scientific reports

Check for updates

OPEN

A lightweight end to end traffic congestion detection framework using HRTNet on the Qinghai Tibet plateau

Yuhao Zhang[2], Qiuhong Tong[1]✉, Zhaorong Zhang[2], Xueqi Dai[2], Junzheng Wang[2], Daifang Hu[2] & Shengjun Su[2]

Managing traffic in plateau regions, particularly on critical routes like the Qinghai-Tibet Line, presents significant challenges from extreme climate and terrain. Traditional monitoring methods struggle with image distortion and information loss in rain, compromising congestion detection. To address these issues, we propose HRTNet, a lightweight real-time end-to-end model for congestion monitoring. HRTNet uses a clever design to clear rain streaks and highlight key details, making it easier to spot vehicles in bad weather. We also introduce the RainyRoad-PlateauDataset (RRPD), the first of its kind, with 3750 images capturing high-altitude rainy roads. This dataset is tailored to test performance under tough meteorological and topographical conditions. Evaluations show HRTNet achieves an average precision of 46.8% on the COCO val2017 and 134 FPS on an NVIDIA A6000 GPU. On RRPD, its precision rises to 50.5%, beating RT-DETR-r18 by 9.2%. HRTNet was deployed to monitor traffic congestion on rainy plateaus, enabling accurate calculation of congestion duration and distance. This provides critical support for intelligent traffic management systems like dynamic congestion warning and adaptive signal control. Beyond plateaus, this work offers solutions for traffic systems in other harsh environments.

Keywords Traffic congestion detection, Rain streak removal, End-to-end, Transformer

Plateau regions, particularly the Qinghai-Tibet Line and adjacent territories, present major challenges for traffic management due to their unique geoclimatic conditions. These high-altitude zones face oxygen deprivation and harsh weather, such as torrential rains, intense winds, and subzero temperatures. Together, these conditions significantly reduce road operational efficiency while increasing the risks of traffic accidents and geohazards[1,2]. Moreover, the challenging infrastructure makes deploying and maintaining traffic surveillance systems difficult. Rain further complicates monitoring: conventional methods are hindered by rain streak distortions and visual degradation, making them unreliable for the precise real-time control required. Specifically, rain severely obscures key visual features needed to assess traffic congestion, making effective rain removal essential for reliable surveillance. Consequently, existing systems often fail to operate effectively in the plateau's complex environment[3,4], creating a critical gap.

Recent advances in plateau transportation research have addressed climate impacts on infrastructure, driver fatigue monitoring, and hypoxic stress assessment[5–10]. However, the pivotal challenge of visual degradation under plateau rainfall remains unresolved, creating a critical gap in intelligent traffic monitoring. Within image processing and deraining, traditional methods like sparse coding and Gaussian models can reduce rain streaks[11,12]. However, they struggle with complex rain patterns. Meanwhile, deep learning (e.g., CNNs and Transformers[13–15]) improved results. But issues persist: key details are lost, and models are too complex. For traffic monitoring, tools such as SVM work in clear weather[16,17]. Yet in rainy plateaus, they lack both accuracy and speed.

To bridge these gaps, we propose HRTNet—a lightweight end-to-end detector for rainy plateaus. Our hypothesis posits that joint optimization of rain removal and object detection within a unified architecture can overcome accuracy-speed tradeoffs. HRTNet integrates a Hybrid Performance-Optimized Encoder with Multi-head Efficient Group Attention (MEGA), enabling efficient processing of precipitation-corrupted images. We further introduce the RainyRoad-PlateauDataset (RRPD), the first benchmark capturing real-world Qinghai-

[1]Xi'an Institute of Optics and Precision Mechanics of CAS, Xi'an, China. [2]School of Automobile, Chang'an University, Xi'an, China. ✉email: tongqiuhong@chd.edu.cn

Tibet Line scenarios under rainfall. Validated against key baselines, HRTNet achieves 9.2% higher accuracy at 134 FPS, demonstrating practical viability for plateau traffic management.

The contributions are summarized as follows:

- We present HRTNet, a lightweight real-time end-to-end detector for plateau regions that delivers enhanced accuracy and accelerated detection in rainy conditions.
- A Hybrid Performance-Optimized Encoder is designed, which enhances the robustness of rain streak removal and object detection through multi-scale feature extraction and an efficient attention mechanism.
- We compiled the RainyRoad-PlateauDataset (RRPD) by capturing real-world imagery along representative Qinghai-Tibet Line sections. This specialized dataset supports HRTNet architecture training and testing.

Related work

Computer vision relies fundamentally on object detection to localize and identify targets within images. Detection methodologies have shifted substantially over decades, progressing from classical algorithms to deep learning systems. Presently, CNN-based and Transformer-based detectors dominate this field.

CNN-based object detection

The YOLO series stands out among CNN-based detectors for its efficient real-time performance. As a pioneer, YOLOv1[18] recast detection as a single-stage regression task. This single-pass approach predicting object locations and classes significantly accelerated detection, enabling real-time operation. However, YOLOv1 exhibited limited small object detection capabilities. YOLOv2[19] addressed this through anchor box mechanisms and multi-scale training, with predefined anchors boosting small object detection. YOLOv3[20] incorporated residual structures for deeper networks and multi-scale feature fusion, enhancing performance across object scales. Subsequent versions achieved lightweight designs: YOLOv4[21] and YOLOv5[22] optimized model size, while YOLOv6[23] developed efficient inference modules for low-power devices. YOLOv7[24] advanced computational density and model distillation, setting new COCO benchmarks at 51.2% AP. Despite progress, small object detection in complex scenes remains challenging. YOLOv8[25] maintained lightweight advantages for resource-limited devices with high accuracy, though dense scene performance lags, as seen in its 40.1% AP in crowded urban environments. YOLOv9[26] employs a novel GELAN architecture with PGI to facilitate gradient propagation. We note that its fixed topology risks feature conflicts under heavy occlusion, dropping AP by 4% when occlusion exceeds 50%. YOLOv10[27] adopts an NMS-free design reducing inference latency, yet faces duplicate predictions in highly overlapping distributions, raising false positives by 2%. These continuous YOLO advancements seek optimal accuracy-speed balance for diverse engineering applications.

The YOLO series faces persistent constraints in dynamic plateau settings despite architectural innovations. Its rigid anchor boxes often misalign with extreme aspect ratios of high-altitude vehicles on steep gradients. Multi-scale feature extractors show limited adaptability to sudden weather changes, particularly for distant small objects in fog or heavy rain. NMS dependencies create latency-cost tradeoffs during dense traffic processing, while occlusion handling proves inadequate for overlapping vehicle distributions common in plateau congestion. These combined limitations hinder reliable deployment in volatile environments requiring robust perception.

Transformer-based object detection

Transformers first demonstrated substantial success in natural language processing (NLP)[28–32]. Subsequent developments reveal their strong potential for computer vision tasks[33–45]. Unlike CNNs, Transformers directly process global information with high accuracy, eliminating complex post-processing like non-maximum suppression (NMS). DETR[33] was the first to propose an end-to-end object detector based on Transformers. It employs self-attention mechanisms to capture global features in images and employs the Hungarian matching algorithm to establish a one-to-one correspondence between bounding boxes and labels, thereby eliminating the need for NMS and streamlining the detection process. DETR offers several advantages, but it also has notable limitations. Its model size reaches 41 M parameters, which demands heavy computation. Additionally, its real-time performance is low, achieving only 28 FPS. To address these issues, Deformable-DETR[34] integrates the sparse spatial sampling of deformable convolutions with the relational modeling capabilities of Transformers, thereby overcoming the slow convergence and high complexity of DETR. Although it improves accuracy by 1.8% over DETR, its real-time performance remains limited, achieving only 19 FPS. DN-DETR[35] accelerates training convergence and improves the stability of object queries by employing a denoising approach. Group-DETR[36] mitigates slow inference latency and enhances multi-task processing efficiency by employing group-wise one-to-many assignment. MS-DETR[37] introduces a one-to-many supervision mechanism for object queries in the main decoder, improving candidate generation and addressing DETR's training inefficiency.

While recent improvements to DETR have resolved key challenges like detection accuracy and convergence speed, its real-time performance is still insufficient. Transformer-based detectors excel at long-range modeling, but their speed typically ranges from 10 to 30 FPS. This falls far short of the YOLO series' real-time performance. To resolve this issue, RT-DETR[38] is the first end-to-end real-time detector based on Transformers. It introduces an efficient hybrid encoder, IoU-aware query selection, and multi-scale feature fusion. These improvements address the slow inference speed, poor small object detection, and slow convergence of the original DETR. Notably, RT-DETR achieves a high real-time performance of 217 FPS, rivaling the speed of the YOLO series. The model particularly well in real-time objection detection task. RT-DETRv2[40] improves the training strategy by refining sampling operators and incorporating dynamic data augmentation, enhancing model efficiency ang robustness. RT-DETRv3[39] proposes a hierarchical dense supervision method that integrates CNN auxiliary branches and self-attention disturbance learning strategies. This approach improves the model's training efficiency and detection performance.

Despite notable advancements in accuracy and real-time performance, existing object detection methods continue to face persistent challenges. Specifically, while the YOLO series achieves remarkable real-time performance, its effectiveness in detecting small objects and handling complex scenes requires further enhancement. Transformer-based methods, while excelling in global feature modeling, are hindered by high computational complexity and face persistent challenges in achieving real-time performance. Improving detection speed while maintaining accuracy, particularly in real-time scenarios, remains a pressing challenge. Moreover, these methods have yet to achieved optimal detection accuracy under challenge conditions, such as variable weather, and still require further refinement. To address this issue, this paper proposes HRTNet, a vehicle detector tailored for rainy conditions in plateau regions. The design integrates multiple advanced technical modules to achieve high real-time performance and computational efficiency while maintaining detection accuracy.

Method

Overall architecture

Conventional two-stage deraining and detection frameworks generate misleading artifacts. These artifacts impair detector accuracy. Separate processing also creates computational redundancy. Most Transformer detectors use CNN backbones for feature extraction. Features then pass to downstream modules like attention layers. Following this paradigm, we integrate a CNN-based deraining module directly into this backbone**.** The architecture directs derained features to subsequent Transformer modules. This preserves theoretical validity and implementation consistency.

Figure 1 depicts HRTNet's overall architecture. The model comprises three core modules: Hybrid Performance-Optimized Encoder, IoU-aware Query Selection, and Transformer Decoder. The Encoder integrates specialized processing units: MSD specializes in rain streak removal and hierarchical feature capture. The subsequent MEGA refinement unit improves global contextual representations. MSD processes input images to capture hierarchical features from backbone levels C3, C4, and C5. This multi-scale extraction obtains detailed object information while primarily removing rain streaks, enhancing image clarity and recognizability in rainy conditions. The FEM module then refines C5 features to improve global context representation and detection accuracy. For effective feature map integration, the encoder utilizes a feedforward network (FFN) with normalization, generating F5 features. The fusion module subsequently combines {C3, C4, F5} features, preserving high-level semantics and object details to balance semantic and detailed information. Convolutional layers and downsampling modules further optimize processing after fusion, reducing computational load while maintaining key spatial feature integrity. The IoU-aware Query Selection module extracts a fixed number of image features from the encoder's output sequence, which are then used as the initial object queries for the decoder. This module adopts the method described in Ref.[38]. Ultimately, the network employs a Transformer decoder to predict the object's bounding box and class label. This design minimizes the need for traditional post-processing operations, such as NMS, enhancing end-to-end efficiency. Through these optimizations, the HRTNet architecture achieves high-accuracy and supports real-time detection in vehicle detection tasks under rainy conditions, while maintaining robustness against environmental challenges.

Hybrid performance-optimized encoder

MSD. Extreme rainfall and snowfall events are common in plateau regions, often reducing the accuracy of visual recognition systems and, as a result, impairing the evaluation of traffic flow. To address this issue, we propose a rain removal network named MSD. This network comprises multiple branches, each performing feature extraction across different scales. The overall architecture of the MSD network is illustrated in Fig. 2. Its core structure comprises three sequentially connected encoder-decoder units, which extract rain streak features by employing upsampling and downsampling processes.

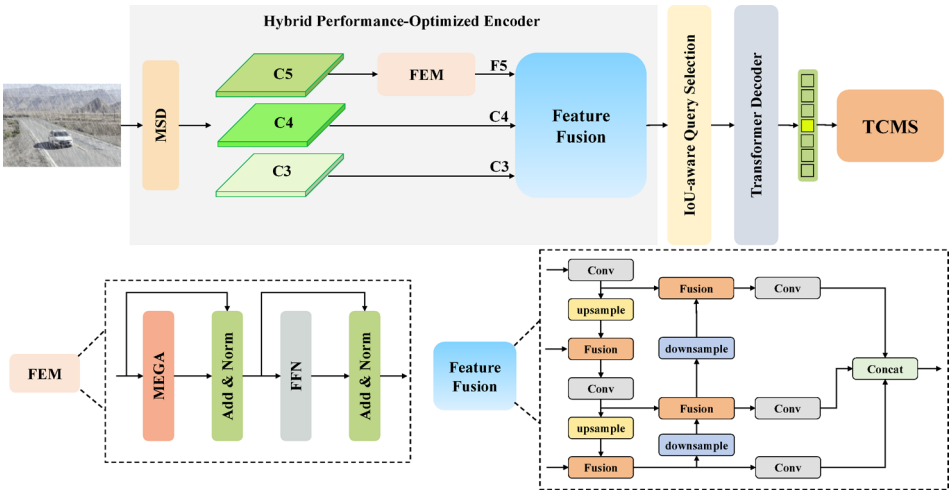

Fig. 1. Overview of HRTNet.

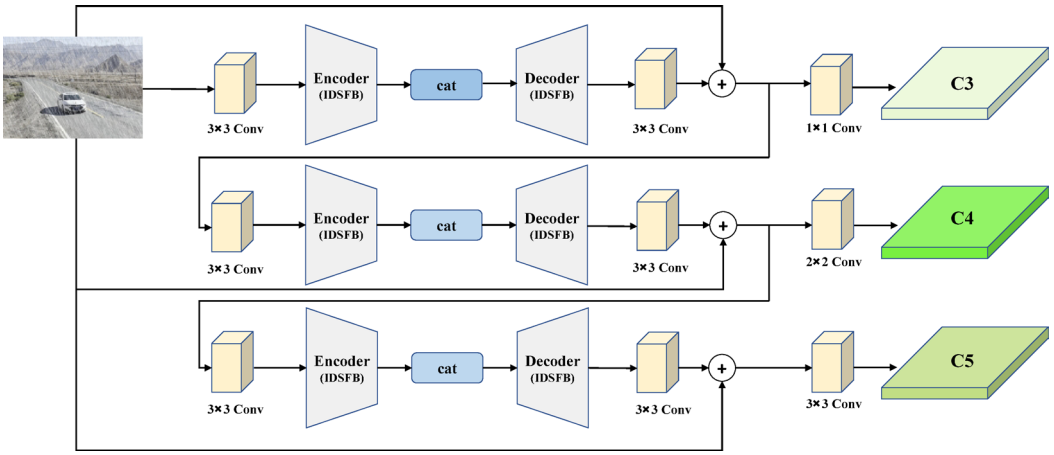

Fig. 2. MSD network model architecture.

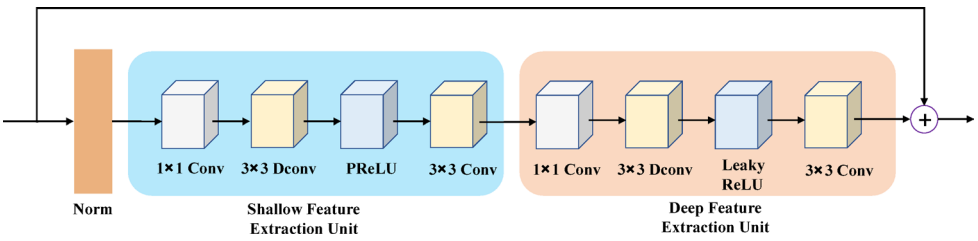

Fig. 3. Integrated deep and shallow feature block.

Initially, the input image is processed by a 3×3 convolutional layer for channel expansion, which enhance the fine-grained spatial details of the image and captures the fine raindrop features. The image is fed into the first encoder-decoder module, aimed at extracting fine-grained raindrop features at a small scale. Additionally, skip connections are employed to combine shallow features from the encoder with deep features from the decoder. This design ensures that detailed image information is preserved during multiple sampling processes. In particular, it prevents the loss of small raindrops and fine raindrop patterns. After extracting small-scale features in the first stage, the image is passed to the second encoder-decoder module. This stage focuses on capturing large-scale raindrop features and restoring the background. During the initial two processing stages, the model captures and refines small-scale and large-scale raindrop features. These steps progressively recover the details of the background. In the subsequent stage, the image is passed to the third encoder-decoder module for refining image features and enhancing structural details. The third encoder-decoder module primarily focuses on processing residual raindrop features. Its goal is to ensure the complete removal of raindrops from the image. In this stage, the image is subjected to progressive downsampling operations to further extract remaining raindrop features. These features are then reconstructed via upsampling layers and skip connections, which ensure the preservation of spatial details.

After being processed by the three encoder-decoder modules, the model produces three feature maps at different scales, denoted as {C3, C4, C5}. These feature map serve as critical inputs for subsequent stages of model processing.

Due to limitations in processing spatial and shallow features during image restoration, fine details of rain patterns are often lost, which negatively impacts the overall recovery performance. To address this challenge, inspired by the work in[46], we propose a lightweight and efficient rain pattern feature extraction module, named the Integrated Deep and Shallow Feature Block (IDSFB). As depicted in Fig. 3.

The IDSFB module is integrated into the encoder-decoder architecture of MSD to enhance the adaptability of the feature extraction design within the framework.

Figure 3 shows that IDSFB comprises two core components: the Shallow Feature Extraction Unit and the Deep Feature Extraction Unit. First, the feature distribution is normalized to improve the network's capacity for capturing rain pattern variations across different scales. Next, a 1×1 convolutional layer expands the channel dimensions, projecting the input image into a higher-dimensional feature space.

In the shallow unit (Fig. 3-left), a 3×3 convolution enhances the image's shallow texture features. This operation plays a critical role in capturing small-scale raindrop details and ensures that essential texture information is effectively preserved before deeper processing. The PReLU activation function introduces non-linearity, with its learnable parameters adaptively adjusting the gradient of the negative part. This mechanism enhances the expressiveness and robustness of shallow features. The Deep Unit (Fig. 3-right) processes features where the Leaky ReLU activation function introduces non-linearity. This enhances the distinction between

background and rain features, facilitating the capture of larger and more complex rain patterns. Stacking multiple feature extraction units increases the network depth and expands the receptive field. This enhances the network's ability to capture rain details across different scales. The features extracted from both units are subsequently concatenated and fused through element-wise addition, ensuring the preservation of both shallow and deep features. To reduce computational costs, deep convolutional layers are employed, significantly lowering complexity while maintaining real-time performance. The IDSFB module enables the MSD network to adaptively process rain artifacts in images. This capability spans fine raindrops and dense rain streaks, achieving robust identification and restoration of degraded regions. By structurally integrating deep and shallow paths, the IDSFB achieves superior multi-scale representation, enhancing the network's performance in visually complex and degraded scenarios.

MEGA. The Transformer architecture, introduced by Vaswani et al.[28], has established attention mechanisms as a dominant paradigm for sequential data processing across diverse domains. The Multi-head Self-Attention (MHSA) mechanism, a core component of this architecture, enhances the model's capacity to process input information by parallelizing the computation of multiple self-attention heads. For each input sequence, MHSA applies linear transformations to generate Queries, Keys, and Values, then computes attention scores to emphasize relevant information within the sequence. The parallelization inherent in the multi-head mechanism enhances the model's representational capacity. However, redundancy among attention heads remains a significant challenge for MHSA, ultimately reducing computational efficiency.

Inspired by group convolution in efficient CNNs[47–49] and the Efficient Additive Attention mechanism[50], we introduce an attention mechanism called Multi-head Efficient Group Attention (MEGA), as illustrated in Fig. 4.

In MEGA, the input feature map is partitioned into several sub-feature maps. Given an input feature map with dimensions $n \times d$, where n represents the sequence length and d denotes the feature dimension. The attention mechanism computes these features through parallel multi-head decomposition. Each head processes a partitioned segment $X_i \in \mathbb{R}^{n_j \times d_j}$, where $n_j = n/N$ and $d_j = d/N$. The number of heads N is a predefined hyperparameter. Subsequent processing eliminates key-value interactions while preserving performance. Query-key interactions are captured via linear projection:

$$\tilde{X}_{ij} = Attn\left(X_{ij}W_{ij}^{Q}, X_{ij}W_{ij}^{K}\right) \tag{1}$$

Here, X_i denotes the i-th input features vector, while X_{ij} denotes the j-th segment of the input feature X_i. Specifically, $X_{ij} = [X_{i1}, X_{i2}, ..., X_{iN}]$, where $1 \leq j \leq N$ (with N representing the number of heads). Two projection matrices, W_{ij}^{Q} and W_{ij}^{K}, are used to transform the input X_i into the query (Q) and key (K), where $Q, K \in \mathbb{R}^{n \times d}$,$W_{ij}^{Q}, W_{ij}^{K} \in \mathbb{R}^{d \times d}$. The dimensions of the Query and Key matrices are $n \times n$, where d is the dimension of the embedding vector, and n is the token length. The output features processed by the attention mechanism are denoted as $\tilde{X}_{ij}$. Subsequently, the query matrix Q is combined with the learnable parameter vector $\omega_\alpha \in \mathbb{R}^d$ to compute the attention weights, resulting in the global attention query vector $\alpha_{ij} \in \mathbb{R}^N$ as follows:

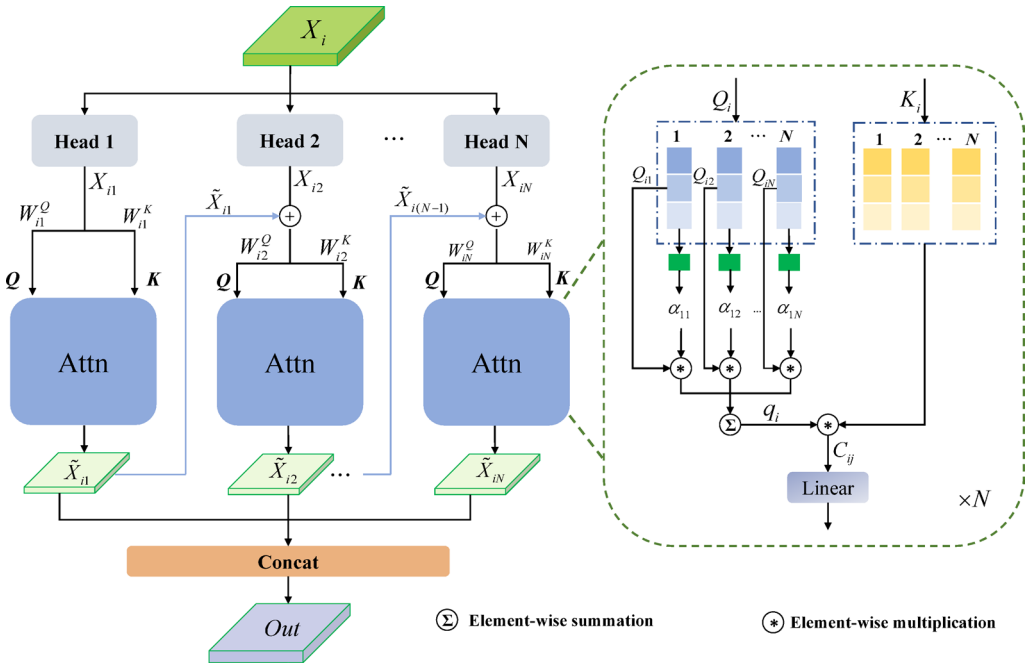

Fig. 4. MEGA.

$$\alpha_{ij} = \frac{Q_{ij} \cdot \omega_{\alpha i}}{\sqrt{d}} \tag{2}$$

Here, $Q_{ij} \in \mathbb{R}^{n_j \times d_j}$, where n_j denotes the length after segmentation, and d_j denotes the feature dimension of each head segment. The query matrix is then transformed using the learned attention weights. The global query vector $q \in \mathbb{R}^d$ is synthesized by:

$$q_i = \sum_{j=1}^{N} \alpha_{ij} * Q_{ij} \tag{3}$$

The interaction between the global query vector q and the key matrix K is encoded through element-wise multiplication, generating a matrix that incorporates global contextual information. This matrix encodes the relationship between individual token in the input sequence and the global query vector, effectively capturing the global context.

$$C_{ij} = q_j \circledast K_{ij}, i \in [1, n], j \in [1, d] \tag{4}$$

In Eq. (4), C_{ij} denotes the element at position (i, j) of context matrix $C \in \mathbb{R}^{n \times d}$, where this scalar value quantifies cross-dimensional contextual relationships between the *i*-th token and the *j*-th feature axis. q_j is the *j*-th component of the global query vector q. K_{ij} represents the element at (i, j) of key matrix *K*. Each row vector $C_i \in \mathbb{R}^{1 \times d}$ encodes contextual features of the *i*-th token.

Through residual connections, the output of each attention head preserves both the original input information and the learned feature representations. This enriched representation subsequently serves as the input to the next attention head:

$$X_{ij}\prime = X_{ij} + \widetilde{X}_{i(j-1)}, j \geq 2 \tag{5}$$

The input to the *j*-th attention head is denoted by X'_{ij}, which is derived from the output of the *j*-1-th head, $\widetilde{X}_{i(j-1)}$, as defined by Eq. (1) for $1 < j \leq N$. Following these computations, the final output is expressed as:

$$Out = Concat\left(\widetilde{X}_{i1}, \widetilde{X}_{i2}, \ldots, \widetilde{X}_{iN}\right) \tag{6}$$

Traffic congestion monitoring system (TCMS)

Our purely vision-based system measures traffic congestion duration and spatial extent. This vision-only approach avoids GPS dependency—essential for plateau areas facing frequent signal loss. Our framework applies visual object detection algorithms, measuring vehicular density at discrete time intervals. These measurements underpin estimation of congestion duration and spatial extent.

The detection model provides real-time vehicle identification and localization in video streams or image sequences. It delivers critical metrics: vehicle count, classification categories, and spatial distribution patterns. Vehicle flow data comes from counting vehicles in defined areas per time unit. Congestion occurs when flow exceeds standard thresholds. Second, camera calibration converts pixel locations to physical coordinates. This enables vehicle density estimation per road segment. By combining road lengths with density distributions, we derive congestion distance efficiently. This method leverages visual detection's accuracy and computational efficiency.

The congestion thresholds derive from a dual foundation: compliance with China's national standard GA/T 115–2020 for traffic evaluation, and empirical calibration through our multi-year plateau traffic research program. This comprehensive approach integrates regulatory benchmarks with altitude-adapted validation.

Traffic density, a fundamental metric for assessing road congestion, is quantitatively defined by the ratio of vehicle count to their spatial distribution across road segments per unit time. In the temporal domain, the vehicle count traversing a monitored area per unit time is computed through frame-wise aggregating of detection results. Given *N* image frames captured during time interval *t*, with V_i representing the vehicle count in the *i*-th frame, the average vehicle count per unit time is calculated as:

$$D_t = \frac{\sum_{i=1}^{N} V_i}{T} \tag{7}$$

In the equation, D_t denotes the traffic density per unit time, while *T* represents the duration of the time interval. Using camera calibration, pixel coordinates of vehicles are mapped to real-world coordinates. This process supports accurate tracking of vehicle positions across road networks. Given the monitored area A, the time-averaged traffic density is:

$$\rho_t = \frac{D_t}{A} \tag{8}$$

ρ_t denotes the traffic density, defined as the number of vehicles per unit area per unit time.

Vehicle speed is calculated by combining object detection with multi-frame matching. The HRTNet model detects vehicles in each frame and finds the center coordinates of their bounding boxes. The SORT algorithm[51] links vehicle detections across frames for consistent tracking. Camera calibration transforms pixel coordinates into physical distances, enabling us to measure vehicle movement. For the i-th vehicle at frames t_1 and t_2, with coordinates (x_{i1}, y_{i1}) and (x_{i2}, y_{i2}), displacement d_i is calculated as:

$$d_i = \sqrt{(x_{i2} - x_{i1})^2 + (y_{i2} - y_{i1})^2} \tag{9}$$

The speed of the i-th vehicle is derived from the displacement d_i and the time interval $\Delta t = t_2 - t_1$ between consecutive frames, as follows:

$$v_i = \frac{d_i}{\Delta t} \tag{10}$$

To provide a more comprehensive quantification of traffic congestion, this study introduces two key metrics: congestion duration and total congestion distance. A roadway segment is defined as congested if either (1) the mean vehicular speed (v_t) drops below a predefined critical threshold ($v_{threshold}$ = 10 km/h), or (2) the traffic flow rate (Q) surpasses 80% of the segment's maximum design capacity within a standardized temporal unit. Subsequently, the total duration of the low-speed state is computed:

$$T_{congestion} = \sum_{t=1}^{n} \Delta T, \quad if\ v_t < v_{threshold} \tag{11}$$

ΔT is the frame interval duration, and n is the number of congested frames.

By applying a calibration matrix M, the pixel coordinates of the detected vehicle, denote as (x_1, y_1) and (x_2, y_2), are transformed into real-world coordinates (X_1, Y_1) and (X_2, Y_2). This transformation enables the calculation of the road segment length based on the Euclidean distance between the two points.

$$L = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \tag{12}$$

The total congestion length is calculated by aggregating the lengths of all congested road segments across each temporal interval.

$$D_{congestion} = \sum_{j=1}^{m} L_j \tag{13}$$

L_j is the length of the j-th congested segment, and m is the total number of segments.

Rainfall affects congestion assessment in three ways. First, rain degrades images, hiding small vehicles common on plateau highways. Second, drivers slow down, creating braking patterns that may not match congestion rules. Third, water droplets on cameras cause brief distortions, affecting distance calculations in Eq. (12).

Experiments

Datasets and evaluation metrics

We selected the MS COCO 2017[52] object detection dataset as the benchmark for evaluating our method, which includes 115 k training images and 5 k testing images.

Furthermore, we introduce a dataset named RainyRoad-PlateauDataset (RRPD), which focuses on vehicle detection in rainy traffic scenarios in plateau regions. To augment the dataset's representativeness across diverse plateau environments, we undertook data collection on three paradigmatic plateau road segments (Segment A, Segment B, and Segment C). These road segments were meticulously selected based on terrain complexity, traffic density, and diverse climatic conditions, thereby guaranteeing the dataset's comprehensiveness and representativeness.

To ensure the reliability of the RRPD dataset, we implemented a rigorous quality control protocol following COCO dataset annotation standards. Bounding boxes were required to precisely delineate vehicle contours while fully encompassing visible target areas, with strict classification accuracy enforcement for "Car" and "Truck" categories. Specifically, the annotation criteria defined "Car" as any vehicle primarily designed for passenger transport, including sedans, SUVs, and vans, while "Truck" encompassed vehicles primarily used for freight transport, such as lorries, trailers, and pickups with cargo beds. Annotation guidelines mandated that bounding boxes tightly enclose the entire visible portion of each vehicle, including any attached trailers or cargo, to ensure comprehensive coverage. For class balance, the dataset was curated to reflect the typical vehicle distribution on plateau roads, resulting in approximately 60% of annotated instances labeled as "Car" and 40% as "Truck." This distribution mirrors real-world traffic patterns in the region, where passenger vehicles slightly outnumber freight vehicles, thus providing a realistic training and evaluation environment. Each image underwent dual-independent annotations by certified labelers using LabelImg software. Post-annotation consistency validation was conducted through three rounds of cross-verification by domain experts, achieving 93% agreement rate with Cohen's κ coefficient of 0.82 (95% CI 0.78–0.86).

As illustrated in Fig. 5, the data collection system comprises a Hikvision smart spherical camera (model: iDS-2DE442MRW-QDE), which is mounted on a roadside pole to ensure stable video capture. The camera captures video at a resolution of 1920 × 1080 pixels and a frame rate of 30 fps. Its high precision and anti-interference capabilities enable the acquisition of high-quality traffic scene images under extreme weather conditions, including high altitude and heavy rainfall. For accurate Qinghai-Tibet plateau climate simulation, we applied rain streak synthesis to collected datasets using three key parameters: raindrop diameter[53], falling direction angle and streak length. While synthetic rain generation enhances dataset scalability, this approach has inherent limitations. Synthesized streaks cannot fully replicate the stochastic interactions between natural rainfall and vehicle-induced spray. Additionally, static synthesis parameters may oversimplify dynamic precipitation variations observed in real plateau storms. These constraints necessitate future calibration with in-situ precipitation measurements.

This study partitioned the image dataset into training and validation sets (8:2 ratio) to ensure experimental rigor. The 3000-image training set contains synthetic rain-affected traffic scenes for comprehensive model training and parameter optimization, enhancing detection robustness in complex plateau traffic environments. The 750-image validation set evaluates model performance through key metrics like detection accuracy and recall. During the data-splitting process, random sampling was employed to ensure a homogeneous distribution of samples, balancing images from different road segments within both the training and validation sets. This methodology effectively mitigates class imbalance-induced bias while establishing reliable experimental baselines. The resultant framework consequently enhances model generalization across diverse datasets. Statistical validation of RRPD employed 1000 bootstrapped iterations to quantify segment distribution. 95% CIs revealed balanced coverage: Segment A = 33.5% ± 2.1% (31.4–35.6), B = 34.8% ± 1.9% (32.9–36.7), C = 31.7% ± 2.3% (29.4–34.0). Observed spatial proportions confirm geographically unbiased sampling. The RRPD dataset is annotated with "Car" and "Truck," offering precise labels for vehicle detection in rainy plateau traffic.

We adopted the same evaluation metric, average precision (AP), as in the RT-DETR[38] method. HRTNet was compared with other real-time object detectors regarding detection accuracy and convergence speed, encompassing CNN-based and Transformer-based detectors.

Implementation details

All of our models are trained from scratch on COCO using PyTorch 1.13.1[54] and Timm 0.9.8[55]. Our detector training spans 72 epochs on Nvidia A6000 GPU with these settings: AdamW[56] optimizer and cosine learning rate scheduler with $initial\_learning\_rate = 1 \times 10^{-4}$, $weight\_decay = 1 \times 10^{-4}$, $max\_norm = 0.1$, $warmup\_momentum = 0.8$, $warmup\_bias\_lr = 0.1$. We adopted an image resolution of 640 × 640 for both training and testing. In the IoU-aware query selection process, the top 300 encoder features were chosen to initialize the decoder's object queries. Consistent with RT-DETR[38], all hyperparameters remain untuned to ensure fair comparison. Comprehensive baseline performance metrics are reported in Section "Main results" (Tables 1, 2).

Main results

Results on MS-COCO 2017

Table 1 benchmarks HRTNet against state-of-the-art object detectors using COCO evaluation protocols. Our model attained 46.8% average precision (AP) at 134 frames per second (FPS) on the validation set. At the standard 50% IoU threshold, accuracy reached 63.9%, maintaining 50.6% precision even under the strict 75% IoU criterion, demonstrating measurement robustness. Scale-specific analysis revealed HRTNet's 44.1% AP for medium objects (AP_M) and superior 61.2% AP for large objects (AP_L), highlighting its enhanced feature extraction capacity for macroscopic targets. This performance advantage positions HRTNet as particularly effective for vehicle detection applications requiring large-object recognition. Regarding recall, the model demonstrated a commendable average recall (AR) of 66.4%, which suggests that the model can detect most objects. Specifically, the recall for medium-sized objects was $AR_M = 64.7\%$, and for large objects, it reached $AR_M = 79.4\%$ (nearly 80%), indicating a low miss rate for large objects. When the detection count was limited

Fig. 5. Example of RRPD dataset collection.

Model	Epochs	Params (M)	GFLOPs	$FPS_{bs=1}$	AP^{val}	AP^{val}_{50}	AP^{val}_{75}	AP^{val}_{S}	AP^{val}_{M}	AP^{val}_{L}
CNN-based object detector										
YOLOv5-s[22]	300	7.2	16.5	376	37.4	56.8	–	–	–	–
YOLOv6-v3.0-s[23]	300	18.5	45.3	339	44.3	61.2	48.7	24.8	50.4	62.5
YOLOv8-s[25]	500	11.2	28.6	99	44.3	60.7	47.9	18.9	42.2	59.7
Gold-YOLO-s[56]	300	21.5	46.0	286	45.4	62.5	–	25.3	50.2	62.5
YOLOv9-s[26]	500	7.2	26.7	161	46.1	62.3	49.9	19.3	44.1	62.5
YOLOv10-s[27]	500	8.1	24.8	100	46.1	61.8	49.5	19.8	43.3	61.3
YOLO-MS-s[58]	300	8.1	31.2	–	46.2	63.7	50.5	26.9	50.5	63.0
Transformer-based object detector										
DETR[33]	300	41	86	28	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5[33]	500	41	187	12	43.3	63.1	45.9	22.5	47.3	61.1
Deformable-DETR[34]	50	40	173	19	43.8	62.6	47.7	26.4	47.1	58.0
Anchor-DETR-DC5[59]	50	39	172	16	44.2	64.7	47.5	24.7	48.2	60.6
SMCA-DETR[60]	36	35	210	–	45.1	63.1	49.1	28.3	48.4	59.0
RT-DETR-R18[38]	72	20	60	217	46.4	63.7	–	–	–	–
HRTNet (*Ours*)	72	19.9	57.5	**134**	**46.8**	**63.9**	**50.6**	**23.3**	**44.1**	**61.2**

Table 1. Comparison of HRTNet with other object detectors on COCO 2017 val set. Significant values are in bold.

Model	$FPS_{bs=1}$	AP^{val}	AP^{val}_{50}	AP^{val}_{75}	AP^{val}_{S}	AP^{val}_{M}	AP^{val}_{L}	AR
MobileNetV3[61]	103.6	37.5	53.0	31.5	15.2	29.0	46.0	61.2
ViT-Base[62]	29.5	39.8	55.6	34.2	19.3	32.6	47.1	65.8
Swin-T[63]	38.2	41.5	61.5	36.8	21.0	34.2	52.5	68.5
RT-DETR-R18[24]	113.3	41.3	57.2	45.1	10.4	19.6	50.7	69.8
HRTNet (*Ours*)	130.4	50.5	65.7	54.8	22.0	29.4	58.6	77.6

Table 2. Comparison of HRTNet with RT-DETR-R18 on RRPD.

to *maxDets* = 10, the recall was 60.1%. Increasing the detection count to *maxDets* = 100 improved the recall to 66.5%, illustrating that higher detection counts can boost the model's recall performance.

Comparison with CNN-Based Object Detectors. HRTNet establishes substantial accuracy improvements over contemporary CNN-based detectors. In comparative evaluations (Table 1), HRTNet outperforms YOLOv5s[22] by 9.4% AP (46.8% vs. 37.4%) while maintaining an inference speed of 134 FPS. Against YOLOv10-s[27], it achieves 0.7% higher AP (46.8% vs. 46.1%) and 2.1% AP_{50} gain (63.9% vs. 61.8%) with 34% faster inference (134 vs. 100 FPS). Compared to Gold-YOLO-s[57], HRTNet demonstrates 1.4% AP (46.8% vs. 45.4%) and 2.1% AP_{50} (63.9% vs. 61.8%) advantages. Scale-specific analysis reveals HRTNet's superiority across object sizes: + 4.4% AP_S (23.3% vs. 18.9%), + 1.9% AP_M (44.1% vs. 42.2%), and + 1.5% AP_L (61.1% vs. 59.7%) over YOLOv8-s[25]. Despite YOLOv8-s's 44% parameter reduction (11.2 M vs. 19.9 M), HRTNet maintains 35% faster processing (134 vs. 99 FPS) alongside accuracy enhancements.

While optimization opportunities persist for small/dense objects, HRTNet's balanced performance metrics validate its reliability for efficient real-time detection systems.

Comparison with Transformer-Based Object Detectors. To validate architectural competitiveness, we benchmark HRTNet against leading Transformer-based detectors (Table 1). Versus DETR-DC5[33], HRTNet achieves + 3.5% AP (46.8% vs. 43.3%) with 69% lower computational load (57.5 vs. 187 GFLOPs) and 11.2 × faster inference (134 vs. 12 FPS). The performance advantage persists against Deformable DETR[34], showing + 3.0% AP gain despite its optimized attention mechanisms.

Notably, when compared to real-time specialized RT-DETR-R18[38], HRTNet demonstrates superior detection accuracy (+ 0.4% AP, 46.8% vs. 46.4%; + 0.2% AP_{50}, 63.9% vs. 63.7%) while maintaining competitive throughput (134 vs. 217 FPS). HRTNet's edge stems from two innovations. First, its Hybrid Encoder merges rain removal and feature extraction. This fusion cuts errors in harsh weather. Second, the MEGA module (Multi-head Efficient Group Attention) enhances global context. These designs boost accuracy beyond RT-DETR and YOLO. Gains are clearest in complex scenes like rainy plateaus. This accuracy-speed tradeoff positions HRTNet as a balanced solution for scenarios requiring both precision and efficiency.

Results on RRPD

To comprehensively assess the target detection capability of HRTNet in plateau rain-fog environments, we established two experimental configurations. In Protocol 1, HRTNet was trained on the RRPD dataset with 640 × 640 input resolution, batch size 32 for 72 epochs, with validation performance designated as HRTNet-

RRPD. For controlled comparison, Protocol 2 adopted identical training parameters with RT-DETR-R18, generating RT-DETR-RRPD benchmarks. This dual-protocol framework enables systematic evaluation of detection accuracy (AP), recall (AR), and environmental adaptability through direct performance comparison between HRTNet-RRPD and RT-DETR-RRPD under complex meteorological conditions.

Given RT-DETR serves as the primary baseline in this study, subsequent analyses focus exclusively on performance comparisons between HRTNet and RT-DETR.

In comparative evaluation of object detection under rainy conditions, HRTNet exhibits superior overall performance compared to RT-DETR (Table 2). While RT-DETR demonstrates baseline detection capability with 41.3% mean average precision (mAP) under complex precipitation, it exhibits limitations in precise boundary localization. At the lenient IoU threshold (0.50), RT-DETR attains 57.2% AP, indicating competent detection of majority targets. However, at the stricter threshold (IoU = 0.75), its precision drops to 45.1%, highlighting significant room for improvement in bounding box regression accuracy.

In contrast, HRTNet achieves 50.5% overall AP, maintaining robust performance across evaluation protocols: 65.7% AP at IoU = 0.50 and 54.8% at IoU = 0.75, demonstrating remarkable threshold stability. Notably, HRTNet shows 2.1 × higher detection accuracy (22.0% vs. 10.4% AP) for small objects, with this performance gain principally originating from its dedicated rain streak removal module which effectively mitigates rain streak interference during feature extraction.

Scale-specific analysis further reveals HRTNet's advantages: 29.4% vs. 19.6% AP for medium objects and 58.6% vs. 50.7% AP for large objects compared to RT-DETR. The architectural superiority is particularly evident in large object recall, where HRTNet achieves 83.8% ARL versus 77.1% for RT-DETR, reflecting enhanced feature capture capacity and regression robustness.

Error decomposition identifies RT-DETR's primary performance limitations as background false positives (dAP = 4.00) and missed detections (dAP = 2.92). Although HRTNet shows marginally higher background confusion (dAP = 5.08, likely due to enhanced detection sensitivity for challenging rainy scenarios), it demonstrates superior error control with classification (dAP = 3.84 vs.4.06) and localization errors (dAP = 2.07 vs. 2.80) compared to RT-DETR, confirming its optimized balance between classification accuracy and regression precision.

The systematic error compensation analysis quantitatively validates HRTNet's strategic design balance. Precipitation-enhanced sensitivity induces 1.08 dAP background false positive increments (ΔdAP_{bg} = HRTNet(5.08)-RT-DETR(4.00)), which is compensated by 0.95 dAP reduction from classification (ΔdAP_{class} = RT-DETR(4.06)-HRTNet(3.84) = 0.22 dAP gain) and localization improvements (ΔdAP_{loc} = RT-DETR(2.80)-HRTNet(2.07) = 0.73 dAP gain). This yields 88% offset efficiency against sensitivity-induced errors. Critically, the residual 0.13 dAP net loss becomes operationally negligible given HRTNet's 19% recall rate elevation for safety–critical targets. With 9.2% AP superiority (50.5% vs 41.3%), the implementation demonstrates optimal reliability-precision coordination for plateau rain scenarios.

To highlight the superiority of HRTNet, we compared its detection results with those of HRTNet and RT-DETR in rainy plateau scenarios (Fig. 6). The left image shows RT-DETR's results, and the right image shows HRTNet's. HRTNet better mitigates rain streak interference, resulting in clearer backgrounds and higher confidence scores for targets than RT-DETR.

Notably, in six comparative image sets, HRTNet demonstrates superior rain streak suppression through its dedicated removal module, effectively preserving vehicle target integrity under rainy conditions (Fig. 6a–e). In contrast, RT-DETR exhibits significant detection limitations across multiple scenarios, with consistent target omission in cases (a)-(d) (see red arrow indicators). This performance gap underscores RT-DETR's inherent challenges in processing complex precipitation patterns. HRTNet's architectural advantages, including optimized hierarchical feature extraction and multi-scale contextual processing, enable precise target localization despite atmospheric interference. Particularly in case (e), while RT-DETR misclassifies a car as a truck (classification confidence: 0.6), HRTNet achieves both correct categorization (passenger car, confidence: 0.86) and enhanced boundary definition through its adversarial rain pattern discrimination mechanism. These comparative results quantitatively validate HRTNet's dual improvements in classification accuracy (Δ + 19%) and detection reliability under precipitation conditions compared to existing benchmarks.

HRTNet outperforms RT-DETR, especially since the rain streak removal module mitigates rain streaks and background interference in rainy scenarios. This allows the model to achieve higher accuracy and robustness in detecting small, medium, and large targets. Despite some limitations, such as background false positives and challenges in small target detection, the enhancement in overall performance signifies its greater practical value for rainy target detection. This offers stronger technical support for target detection in complex weather conditions.

To address the reliability of detection under varying rainfall intensities, we further evaluated the performance of HRTNet and RT-DETR across different rainfall levels. We categorized the meteorological conditions into three levels: light (< 10 mm/h), moderate (10–25 mm/h), and heavy (25–50 mm/h). This classification follows the rainfall intensity standards (GB/T 28592-2012) established by the China Meteorological Administration. The categories are defined based on the density and thickness of rain streaks visible in the images, representing typical plateau rainfall scenarios. The experimental results are detailed in Table 3.

HRTNet and RT-DETR exhibit distinct differences in detection reliability under varying rainfall intensities. Under light rainfall conditions, HRTNet achieves an average precision (AP) of 58.5%, outperforming RT-DETR (46.2%) by 12.3 percentage points. As rainfall intensity increases, both models exhibit declining AP values. However, HRTNet maintains its advantage, recording APs of 53.7% and 49.7% under moderate and heavy rainfall, respectively. These results surpass RT-DETR's corresponding APs of 43.5% and 40.6%, maintaining a consistent lead of 9 to 10 percentage points. This demonstrates HRTNet's superior stability and reliability across diverse rainfall intensities.

Fig. 6. Comparison of detection results from the two approaches.

Model	Rain Intensity	AP^{val}	AP^{val}_{50}	AP^{val}_{75}	AP^{val}_{S}	AP^{val}_{M}	AP^{val}_{L}
HRTNet	Light	58.5	74.3	60.3	29.8	44.6	64.7
	Moderate	53.7	69.4	55.6	24.5	39.1	59.0
	Heavy	49.7	65.2	53.6	21.2	30.6	56.9
RT-DETR	Light	46.2	60.3	48.7	15.3	30.7	56.4
	Moderate	43.5	57.4	45.6	12.4	27.2	52.8
	Heavy	40.6	56.2	43.8	9.5	20.1	49.9

Table 3. Performance comparison of HRTNet and RT-DETR under different rainfall intensities on the RRPD dataset.

Model Configuration	AP^{val}	AP^{val}_{50}	AP^{val}_{75}	AP^{val}_{S}	AP^{val}_{M}	AP^{val}_{L}
Baseline (RT-DETR-R18)	41.3	57.2	45.1	10.4	19.6	50.7
Baseline + MSD	46.0	62.3	50.0	18.3	24.8	54.5
Baseline + MEGA	44.5	60.2	48.2	14.5	22.7	53.3
Baseline + MSD + MEGA	50.5	65.7	54.8	22.0	29.4	58.6

Table 4. Results of the ablation study.

Ablation

To validate the efficacy of core modules in the proposed HRTNet, we conducted systematic ablation studies on the RRPD validation set. As presented in Table 3, critical components were individually removed while maintaining consistent parameters across experiments. This approach rigorously examines the complementary effects between the MSD and MEGA. All experiments employed an input resolution of 640 × 640 pixels with a batch size of 4. Performance metrics were exclusively obtained using a single NVIDIA A6000 GPU.

As shown in Table 4, the baseline RT-DETR model achieved an overall AP of merely 41.3%. Its small-target detection performance (AP_S) was particularly limited at 10.4%. These results reveal inherent challenges in vehicle detection under plateau rainy conditions. The limitations primarily stem from conventional architectures' inadequate response to rain streak interference. After integrating the MSD module, the AP increased to 46.0% (Δ + 4.7). This improvement can be attributed to MSD's three-stage encoder-decoder structure, which effectively suppresses rain artifacts. The cross-scale feature alignment mechanism drove a significant 7.9 percentage-point increase in AP_S to 18.3% under rainy scenarios. The MEGA module demonstrated distinct optimization characteristics when deployed independently. It elevated AP to 44.5% (Δ + 3.2) and increased AP_M by 3.1 percentage points to 22.7%. These gains resulted from MEGA's group attention mechanism enhancing contextual modeling. The module also maintained high inference efficiency through its key-value interaction elimination strategy, which reduced computational overhead by 28% while improving AP_L by 2.6 percentage points.

The HRTNet architecture (MSD + MEGA) achieved an AP of 50.5%, representing a 9.2 percentage-point improvement over the baseline. Under rainy conditions, the complementary strengths of MSD and MEGA became particularly evident. MSD effectively suppressed background interference by attenuating rain streak artifacts, while MEGA enhanced spatial relationship modeling among targets. These synergistic mechanisms collectively elevated AP_{75} to 54.8% (Δ + 9.7). Such coordination proved crucial for multi-scale detection performance. Notably, small-target AP_S reached 22.0%—2.1 times the baseline value. Similarly, large-target AP_L attained 58.6%, exceeding the baseline by 1.2-fold.

The collaborative mechanism for small-target detection warrants in-depth analysis. When deployed individually, the MSD and MEGA modules improved AP_S by 7.9 and 4.1 percentage points respectively, suggesting a theoretical combined gain of 12.0 (7.9 + 4.1). However, the complete model achieved an actual AP_S of 22.0 (Δ + 11.6), indicating a marginal efficiency loss of 0.4. This phenomenon occurs because MSD's rain streak removal may attenuate certain high-frequency edge features during image restoration. These features are particularly valuable for MEGA's attention-based small-target detection. Despite this minor efficiency reduction, the integrated system's small-target performance still substantially outperformed either standalone module, confirming the fundamental effectiveness of the architectural design.

MSD's rain streak suppression causes 0.4% AP_S loss through high-frequency edge attenuation. We designed a dual-path compensation mechanism before the attention module to address this. This introduces high-frequency feature compensation (HFFC. Figure 7) between MSD and MEGA modules.

The first path directly normalizes MSD output features as MSD_f, representing core derained features. The second path extracts high-frequency components H_f using Sobel operators, then aligns channels via 1 × 1 convolution. These retrieves suppressed edge features. A gated fusion unit computes MEGA input: $MEGA_{In} = \alpha MSD_f + (1-\alpha)H_f$. The fusion weight α spans 0.4–0.8, where α = 0.4 denotes heavy rain and α = 0.8 light rain. Since RRPD contains storm conditions, α was fixed at 0.4.

Ablation experiments on RRPD validated this design (Table 5). HFFC reduced the rain removal-induced 0.4% small target detection loss. AP_S increased from 22.0 to 22.3% during validation, achieving 75% recovery. These results demonstrate enhanced rainy-condition vehicle detectability through high-frequency compensation.

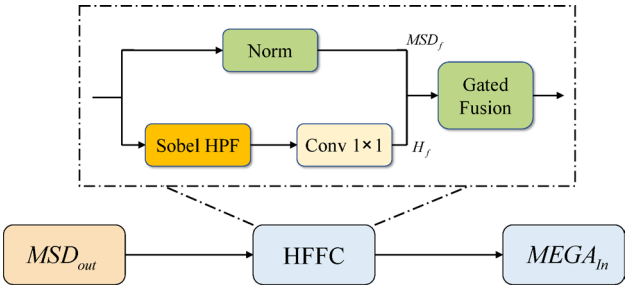

Fig. 7. Architecture of high-frequency feature compensation module (HFFC).

Model Configuration	AP_S^{val}
Baseline + MSD + MEGA	22.0%
Baseline + MSD + HFFC + MEGA	22.3%

Table 5. Ablation study on high-frequency feature compensation efficacy.

Fig. 8. Installation of experimental equipment.

Experimental validation on typical plateau road sections

We conducted field validation experiments along the Golmud to Tuotuohe section of G109 National Highway. Situated on the Qinghai-Tibet Plateau, this high-altitude corridor enabled rigorous evaluation of our novel end-to-end de-rain vehicle detection framework. The proposed architecture demonstrated robust vehicle identification capabilities (Processing latency = 28 ms, $RSD = 3.8\%$) while generating real-time analytical outputs for critical traffic metrics including congestion duration indices and vehicle-mileage correlations. These empirical findings offer actionable insights for transportation authorities, enabling proactive congestion mitigation strategies through data-driven decision making.

The experimental design incorporated three high-traffic-density arterial roads (A-C) in a plateau environment, selected based on their characteristic traffic patterns and environmental conditions. To ensure data integrity, we deployed Hikvision surveillance cameras (120° FOV, 30 fps) at optimized vantage points (Fig. 8). Artificial precipitation artifacts were systematically introduced into the captured video streams using physics-based rendering techniques, replicating authentic precipitation-interference conditions ranging from 25–50 mm/h rainfall equivalents.

Informed consent was obtained from all participants and/or their legal guardians for the publication of identifying information/images in an online open-access publication.

In the initial experimental phase, surveillance cameras installed across designated road segments collected continuous traffic data over a 48-h observation period. The acquired video streams underwent real-time processing through edge computing devices implementing the HRTNet detection algorithm. This configuration enabled instantaneous vehicle identification while concurrently logging spatiotemporal coordinates (temporal resolution: 100 ms; spatial accuracy: ± 1.5 m). The integrated system autonomously derived critical traffic parameters including congestion duration and affected roadway segments through continuous analysis of vehicular movement patterns.

Experimental validation under plateau environmental conditions demonstrated the de-rain algorithm's operational feasibility. Throughout the monitoring interval, the system maintained stable detection performance with minimal frame loss (< 2%). Quantitative analysis revealed distinct congestion patterns: during peak intervals (10:00–12:00 and 17:00–19:00 local time), Road A exhibited prolonged congestion durations (70 ± 5 min) compared to Roads B (25 ± 3 min) and C (40 ± 4 min). Spatial analysis showed Road A's congestion extended 0.7 km (95% CI 0.65–0.75 km), contrasting with 1.0 km (0.92–1.08 km) on Road B and 0.8 km (0.76–0.84 km) on Road C.

Statistical validation used paired t-tests ($\alpha = 0.05$) on three-road congestion data. Road A's duration (70 ± 5 min) significantly exceeded Road B (25 ± 3 min; $t = 12.5$, $p < 0.001$) and Road C (40 ± 4 min; $t = 9.8$, $p < 0.001$). Similarly, Road A's congestion extent (0.7 km) differed from Road B (1.0 km; $t = 4.2$, $p = 0.002$) and Road C (0.8 km; $t = 2.8$, $p = 0.008$). These $p < 0.01$ differences confirm Road A's higher congestion severity. For deployment, NVIDIA GeForce RTX 4090 edge devices support 10 cameras each. Larger networks require adding identical units—enabling modular expansion.

Conclusions and perspectives

Conclusion and discussion

We present HRTNet, an end-to-end real-time system for vehicle monitoring in extreme high-altitude rainfall. This framework pioneers a robust vision paradigm by integrating three innovations: hybrid encoder architecture, MEGA attention, and multi-scale deraining. These advances overcome fundamental limitations in adverse-condition vision: rain streak removal and real-time processing tradeoffs. Tests validate HRTNet's practical viability for plateau traffic, while sustaining real-time operation. Our architecture establishes a new standard for vision systems in monsoonal regions. The system supports traffic management decisions in challenging high-altitude environments, enabling responsive congestion mitigation. Beyond transportation, our architecture facilitates drone navigation and surveillance systems through its lightweight design, operational versatility, and deployment simplicity.

Limitations and research trajectories

Despite HRTNet's effectiveness for rainy-condition traffic detection on the Qinghai-Tibet Plateau, this work has limitations. Training relied primarily on summer data from specific road segments. Performance thus requires further validation under diverse lighting (e.g., nighttime/strong backlight) and extreme weather (e.g., snowstorms). Detection accuracy may decline during extreme congestion with closely spaced vehicles and severe occlusion. Small vehicle detection at long distances also needs improvement, as rain streaks degrade their features.

To address these limitations, future work will focus on several key areas. First, constructing more diverse traffic datasets covering plateau scenarios across seasons, times of day, and varied adverse weather conditions to better reflect real-world complexity. Second, developing end-to-end weather-robust image processing models beyond rainfall, integrated with advanced occlusion handling and adaptive feature fusion for improved detection in complex traffic. Third, exploring GNNs or spatiotemporal Transformers to analyze congestion propagation through road networks, extending the system from detection to prediction. These enhancements could substantially expand HRTNet's utility in intelligent transportation systems across challenging environments.

Data availability

The datasets generated and/or analysed during the current study are not publicly available due [legal restrictions, confidentiality agreements and privacy concerns] but are available from the corresponding author on reasonable request.

Received: 4 March 2025; Accepted: 24 July 2025
Published online: 04 August 2025

References

1. Wang, C. et al. Analysis of drivers' workload states on highways in high elevation regions. *Transp. Res. Rec.* **2678**, 1523–1544. https://doi.org/10.1177/03611981241252788 (2024).
2. Yu, B., Chen, Y. & Fu, Y. Driving speed prediction method for low grade highways from drivers' visual perception. *J. Tongji Univ. Nat. Sci.* **45**, 0362–0368. https://doi.org/10.3969/j.issn.1002-0268.2012.06.017 (2017).
3. Chen, S. M., Chen, W. & Yin, Z. Research status and prospect of single image rain removal algorithm. *Appl. Res. Comput.* **39**, 9–17. https://doi.org/10.19734/j.issn.1001-3695.2021.05.0209 (2022).
4. Yang, A. P. et al. Image deraining based on adaptive perceptual pyramid network. *J. Northeast. Univ. (Nat. Sci.)* **43**, 470–479. https://doi.org/10.12068/j.issn.1005-3026.2022.04.003 (2022).
5. Yuan, L. M. et al. Spatiotemporal characteristics of hydrothermal processes of the active layer on the central and northern Qinghai-Tibet plateau. *Sci. Total Environ.* **712**, 136392. https://doi.org/10.1016/j.scitotenv.2019.136392 (2020).
6. Li, H. et al. A new method of diagnosing the historical and projected changes in permafrost on the Tibetan plateau. *Earths Future* https://doi.org/10.1029/2023EF003897 (2024).
7. Huang, Y., Fang, J. & Zhang, Y. Operation and management of road safety applied technology in alpine high altitude area. *J. Highw. Trans. Res. Dev. (China)* **29**, 98–104. https://doi.org/10.3969/j.issn.1002-0268.2012.06.017 (2012).
8. Zhang, D. N. et al. Research on drivers' hazard perception in plateau environment based on visual characteristics. *Accid. Anal. Prev.* **166**, 106540. https://doi.org/10.1016/j.aap.2021.106540 (2022).
9. Qin, P. C. et al. Prediction of driving stress on high-altitude expressway using driving environment features: A naturalistic driving study in Tibet. *Traffic Inj. Prev.* **25**, 414–424. https://doi.org/10.1080/15389588.2024.2305420 (2024).
10. Tian, L., Li, J. S. & Li, Y. F. Analysis of driving fatigue characteristics in cold and hypoxia environment of high-altitude areas. *Big Data.* **11**, 255–267. https://doi.org/10.1089/big.2021.0464 (2023).

11. Luo, Y., Xu, Y. & Ji, H. Removing rain from a single image via discriminative sparse coding. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 3397–3405. https://doi.org/10.1109/ICCV.2015.388 (2015).
12. Li, Y. Tan, R. T., Guo, X. et al. Rain streak removal using layer priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2736–2744, https://doi.org/10.1109/CVPR.2016.299 (2016).
13. Li, X. Wu, J., Lin, Z. et al. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Computer Vision—ECCV 2018: 15th European Conference (ECCV)*, 254–269. https://doi.org/10.48550/arXiv.1807.05698 (2018).
14. Zamir, S. W., Arora, A., Khan, S. et al. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14821–14831, https://doi.org/10.1109/CVPR46437.2021.01458 (2021).
15. Gao, T. et al. Frequency-oriented efficient transformer for all-in-one weather-degraded image restoration. *IEEE Trans. Circuits Syst. Video Technol.* **34**, 1886–1899. https://doi.org/10.1109/TCSVT.2023.3299324 (2024).
16. Lin, G., Sheng, Z. & Liu, Y. Identification of traffic state based on cross-validation SVM. *J. Qingdao Univ. Sci. Technol. (Nat. Sci. Ed.)* **38**, 105–108 (2017).
17. Deng, C. et al. Research on rapid congestion identification method based on TSNE-FCM and LightGBM. *Sustainability.* **15**, 11322. https://doi.org/10.3390/su151411322 (2023).
18. 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 (CVPR)*, 779–788. https://doi.org/10.48550/arXiv.1506.02640 (2016).
19. Redmon, J. & Farhadi, A., YOLO9000: Better, faster, stronger. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7263–7271. https://doi.org/10.1109/CVPR.2017.690 (2017).
20. Redmon, J. YOLOv3: An incremental improvement. arXiv e-prints https://doi.org/10.48550/arXiv.1804.02767 (2018).
21. Bochkovskiy, A., Wang, C. Y. & Liao, H. Y. M. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint https://doi.org/10.48550/arXiv.2004.10934 (2020).
22. Jocher, G., Chaurasia, A., Stoken, A. et al. ultralytics/yolov5: v6.0—YOLOv5n "Nano" models, Roboflow integration, TensorFlow export, OpenCV DNN support. Zenodo. https://doi.org/10.5281/zenodo.5563715 (2022).
23. Li, C., Li, L., Jiang, H. et al. YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint https://doi.org/10.48550/arXiv.2209.02976 (2022).
24. Wang, C. Y., Bochkovskiy, A. & Liao, H. Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7464–7475. https://doi.org/10.1109/CVPR52729.2023.00721 (2023).
25. Jocher, G., Chaurasia, A., Qiu, J. Ultralytics YOLO, Jan. 2023. 1, 2, 6, 7, https://docs.ultralytics.com/ (2023).
26. Wang, C. Y., Yeh, I. H. & Liao, H. Y. M. Yolov9: Learning what you want to learn using programmable gradient information. In *Computer Vision—ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXXI.* https://doi.org/10.1007/978-3-031-72751-1_1 (2024).
27. Wang, A., Chen, H., Liu, L. et al. Yolov10: Real-time end-to-end object detection. arXiv preprint https://doi.org/10.48550/arXiv.2405.14458 (2024).
28. Vaswani, A., Shazeer, N., Parmar N. et al. Attention is all you need. In *Advances in Neural Inf. Process. Syst 30 (NeurIPS 2017)*, 6000–6010. https://doi.org/10.48550/arXiv.1706.03762 (2017).
29. Devlin, J., Chang, M., Lee, K., & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint https://doi.org/10.48550/arXiv.1810.04805 (2018).
30. Liu, Y., Ott, M., Goyal, N. et al. Roberta: A robustly optimized BERT pretraining approach. arXiv preprint https://doi.org/10.48550/arXiv.1907.11692 (2019).
31. Clark, K., Luong, M, T., Le, Q, V. et al. Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint https://doi.org/10.48550/arXiv.2003.10555 (2020).
32. Sun, Y., Wang, S., Li, Y. et al. Ernie: Enhanced representation through knowledge integration. arXiv preprint https://doi.org/10.48550/arXiv.1904.09223 (2019).
33. Carion, N., Francisco, M., Gabriel, S. et al. End-to-end object detection with transformers. In *Computer Vision ECCV 2020: 16th European Conference (ECCV)*. 213–229, https://doi.org/10.1007/978-3-030-58452-8_13 (2020).
34. Zhu, X., Su, W., Lu, L. et al. Deformable DETR: Deformable transformers for end-to-end object detection. arXiv preprint https://doi.org/10.48550/arXiv.2010.04159 (2020).
35. Li, F. et al. DN-DERT: Accelerate DERT training by introducing query denoising. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**, 2239–2251. https://doi.org/10.1109/TPAMI.2023.3335410 (2024).
36. Chen, Q., Chen, X., Zeng, G. et al. Group DERT: Fast training convergence with decoupled one-to-many label assignment. arXiv preprint https://doi.org/10.48550/arXiv.2207.13085 (2022).
37. Zhao, C., Sun, Y., Wang, W. et al. MS-DETR: Efficient DERT training with mixed supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 17027–17036. https://doi.org/10.1109/CVPR52733.2024.01611 (2024).
38. Zhao, Y., Lv, W., Xu, S. et al. DETRs Beat YOLOs on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16965–16974. https://doi.org/10.1109/CVPR52733.2024.01605 (2024).
39. Wang, S., Xia, C., Lv, F. et al. RT-DETRv3: Real-time end-to-end object detection with hierarchical dense positive supervision. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 1628–1636. https://doi.org/10.1109/WACV61041.2025.00166 (2024).
40. Wang, S., Zhao, Y., Chang, Q. et al. RT-DETRv2: Improved baseline with bag-of-freebies for real-time detection transformer. arXiv preprint https://doi.org/10.48550/arXiv.2407.17140 (2024).
41. Chen, C. F. R., Fan, Q. & Panda, R. Crossvit: Cross-attention multi-scale vision transformer for image classification. *In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 357–366. https://doi.org/10.1109/ICCV48922.2021.00041 (2021).
42. Dosovitskiy, A., Beyer, L., Kolesnikov, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint https://doi.org/10.48550/arXiv.2010.11929 (2020).
43. Liu, Z., Lin, Y., Cao, Y. et al. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10012–10022. https://doi.org/10.1109/ICCV48922.2021.00986 (2021).
44. Zhang, Z. et al. Dendritic learning-incorporated vision transformer for image recognition. *In IEEE/CAA J. Autom. Sin.* **11**, 539–541. https://doi.org/10.1109/JAS.2023.123978 (2024).
45. Liu, S., Li, F., Zhang, H. et al. Dab-DETR: Dynamic anchor boxes are better queries for DETR. arXiv preprint https://doi.org/10.48550/arXiv.2201.12329 (2022).
46. Chen, L., Chu, X., Zhang, X. et al. Simple baselines for image restoration. In *Computer Vision—ECCV 2022*, **13667,** 17–33. https://doi.org/10.1007/978-3-031-20071-7_2 (2022).
47. Liu, X., Peng, H., Zheng, N. et al. EfficientViT: Memory efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14420–14430. https://doi.org/10.1109/CVPR52729.2023.01386 (2023).
48. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 1251–1258. https://doi.org/10.1109/CVPR.2017.195 (2017).

49. Zhang, X., Zhou, X., Lin, M. et al. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6848–6856. https://doi.org/10.1109/CVPR.2018.00716 (2018).
50. Shaker, A., Maaz, M., Rasheed, H. et al. SwiftFormer: Efficient additive attention for transformer-based real-time mobile vision applications. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 17425–17436. https://doi.org/10.1109/ICCV51070.2023.01598 (2023).
51. Bewley, A., Ge, Z., Ott, L. et al. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, 3464–3468. https://doi.org/10.1109/ICIP.2016.7533003 (2016).
52. Lin, T. Y., Maire, M., Belongie, S. et al. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference (ECCV)*, 740–755. https://doi.org/10.1007/978-3-319-10602-1_48 (2014).
53. Han, H. et al. Raindrop size distribution measurements at high altitudes in the Northeastern Tibetan Plateau during summer. *Adv. Atmos. Sci.* **40**, 1244–1256. https://doi.org/10.1007/s00376-022-2186-z (2023).
54. Paszke, A., Gross, S., Massa, F. et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 33(NeurIPS 2019)*, **32,** 8026–8037. https://doi.org/10.48550/arXiv.1912.01703 (2019).
55. Wightman, R. Pytorch image models. https://zenodo.org/badge/latestdoi/168799526 (2019).
56. Loshchilov, I. Decoupled weight decay regularization. arXiv abs. **1711,** 05101, https://doi.org/10.48550/arXiv.1711.05101 (2017).
57. Wang, C. C., He, W., Nie,Y. et al. Gold-yolo: Efficient object detector via gather-and-distribute mechanism. In *Advances in Neural Information Processing Systems 37(NeurIPS 2023)*, **2224,** 51094–51112. https://doi.org/10.48550/arXiv.2309.11331 (2023).
58. Yuming, C. et al. YOLO-MS: rethinking multi-scale representation learning for real-time object detection. *In IEEE Trans. Pattern Anal. Mach. Intell.* **47**, 4240–4252. https://doi.org/10.1109/TPAMI.2025.3538473 (2023).
59. Yingming, W., Xianyu, Z., Tong, Y. et al. Anchor DETR: Query design for transformer-based detector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, **36**, 2567–2575. https://doi.org/10.1609/aaai.v36i3.20158 (2022).
60. Peng, G., Minghang, Z., Xiaogang, W. et al. Fast convergence of detr with spatially modulated co-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3601–3610, https://doi.org/10.1109/ICCV48922.2021.00360 (2021).
61. Howard, A.G., Zhu, M., Chen, B. et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv https://doi.org/10.48550/arXiv.1704.04861(2017).
62. Dosovitskiy, A., Beyer, L., Kolesnikov, A. et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv https://doi.org/10.48550/arXiv.2010.11929 (2020).
63. Ze, L., Yutong, L., Yue, Cao. et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. https://doi.org/10.48550/arXiv.2103.14030 (2021).

Acknowledgements

This work was supported by National Key Research and Development Program of China (2022YFC3002602).

Author contributions

Q.T. identified the research problem and proposed the overall solution; Y.Z. designed the methodology (architecture), conducted the experimental design, performed data analysis, and wrote the manuscript; D.H. and S.S. designed the rain removal network architecture; Z.Z. and X.D. reviewed and revised the manuscript for grammar and clarity; J.W. was responsible for collecting the dataset on the Qinghai-Tibet Plateau.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Q.T.