters and computational load—often through advanced attention mechanisms—while ensuring high accuracy, and to design a lightweight model more suitable for real-time and efficient detection in real road scenarios.

To address the aforementioned challenges, the HAS-DETR model proposed in this paper aims to achieve a superior balance. Unlike CNN-based methods, which rely on NMS and often struggle to reconcile speed and accuracy, HAS-DETR inherits the end-to-end, real-time advantages of the RT-DETR framework. Furthermore, compared to the RT-DETR baseline, HAS-DETR's expected improvements are concentrated in three main areas: It significantly reduces parameters and computational redundancy via our designed HFCSP-Backbone to meet edge deployment requirements; it employs the ASSF module to optimize multi-scale feature fusion efficiency; it introduces an STE detection head to specifically compensate for the baseline model's insufficient perceptual capabilities for small targets (such as traffic signs), thereby achieving higher detection accuracy while remaining lightweight.

3 Proposed method

This study proposes a lightweight model named HAS-DETR to address the challenge of detecting small traffic signs. The overall network architecture of HAS-DETR is shown in Fig. 1, the primary block diagram illustrating the model's components and their interactions.

The model's step-by-step workflow proceeds as follows:

(1) Preprocessing: Input images are first preprocessed, primarily by resizing them to a uniform 640x640 resolution (as detailed in Sect. 4.2), and are fed into the network.
(2) Backbone feature extraction: The image is processed by our proposed HFCSP-Backbone (detailed in Sect. 3.1). This module replaces the baseline's ResNet-18 to efficiently extract multi-scale features (S2, S3, S4, S5), with a specific focus on enhancing high-frequency details while reducing computational redundancy and parameters.
(3) Encoder feature fusion: The multi-scale features from the backbone are passed to the hybrid encoder. This stage incorporates our novel ASSF module (detailed in Sect. 3.2) to dynamically model contextual correlations and effectively fuse the features from different scales.
(4) Decoder and prediction: Finally, the fused features are input to the Transformer decoder. Our STE detection head (detailed in Sect. 3.3) is embedded here, utilizing the high-resolution S2 feature layer to significantly improve the perception and localization of small targets, leading to the final detection results.

This design, from backbone optimization to multi-scale fusion and targeted small-target detection, achieves a superior balance between detection accuracy and efficiency for real-world traffic sign detection. The following subsections detail the design of each innovative component.

3.1 HFCSP-Backbone

Optimization in this study is performed based on the RT-DETR-R18 model. Although the RT-DETR framework

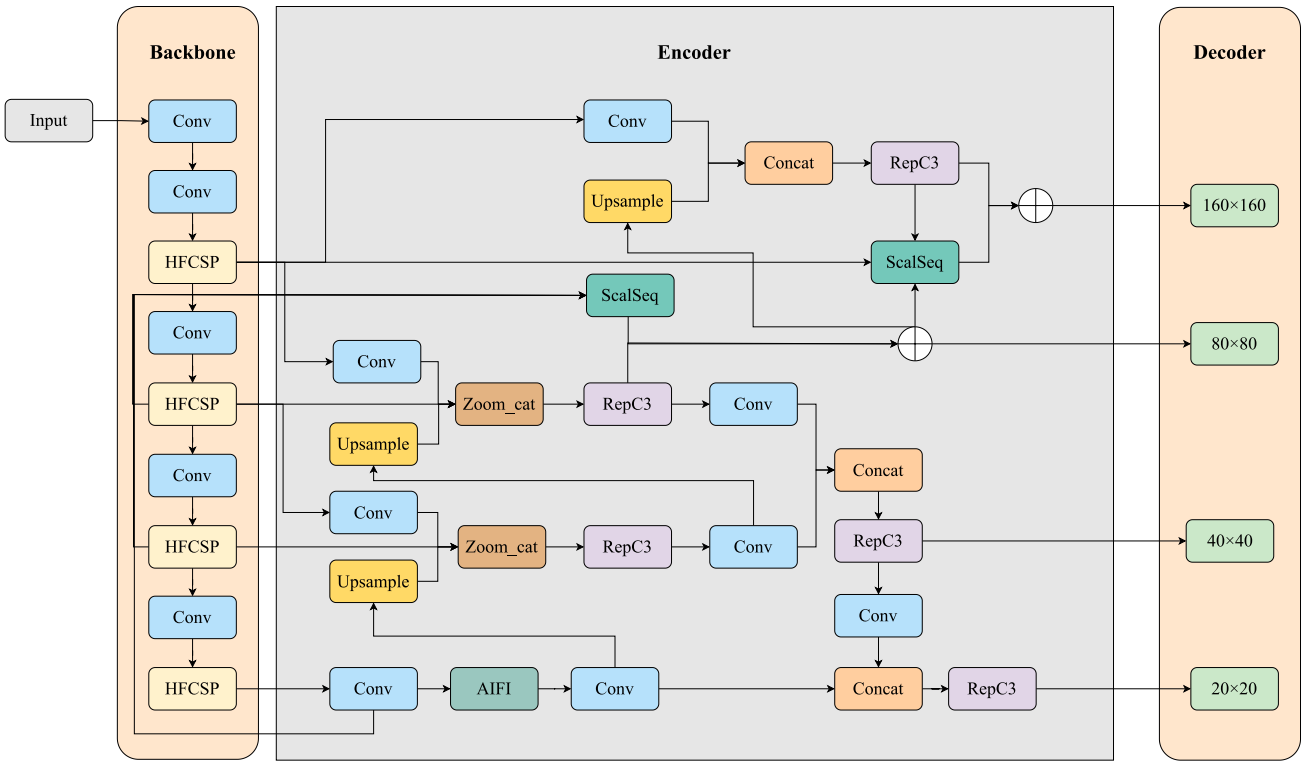

Fig. 1 Network architecture of HAS-DETR

supports more complex backbone networks (e.g., HGNet [23]), ResNet-18 [24]—which balances accuracy and parameter count—was selected as the baseline to meet the requirements of real-time traffic sign detection tasks. However, the baseline still encounters two bottlenecks: first, significant computational redundancy induced by network deepening, which impairs real-time performance; second, insufficient representational capability for high-frequency information (e.g., object edges and textures) during deep feature extraction.

To overcome the above bottlenecks, and inspired by the Cross-Stage Partial (CSP) concept in the YOLO series [25] and the high-frequency enhancement strategy proposed by Li et al. [26], a new efficient backbone network—High-Frequency Enhanced CSP Backbone (HFCSP-Backbone)—was designed. The overall architecture of this network, as illustrated on the left side of Fig. 1, is built upon a new CSP-like design rather than simply modifying the ResNet-18 layers. Specifically, the HFCSP module is used as the core component at each main feature extraction stage. The structure consists of downsampling convolutional layers that define the feature hierarchy (S2, S3, S4, and S5), where each downsampling layer is immediately followed by one or more HFCSP modules to perform in-depth feature extraction at that specific scale.

The core of HFCSP-Backbone is the High-Frequency Enhanced CSP Module (HFCSP), whose detailed structure is shown in Fig. 2. The design of this module subtly integrates high-frequency detail enhancement and computational efficiency optimization. In specific implementation, the input feature map first undergoes a convolution operation for information integration, after which it is split into two parallel branches. Branch 1 is directly transmitted to the end of the module as a cross-stage connection to preserve the original gradient information; Branch 2 enters a deep processing path consisting of N High-Frequency Enhancement Residual Blocks (HFERB), which is responsible for fine-grained feature extraction. Finally, the feature maps from the two branches are fused via a concatenation (Concat) operation and output through a final convolutional layer for feature integration.

In the deep processing path of the HFCSP module, the key component is the High-Frequency Enhancement Residual Block (HFERB), whose structure is shown in Fig. 3. Inspired by the CRAFT network [26], the core objective of HFERB is to explicitly enhance high-frequency information in feature maps. To achieve this, it contains two parallel branches: a Local Feature Extraction (LFE) branch and a High-Frequency Enhancement (HFE) branch

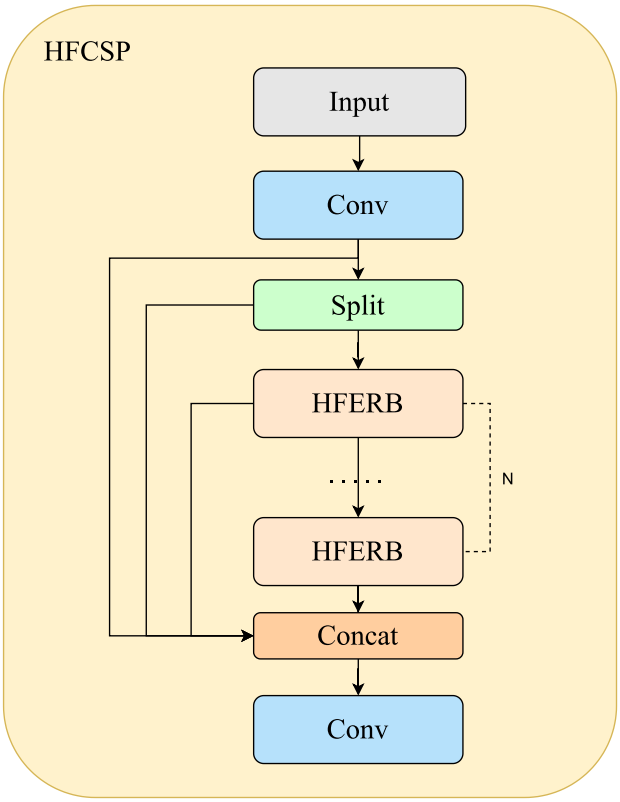

Fig. 2 Structural schematic of the HFCSP module

$$F_{in}^{LFE}, F_{in}^{HFE} = Split(Input), \tag{1}$$

where $F_{in}^{LFE}, F_{in}^{HFE} \in \mathbb{R}^{H\times W\times C/2}$ represent the inputs of the LFE and HFE branches, respectively. For the LFE branch, a 3×3 convolutional layer followed by a GELU activation function is utilized to extract local high-frequency features, where $g_u(\cdot)$ denotes the GELU activation function

$$F_{out}^{LFE} = g_u(Conv_{3\times3}(F_{in}^{LFE})). \tag{2}$$

For the HFE branch, a max-pooling layer is innovatively adopted to explicitly capture high-frequency information in the input feature F_{in}^{HFE}. Then, a 1×1 convolutional layer followed by a GELU activation function is used to enhance high-frequency features

$$F_{out}^{HFE} = g_u(Conv_{1\times1}(MaxPooling(F_{in}^{HFE}))). \tag{3}$$

Subsequently, the outputs of the two branches are concatenated and fused, and information integration is performed through a 1×1 convolutional layer. Finally, a residual connection (Shortcut) is introduced to add the original input to the fused features, maintaining multi-scale information and ensuring the stability of the training process. The entire process can be expressed as

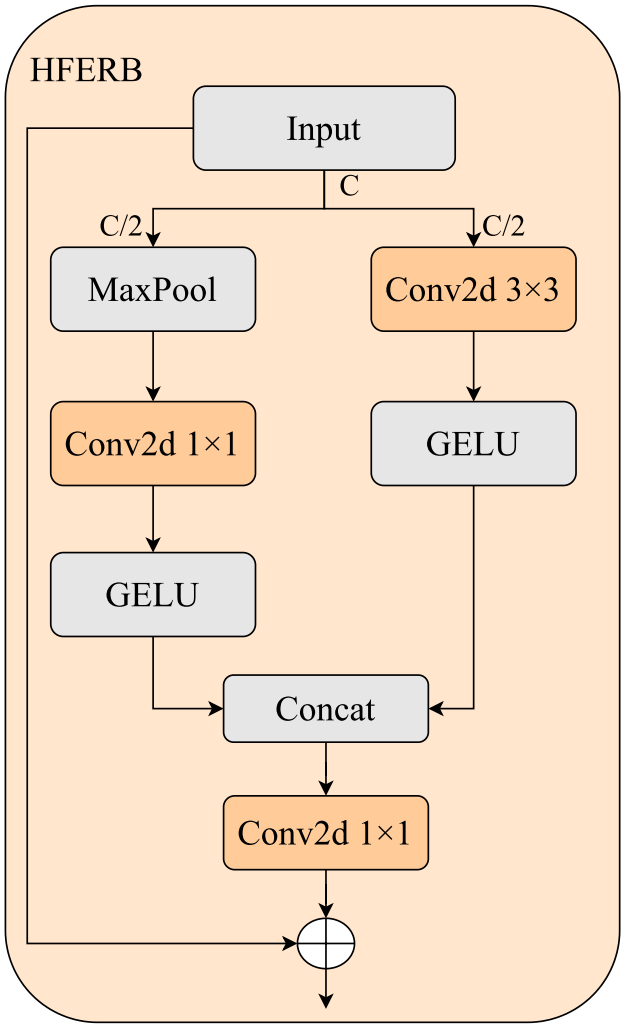

Fig. 3 Structural schematic of the HFERB module

$$Output = \mathrm{Conv}_{1\times1}\left(\mathrm{Concat}\left(F_{out}^{LFE}, F_{out}^{HFE}\right)\right) + Input. \tag{4}$$

Through this series of elaborate designs, the new HFCSP-Backbone achieves dual advantages: on the one hand, it can accurately capture and enhance texture and edge details crucial for traffic sign detection using the HFERB module; on the other hand, it can significantly reduce computational redundancy and memory usage by virtue of the CSP architecture. Ultimately, the HAS-DETR model achieves better inference efficiency while significantly improving detection accuracy.

3.2 ASSF

A core challenge in traffic sign detection tasks lies in the effective recognition of small targets. Owing to sparse pixel

distribution, key information of small targets is prone to loss during network downsampling. To address this problem, an Attention Scale Sequence Fusion (ASSF) module was designed and integrated into the hybrid encoder of HAS-DETR, aiming to efficiently fuse multi-scale features extracted from the HFCSP-Backbone. The innovation of the ASSF module resides in its dual-path collaborative design, which comprises a Scale Sequence Feature Fusion (SSFF) module and two Triple Feature Encoder (TFE) modules [27]—the former responsible for capturing global semantic information and the latter for extracting local detailed features.

As shown in Fig. 4, the SSFF module focuses on fusing global high-level semantic information. It normalizes feature maps from different levels to a uniform size via upsampling and other operations, before efficiently merging them using 3D convolution to integrate multi-scale semantic information. This process effectively integrates multi-scale features, enabling the model to better understand traffic signs of different sizes and shapes.

As shown in Fig. 5, the TFE module is mainly responsible for capturing local fine features of small targets. Each TFE module receives three feature maps of different sizes (denoted as C×S, where C is the number of channels and S is the feature map size) as input, and accurately retains and enhances detail information crucial for detecting small targets by integrating these features in the spatial dimension.

Inside the ASSF module, to effectively integrate local fine information captured by TFE with global features, we fuse the output of TFE with the backbone features through an ADD operation. Finally, the feature stream co-enhanced by SSFF and TFE undergoes final path aggregation through the PANet [28] structure, thereby transmitting rich contextual information to the detection head. This complete fusion

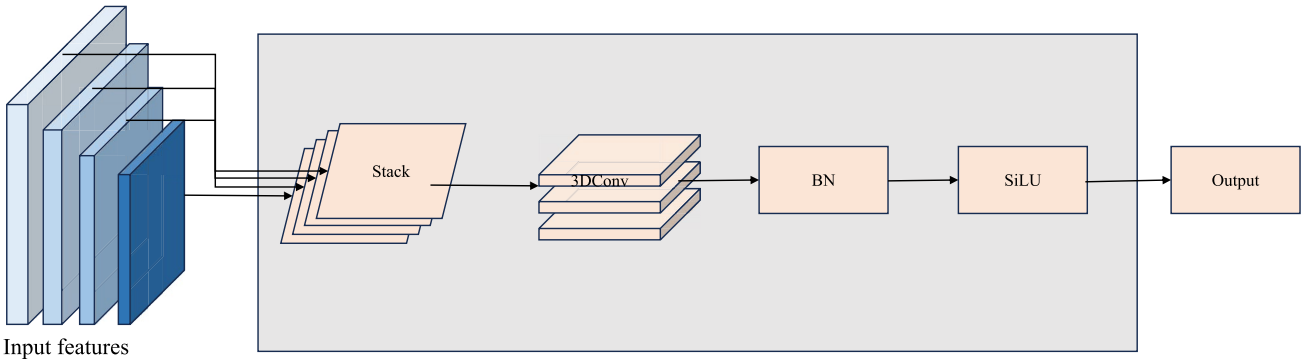

Fig. 4 Structural schematic of the SSFF module

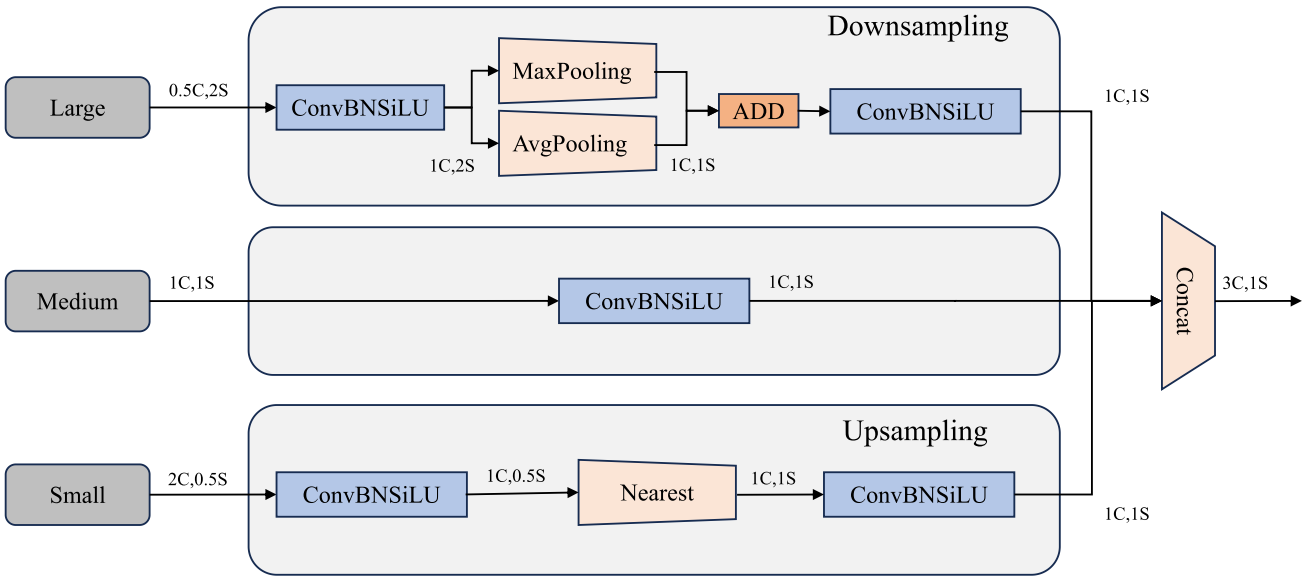

Fig. 5 Structural schematic of the TFE module

mechanism enhances small-target detection performance while ensuring detection capability for targets of regular sizes.

3.3 STE

Traditional multi-scale detectors (such as RT-DETR) usually adopt three levels of feature maps S3, S4, and S5 (e.g., 80×80, 40×40, 20×20) for detecting small, medium, and large targets, respectively. We observed that in traffic sign datasets, most targets are of medium and small sizes, resulting in low efficiency of the low-resolution feature layer (20×20) designed for large targets, which may even introduce interference due to its macro-receptive field, damaging detection accuracy.

To solve this problem, we redesigned the detection levels of the model to focus more on high-resolution features. We introduced a 160×160 feature layer, which is formed by upsampling the 80×80 feature map in the feature fusion network and concatenating it with the high-resolution features of the corresponding level in the backbone network. This improved structure (as shown in Fig. 6) can provide richer fine-grained information for the model, thereby significantly improving the positioning and recognition performance for small targets.

4 Experimental results

4.1 Datasets

To comprehensively evaluate the comprehensive performance of the proposed method under category diversity and complex backgrounds, we selected two challenging public datasets with distinct characteristics: TT100K [2] and CCTSDB 2021 [29].

TT100K is a large-scale, high-resolution dataset jointly released by Tsinghua University and Tencent. The dataset contains 100,000 panoramic images provided by Tencent Data Center, among which approximately 10,000 images contain traffic signs. All images were collected from real road scenes in multiple cities in China, covering variable lighting and weather conditions, providing a solid foundation for simulating real-world detection tasks. However, the original TT100K data have a serious category imbalance problem, with many categories in its 221 sign categories having very few samples. To avoid overfitting of the model to categories with sparse samples and ensure the effectiveness of training, we followed common practices and only selected 45 categories with more than 100 instances in the dataset for this experiment. The total number of images after screening is 10,592. Examples of images from the TT100K dataset are shown in Fig. 7.

CCTSDB 2021 is another real-scene dataset used for evaluation, containing 17,856 images with detailed annotations that show complex background environments. The dataset clearly divides traffic signs into three core categories crucial for intelligent driving: mandatory, prohibitory, and warning. These two datasets, one focusing on the challenges of category diversity and imbalance in large-scale scenarios, and the other on the detection of core safety signs in complex backgrounds, jointly provide strong support for verifying the robustness and generalization ability of the model. Examples of images from the CCTSDB 2021 dataset are shown in Fig. 8.

In summary, these two datasets, one focusing on category diversity and imbalance challenges in large-scale scenarios, and the other on the detection of core safety signs in

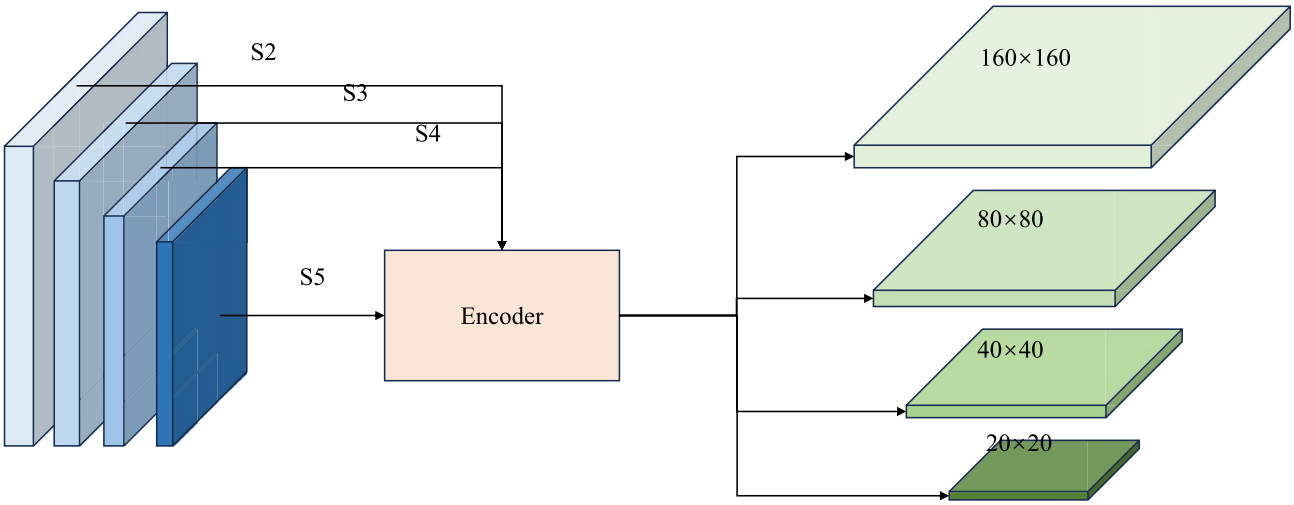

Fig. 6 Output structure of the encoder

423.jpg 425.jpg 428.jpg

Fig. 7 Examples from the TT100K dataset

00010.jpg 00011.jpg 00013.jpg

Fig. 8 Examples from the CCTSDB dataset

complex backgrounds, jointly provide strong support for comprehensively verifying the robustness and generalization ability of our model.

4.2 Parameter settings

In this experiment, the hardware platform configuration used includes an Intel(R) Xeon(R) CPU E5-2620, 64GB memory, and a Tesla P100-PCIE-16GB graphics card, with the operating system being CentOS 7, Python version 3.8.16, and the deep learning framework PyTorch 1.13.1.

To optimize the training effect of the model, we adjusted the corresponding hyperparameters according to the model's performance during actual training. During training, we set the initial learning rate to 0.0001, the batch size to 8, the training epoch to 160, and selected Adam as the optimization algorithm, with the input image size being 640×640. It is important to note that all experiments were conducted on this server-grade GPU platform (Tesla P100) to ensure consistent and reproducible benchmarking. We did not perform validation on embedded or edge hardware platforms (e.g., NVIDIA Jetson series or ARM-based devices).

4.3 Evaluation metrics and training process

To comprehensively evaluate the model's performance and efficiency, we employed a set of standard metrics. Detection accuracy was measured using Precision (P), Recall (R), Average Precision (AP), and mean Average Precision (mAP). The calculation for P, R, and AP is shown in Eqs. 5, 6, and 7, respectively. To more accurately measure the model's localization quality, we report two core mAP metrics: mAP@0.5 (calculated with an IoU threshold of 0.5) and mAP@0.5:0.95 (the average mAP across 10 IoU thresholds from 0.5 to 0.95).

Table 1 Comparative experiments on the TT100K dataset

Model	Backbone	P (%)	R (%)	mAP@0.5 (%)	FLOPs (G)	Params (M)	Latency (ms)
SSD (2016)	VGG-16	70.2	67.1	67.5	–	–	–
Faster R-CNN (2016)	ResNet-50	67.1	70.5	70.3	–	–	–
YOLOv8m (2023)	CSPDarknet-53	84.5	72.5	82	78.8	25.86	**12.6**
YOLOv9m (2024)	CSPDarknet-53	85.7	76.1	84.3	77.7	20.19	13.7
YOLOv10m (2024)	CSPDarknet-53	83.6	68.5	79.5	64.3	16.53	13.3
YOLOv11m (2024)	CSPDarknet-53	87.2	72.4	83.3	68.4	20.08	13.9
YOLOv12m (2025)	CSPDarknet-53	83.6	74.4	82.9	67.3	20.13	12.8
RT-DETR-R18 (2024)	ResNet-18	86.5	**81.7**	84.8	57.1	19.92	12.7
HAS-DETR (ours)	HFCSP	**90**	81.1	**86.6**	**45.4**	**8.17**	13.1

The data marked with bold represents the optimal performance metrics of the proposed model, while the data with underline denotes the suboptimal performance metrics

These metrics collectively evaluate the model's fundamental detection capability and localization robustness. The formula for mAP is given in Eq. 8, where N is the total number of target classes. In the tables of comparative experiments and ablation experiments, values in bold indicate the optimal performance metrics among all evaluated models, while underlined values represent the suboptimal ones.

For model complexity, we measured the number of Parameters (M) and the computational cost in Floating-Point Operations (FLOPs) (G). Latency (ms) was used to evaluate the model's real-time processing capability

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

$$AP = \int_0^1 p(r)dr \tag{7}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i. \tag{8}$$

4.4 Comparative experimental results

To demonstrate the advantages of the proposed method in traffic sign detection, we compared HAS-DETR with several classic object detection algorithms on the TT100K and CCTSDB datasets, including SSD, Faster R-CNN, YOLOv8m, YOLOv9m [30], YOLOv10m [31], YOLOv11m, YOLOv12m, and the baseline model (RT-DETR-R18). The experimental results are shown in Tables 1 and 2.

Tables 1 and 2 present the comparative experimental results of the proposed HAS-DETR model with other mainstream models on the TT100K and CCTSDB datasets, respectively. Overall, HAS-DETR demonstrates superior performance in detection accuracy, model lightweighting, and inference efficiency, achieving an optimal balance between accuracy and efficiency.

Table 2 Comparative experiments on the CCTSDB dataset

Model	Backbone	P (%)	R (%)	mAP@0.5 (%)	FLOPs (G)	Params (M)	Latency (ms)
SSD (2016)	VGG-16	86.5	27.7	49.2	–	–	–
Faster R-CNN (2016)	ResNet-50	84.4	54.9	56.6	–	–	–
YOLOv8m (2023)	CSPDarknet-53	88.5	75.8	83.1	79.1	25.85	**12.9**
YOLOv9m (2024)	CSPDarknet-53	87.3	76.3	83.1	77.6	20.16	13.9
YOLOv10m (2024)	CSPDarknet-53	89.5	74.3	82.6	64.0	16.48	13.6
YOLOv11m (2024)	CSPDarknet-53	87.2	72.4	83.3	68.4	20.08	14.3
YOLOv12m (2025)	CSPDarknet-53	89	76.3	83.1	67.1	20.1	13.2
RT-DETR-R18 (2024)	ResNet-18	88.6	**80.2**	85	56.9	19.87	13.1
HAS-DETR (ours)	HFCSP	**91.7**	80.0	**87.2**	**45.0**	**8.14**	13.3

The data marked with bold represents the optimal performance metrics of the proposed model, while the data with underline denotes the suboptimal performance metrics

The comparative experimental results on the TT100K dataset show that HAS-DETR achieves the optimal performance among all models. As shown in the table, its Precision (P) and mAP@0.5 reach 90% and 86.6%, respectively, both being the highest. Meanwhile, it has the lowest number of model parameters (8.17M) and computational load (45.4G FLOPs). Specifically, compared with the baseline model RT-DETR-R18, our model improves mAP@0.5 by 1.8 percentage points, while the number of parameters and computational load are significantly reduced by 59% and 20.5%, respectively. Compared with the state-of-the-art model YOLOv9m, HAS-DETR's mAP@0.5 is 2.3 percentage points higher, while its parameter count is only 40% of YOLOv9m, thus fully demonstrating its superior performance and efficiency.

Notably, although HAS-DETR has achieved significant optimization in computational load and parameters, its inference latency does not decrease proportionally (13.1ms vs 12.7ms). This reflects the non-linear relationship between theoretical computational load and actual hardware execution efficiency. The primary reason is that our designed modules, such as HFCSP and ASSF, introduce more branches and fusion operations in pursuit of high accuracy and low parameters. While reducing computational redundancy, these operations also increase memory access cost (MAC) and may reduce computational parallelism—a common trade-off in lightweight model design involving multi-branch architectures. Therefore, this strategy—exchanging a slight increase in latency for a substantial reduction in model parameters and a significant improvement in detection accuracy—demonstrates the superior trade-off capability of HAS-DETR between model lightweighting and performance enhancement, thereby exhibiting stronger application potential in resource-constrained deployment scenarios.

This advantage is further verified on the CCTSDB dataset. HAS-DETR once again ranks first with a precision of 91.7% and mAP@0.5 of 87.2%, surpassing all models including the YOLO series and RT-DETR. It is worth noting that on both datasets, the inference latency of HAS-DETR (13.1ms and 13.3ms, respectively) is comparable to that of current optimal real-time detectors (such as YOLOv8m and RT-DETR), but achieves higher detection accuracy and lower model complexity.

Taking some samples from the TT100K dataset as examples, Fig. 9 visually compares the performance of the HAS-DETR model and the YOLOv10m model in traffic sign detection tasks. As a comparative model, YOLOv10m is a recently proposed end-to-end detection model that performs well in both detection accuracy and speed, as shown in the results in Tables 1 and 2. By comparing the detection results of HAS-DETR and YOLOv10, the difference in detection performance can be visually observed. The experimental results show that the HAS-DETR model outperforms the

Fig. 9 Comparison of detection results of different models (A: Original image; B: YOLOv10m; C: HAS-DETR)

YOLOv10m model in detecting real traffic signs. In images 355, 723, 2394, and 8320, the confidence scores of the detection boxes generated by the HAS-DETR model are significantly higher than those generated by the YOLOv10m model. In addition, the YOLOv10m model exhibits missed detections, for example, it only detects the traffic sign labeled i4 in image 33533, but in fact, i2r, i4, and i5 exist. Despite achieving higher accuracy in detecting real traffic signs, the HAS-DETR model is not without limitations. In some cases, it generates false positives, as shown in the red circle area of image 33533, where there is no i2 traffic sign, and some non-traffic sign objects are incorrectly classified as traffic signs. This indicates that the HAS-DETR model still faces challenges in distinguishing targets with similar visual features, highlighting areas for further improvement.

In summary, the experimental results fully demonstrate that the proposed HAS-DETR model achieves a better balance between accuracy, speed, and model size, especially showing significant advantages in model lightweighting, verifying its great potential and practical value for efficient and accurate real-time detection in real-world traffic scenarios.

4.5 Ablation experimental results

To further quantify the contribution of each module, we analyze the trade-offs from Table 3.

(1) HFCSP: Introducing HFCSP alone (Model 2 vs. 1) confirms its role as a lightweighting module. While it slightly reduces mAP by 1.5%, it achieves a substantial reduction in computational load (FLOPs by 23.5%) and model size (Params by 34.6%), significantly improving inference speed (Latency from 13.1ms to 12.5ms).
(2) ASSF: Introducing ASSF alone (Model 3 vs. 1) provides a significant +1.7% mAP gain, confirming its effectiveness in feature fusion, though at the cost of increased latency.
(3) STE: Introducing STE alone (Model 4 vs. 1) yields the highest accuracy gain of +2.3% mAP, validating its powerful capability in enhancing small-target detection. This gain, however, comes with a considerable increase in FLOPs and latency.
(4) Synergy: The most valuable insights come from module combinations. Model 8 (our full model) achieves the lowest FLOPs and Params by leveraging HFCSP, while ASSF and STE work together to secure high accuracy. Notably, HFCSP and STE (Model 6) achieves 88.0% mAP, which is 0.9% higher than STE alone (Model 4). This demonstrates a positive synergistic effect: the HFCSP backbone retains high-frequency details that provide better input for the STE head, allowing it to perform better than on the baseline backbone.

To assess the impact of the three proposed strategies (HFCSP, ASSF, and STE) on the performance of the RT-DETR model, ablation experiments were conducted—following standard protocol in detection research to isolate individual component effects. By sequentially removing these innovative components from the HAS-DETR model, the individual effects of HFCSP, ASSF, and STE on model performance were investigated—enabling quantitative analysis of each component's specific contribution. The experimental results on the TT100K dataset show that removing different components significantly affects the model's accuracy, computational load, parameter count, and detection speed. The specific experimental results are shown in Table 3.

Specifically, High-Frequency Enhanced CSP (HFCSP) balances model efficiency and performance by enhancing details and reducing redundant computations. From the comparison between Model 1 and Model 2, introducing HFCSP alone can substantially reduce the computational load and parameter count by 23.5% and 34.6% respectively, thus verifying its superior lightweight capability. A more valuable finding emerges from the combination experiment: comparing Model 3 (ASSF, Attention Scale Sequence Fusion) and Model 5 (HFCSP+ASSF), adding HFCSP to ASSF also results in a significant decrease in FLOPs and Params, thus proving that its optimization effect is universally applicable. The key point is

Table 3 Ablation experiments on the TT100K dataset

Model	HFCSP	ASSF	STE	mAP@0.5 (%)	FLOPs (G)	Params (M)	Latency (ms)
1				84.8	57.1	19.92	13.1
2	✓			83.3	**43.7**	13.03	**12.5**
3		✓		86.5	61.6	20.20	16
4			✓	87.1	78.5	18.62	19.6
5	✓	✓		85.6	49.1	13.36	14.7
6	✓		✓	**88**	65.9	11.74	16.5
7		✓	✓	85.4	58.0	15.02	18.9
8	✓	✓	✓	86.6	45.4	**8.17**	13.3

The data marked with bold represents the optimal performance metrics of the proposed model, while the data with underline denotes the suboptimal performance metrics

the comparison between Model 4 (STE, Small-Target Enhancement) and Model 6 (HFCSP+STE). Adding HFCSP not only reduces the parameter quantity from 18.62M to 11.74M, but also increases mAP@0.5 from 87.1% to 88.0%. This reveals that there is a positive synergistic effect between HFCSP and STE: HFCSP retains the high-frequency details, which provides a better input for STE's small-target detection head.

ASSF improves detection accuracy through adaptive feature fusion. The comparison between Model 1 and Model 3 shows that using ASSF alone can increase mAP@0.5 by 1.7 percentage points, confirming its basic effectiveness. When combined with the lightweight HFCSP (Model 2 vs. Model 5), it increases mAP@0.5 from 83.3% to 85.6% (+2.3%), indicating that it can effectively compensate for the potential accuracy loss of the lightweight backbone network.

STE focuses on improving small-target detection capability by introducing the S2 small-scale feature layer. The comparison between Model 1 and Model 4 shows that STE is the single module with the highest accuracy improvement, increasing mAP@0.5 by 2.3%. Adding STE on the basis of the HFCSP backbone network (Model 2 and Model 6) increases mAP@0.5 by 4.7%, fully illustrating that STE can exert its maximum potential when combined with a backbone network with strong detail retention capability. Similarly, introducing STE on the basis of the ASSF module (Model 3 and Model 7) also brings accuracy improvement and architecture optimization, especially playing a crucial role in handling challenging small-target scenarios.

To verify that the three improvements proposed in this paper can effectively enhance the performance of traffic sign detection, a comparative analysis of the model's feature heatmaps was conducted. In the feature heatmaps, the temperature level represents the model's attention to the corresponding area; higher temperature indicates higher attention of the model to that location. It is clear from the comparison in Fig. 10 that the feature heatmap of HAS-DETR can more accurately focus on the position of traffic signs, showing higher accuracy in attention to traffic sign positions. In addition, RT-DETR-R18 misjudges a part of the background as an ip-type traffic sign in image 45087. More critically, the feature heatmap of RT-DETR-R18 has obvious shortcomings, because it pays excessive attention to many background features unrelated to traffic signs, which can easily lead to misjudgments. In contrast, HAS-DETR shows good performance, effectively suppressing background features and thus

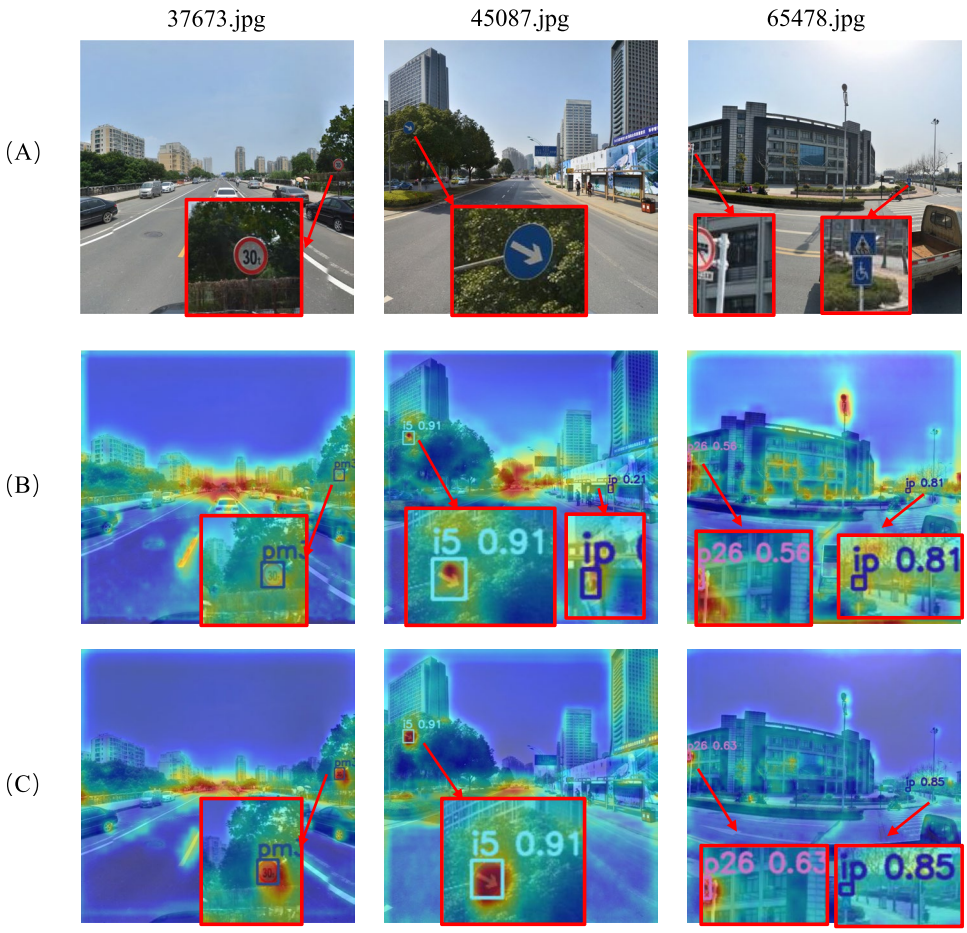

Fig. 10 Comparison of feature heatmaps (A: Original image; B: RT-DETR-R18; C: HAS-DETR)

focusing more on the detection and recognition of traffic signs.

5 Conclusion

In this study, we propose HAS-DETR, a high-performance real-time detection model for traffic scenarios, presenting a novel solution to the key challenges of accuracy, speed, and lightweighting in traffic sign detection. Based on the end-to-end RT-DETR architecture, the model is designed to meet the comprehensive requirements of high accuracy, lightweighting, and high efficiency for real-world traffic sign detection tasks through three novel components: the backbone network, feature fusion, and detection heads. First, we propose a High-Frequency Enhanced CSP Backbone (HFCSP-Backbone), which strengthens the capture of detailed information, such as image textures and edges through a unique residual structure, and combines with the CSP strategy to significantly reduce computational redundancy while maintaining strong feature expression capability, laying the foundation for model lightweighting. Second, in the feature fusion stage, HAS-DETR introduces an Attention Scale Sequence Fusion (ASSF) module, which dynamically models contextual relationships of multi-scale features, adaptively integrates global semantics and local details, and provides richer, more targeted features for the detector without significantly increasing the computational burden. In addition, to solve the problem of small-target detection, we embed a Small-Target Enhancement (STE) detection head in the Transformer decoder, which greatly improves the model's perception and localization accuracy of small-sized targets by introducing an additional high-resolution S2 feature layer. Experimental results demonstrate that compared with existing state-of-the-art models, HAS-DETR achieves a superior balance between detection accuracy, model lightweighting, and real-time detection speed, thereby exhibiting strong comprehensive performance and application potential in real-world traffic scenarios.

Beyond single-camera detection, the lightweight nature of HAS-DETR (8.17M parameters) makes it highly scalable for practical applications. This efficiency is crucial for vehicle-integrated systems, where multiple cameras (e.g., front, side, rear) must be processed in real time on a single embedded compute unit. The model's low computational footprint (45.4G FLOPs) suggests that multiple HAS-DETR instances could run concurrently, or a single model could process sequential frames from different cameras, without overwhelming the hardware. This scalability positions HAS-DETR as a promising solution for achieving 360-degree situational awareness in autonomous driving and multi-camera intersection monitoring.

However, despite the competitive performance of HAS-DETR, it still faces challenges in handling extremely complex and variable real-world traffic environments. As observed in our experimental results (e.g., image 33533 in Fig. 9), the model can still generate false positives. Detection performance may also be degraded under harsh conditions (e.g., severe weather, abrupt lighting changes), or when signs are severely occluded. Furthermore, our study primarily focused on the TT100K dataset, which, despite our filtering, presents challenges related to class imbalance that may affect generalizability to other regions or sign types not well represented. Finally, a key embedded deployment challenge remains: as noted in Table 1, our model's significant reduction in FLOPs and parameters does not perfectly translate to proportional latency improvements (13.1ms vs 12.7ms for the baseline), suggesting that our complex fusion modules may increase memory access costs. This limitation is compounded by the fact that our current validation was performed on a high-performance GPU (Tesla P100), not on resource-constrained edge hardware, which remains a critical next step.

In future work, we plan to explicitly address these limitations. This includes: (1) Expanding test datasets to include more examples under harsh conditions (e.g., fog, rain, and motion blur) to improve robustness; (2) investigating advanced data augmentation, domain adaptation, and loss-balancing techniques to mitigate dataset imbalance and enhance generalizability in open-world scenarios; (3) focusing on embedded deployment challenges; future work will explore network quantization, pruning, and hardware-aware architecture search to bridge the gap between theoretical efficiency (FLOPs) and real-world latency, ensuring efficient operation on resource-constrained devices.

6 Precision–recall curve comparison

To supplement the mAP metrics in Tables 1 and 2, Figs. 11 and 12 provide a visual comparison of the Precision–Recall (P–R) curves for our HAS-DETR model against the RT-DETR-R18 baseline and YOLO series on the TT100K and CCTSDB dataset.

The P–R curve illustrates the trade-off between precision and recall for different confidence thresholds. As shown in the figure, the curve for HAS-DETR is consistently above the other models, indicating a superior performance across all recall levels. This visual evidence confirms that our model's higher mAP score is a result of both higher precision at high recall levels and a more robust performance overall.

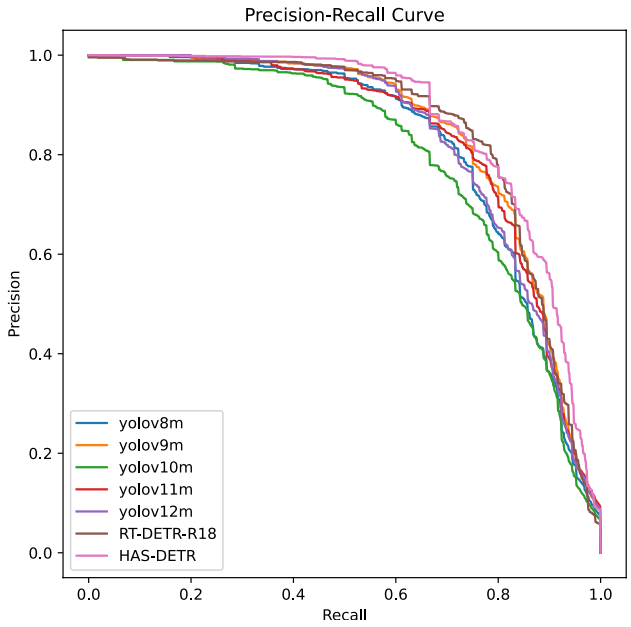

Fig. 11 P–R curve comparison on the TT100K dataset

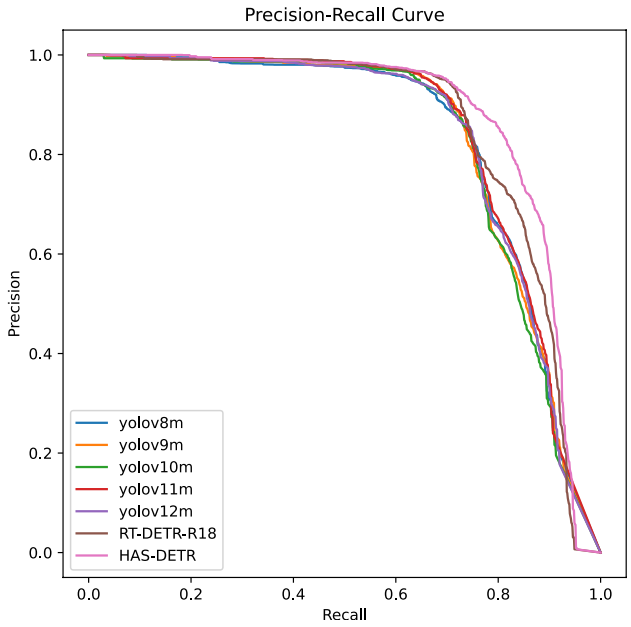

Fig. 12 P–R curve comparison on the CCTSDB dataset

Author Contributions JD: methodology, investigation, experimentation, data curation, and writing—original draft. HL: data curation, visualization, and writing—review & editing. XJ: validation, supervision, and writing—review & editing. All authors have read and approved the final manuscript.

Funding This research was funded by the Program of National Mineral Rock and Fossil Specimens Resource Center from MOST, grant number NCSTI-RMF20250106, and the Research Fund for Discipline Development Projects of China University of Geosciences Beijing, grant number 2024XK104. This research was supported by the High-performance Computing Platform of China University of Geosciences Beijing.

Data Availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Saadna, Y., Behloul, A.: An overview of traffic sign detection and classification methods. Int. J. Multimedia Inform. Retrieval **6**(3), 193–210 (2017)
2. Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., Hu, S.: Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2110–2118 (2016)
3. Edwards, D.J., Akhtar, J., Rillie, I., Chileshe, N., Lai, J.H., Roberts, C.J., Ejohwomu, O.: Systematic analysis of driverless technologies. J. Eng. Design Technol. **20**(6), 1388–1411 (2022)
4. Yang, Y., Luo, H., Xu, H., Wu, F.: Towards real-time traffic sign detection and classification. IEEE Trans. Intell. Transp. Syst. **17**(7), 2022–2031 (2015)
5. Ul Amin, S., Kim, B., Jung, Y., Seo, S., Park, S.: Video anomaly detection utilizing efficient spatiotemporal feature fusion with 3d convolutions and long short-term memory modules. Adv. Intell. Syst. **6**(7), 2300706 (2024)
6. Liu, X., Chu, R., Liu, B.: Tfg-net: a text feature-guided network for small traffic sign detection. In: IEEE Transactions on Neural Networks and Learning Systems (2024)
7. Carion, N., Massa, F.,Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision, pp. 213–229. Springer (2020)
8. Zhu, X., Su,W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv:2010.04159. (2020)
9. Yao, Z., Ai, J., Li, B., Zhang, C.: Efficient detr: improving end-to-end object detector with dense prior. arXiv:2104.01318. (2021)
10. Zhao, Y., Lv, W., Xu, S., Wei, J., Wang,G., Dang, Q., Liu, Y., Chen, J.: Detrs beat yolos on real-time object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 16 965–16 974 (2024)
11. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona,P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: common objects in context. In: European conference on computer vision, pp. 740–755. Springer (2014)
12. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788 (2016)
13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N. Kaiser, Ł., Polosukhin, I.: Attention is all you need. Adv. Neural Inform. Process. Syst. **30** (2017)
14. Brkic, K.: An overview of traffic sign detection methods, Zagreb: Department of Electronics. Microelectronics, Computer and

Intelligent Systems, Faculty of Electrical Engineering and Computing (2010)
15. Liu, C., Li, S., Chang, F., Wang, Y.: Machine vision based traffic sign detection methods: Review, analyses and perspectives. IEEE Access, vol. 7, pp. 86 578–86 596 (2019)
16. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv:1312.6229 (2013)
17. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587 (2014)
18. Girshick, R., Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 1440–1448 (2015)
19. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2016)
20. Liu, W., Anguelov, D., Erhan, D., Szegedy,C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision, pp. 21–37 . Springer (2016)
21. Wang, Y., Zhang, X., Yang, T., Sun, J.: Anchor detr: Query design for transformer-based detector. Proc. AAAI Confer. Artif. Intell. **36**(3), 2567–2575 (2022)
22. Amin, S.U., Jung, Y., Fayaz, M., Kim, B., Seo, S.: Enhancing pine wilt disease detection with synthetic data and external attention-based transformers. Eng. Appl. Artif. Intell. **159**, 111655 (2025)
23. Yao, T., Li, Y., Pan, Y., Mei, T.: Hgnet: learning hierarchical geometry from points, edges, and surfaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21 846–21 855 (2023)
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
25. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: Yolov4: optimal speed and accuracy of object detection. arXiv:2004.10934 (2020)
26. Li, A., Zhang, L., Liu, Y., Zhu, C.: Feature modulation transformer: Cross-refinement of global representation via high-frequency prior for image super-resolution. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 12 514–12 524 (2023)
27. Kang, M., Ting, C.-M., Ting, F.F., Phan, R.C.-W.: Asf-yolo: a novel yolo model with attentional scale sequence fusion for cell instance segmentation. Image Vis. Comput. **147**, 105057 (2024)
28. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8759–8768 (2018)
29. Zhang, J., Zou, X., Kuang, L.-D., Wang, J., Sherratt, R.S., Yu, X.: Cctsdb: a more comprehensive traffic sign detection benchmark. HCIS **12**, 2022 (2021)
30. Wang, C.-Y., Yeh, I.-H., Mark Liao, H.-Y.: Yolov9: learning what you want to learn using programmable gradient information. In: European conference on computer vision, pp. 1–21. Springer (2024)
31. Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., et al.: Yolov10: real-time end-to-end object detection. Adv. Neural Inform. Process. Syst. **37**, 107 984–108 011 (2024)