vehicles

MDPI

Article

A Vehicle Type Recognition Network Based on Feature Comparison and Mixture of Experts Model

Taotao Hu [1], Xiufeng Zhao [1] and Luxia Yang [1,2,*]

[1] School of Computer Science and Technology, Taiyuan Normal University, Jinzhong 030619, China
[2] Shanxi Provincial Key Laboratory of Intelligent Optimization Computing and Blockchain Technology, Jinzhong 030619, China
* Correspondence: luxiayang@tynu.edu.cn

Abstract

To address the challenges of insufficient feature fusion and incomplete multi-scale information capture in complex traffic scenarios, we propose a vehicle type recognition network based on feature comparison and the Mixture of Experts (MoE) model. Specifically, the MobileNetV4 backbone is introduced to enhance deep feature extraction for vehicle targets. Meanwhile, we design a Multi-scale Interleaving Fusion Module (MSIFM), which progressively transmits feature channels via an interleaving structure to capture multi-scale features while enhancing vehicle feature representation. Moreover, we devise a Feature Compare Enhancement Module (FCEM) to efficiently fuse feature maps with different semantic information. By performing feature comparison, it strengthens strongly correlated features while suppressing weakly correlated ones. Finally, we design a Mixture of Experts Feature Enhancement Module (MOEFEM) to aggregate multi-scale feature maps and adaptively capture detailed vehicle features through multiple expert units. Experimental results demonstrate that our method achieves mAP improvements of 2.2% and 2.4% over YOLOv11 on UA-DETRAC and BDD100K, respectively. The proposed method not only improves detection accuracy significantly but also maintains real-time efficiency, providing a practical solution for high-precision vehicle type recognition. It offers valuable technical support for intelligent transportation systems, smart city management, and autonomous driving safety.

Keywords: vehicle type recognition; multi-scale; feature fusion; feature comparison; mixture of experts model

1. Introduction

Vehicle type recognition [1] is a crucial component of intelligent transportation systems [2], with broad applications in traffic control, traffic flow statistics [3], and traffic scheduling. In 2026, with the large-scale deployment of vehicle–road collaboration, smart cities, and autonomous driving technologies, high-precision vehicle type recognition has become an indispensable part of modern traffic systems. On the one hand, it supports refined smart city management, including real-time traffic monitoring, adaptive signal timing, congestion alleviation, and illegal vehicle detection, thereby greatly improving road utilization and management efficiency. On the other hand, it provides reliable environmental perception for autonomous driving, effectively reducing perception errors under complex conditions and enhancing driving safety. Therefore, developing efficient and accurate vehicle type recognition methods is of great practical value for both intelligent

Check for updates

Received: 29 March 2026
Revised: 21 April 2026
Accepted: 29 April 2026
Published: 3 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

transportation and autonomous driving. Currently, vehicle type recognition algorithms can be divided into two categories: traditional methods and deep learning-based methods.

Traditional vehicle type recognition methods primarily rely on low-cost small sensors, LiDAR, manual feature descriptors, and machine learning algorithms. For instance, feature descriptors such as SIFT [4] and HOG [5] are widely utilized. Odat et al. [6] utilized geomagnetic sensors to capture the external contour features of vehicles for type classification. Putra et al. [7] employed Gaussian mixture models to extract the background images of vehicles and then classified pixel points to recognize vehicle types. Li et al. [8] matched data from multiple sensors to form a fused feature waveform for each vehicle and output the vehicle type recognition results.

With the increase in the number of vehicles, traditional methods struggle to achieve satisfactory speed and accuracy when faced with large vehicle datasets. The rapid development of artificial intelligence and hardware facilities such as GPUs has enabled parallel processing of computational data, enhancing the accuracy of vehicle type recognition. Consequently, deep learning-based vehicle type recognition methods have emerged. In real-world applications, vehicle recognition faces great challenges because traffic flow is highly random and dynamic. Traffic volume varies significantly at different times of day, on different days of the week, and in different months of the year. Traffic flow also differs considerably between working days, weekends, and holidays [9]. These temporal variations further increase the difficulty of stable and accurate vehicle recognition.

Deep learning-based vehicle type recognition algorithms are generally categorized into single-stage and two-stage methods. For two-stage algorithms, the RCNN series [10–12] employs selective search algorithms to generate candidate regions and then extracts features from each candidate region to produce recognition results. For instance, Ke et al. [13] proposed a data balancing strategy based on Faster-RCNN to enhance vehicle type recognition performance.

Although two-stage algorithms offer excellent accuracy, they struggle to meet real-time requirements on devices with limited computational resources. In contrast, single-stage algorithms such as the YOLO [14] series, SSD [15], and FCOS [16] directly extract features from the input image and output recognition results. This direct processing mechanism significantly improves their recognition speed. Meanwhile, they are gradually surpassing two-stage algorithms in terms of accuracy, making them more suitable for vehicle type recognition scenarios. For example, Song et al. [17] added Mamba modules to the backbone network, significantly reducing computational consumption, but the recognition capability also decreased. Kasper et al. [18] used YOLOv5 and thermal network cameras for heavy truck recognition, successfully identifying heavy trucks in winter rest areas and allowing real-time prediction of parking space occupancy rates. However, this model fails to effectively detect heavy trucks obscured by other vehicles, resulting in a low detection rate for such occluded targets. Sun et al. [19] utilized depthwise separable convolutions to reduce backbone network parameters and employed SENet to improve vehicle type recognition accuracy. Cao et al. [20] optimized the loss function and introduced weight regularization to develop a model for vehicle type recognition. This model also enabled the systematic implementation of traffic flow statistics.

Although the aforementioned methods hold significant importance for vehicle type recognition, they still exhibit certain limitations. First, different weather conditions easily lead to unclear edge contours of vehicle targets, which interferes with feature extraction in various directions. Second, although most methods improve the accuracy of vehicle type recognition, their processing of feature fusion is incomplete. They ignore the feature correlation between different hierarchical levels [19,20], and the mining of deep semantic information is insufficient. Finally, vehicles are dense in real traffic scenes, and vehicle

overlap is prone to occur. The effective extraction of multi-scale features also restricts the recognition accuracy of targets [18]. Therefore, to address the issues of insufficient feature fusion and incomplete multi-scale information extraction in vehicle type recognition tasks, this paper proposes a vehicle type recognition network based on feature comparison and Mixture of Experts (MoE). First, we propose a Multi-scale Interleaving Fusion Module that utilizes multi-branch channels and interleaving transmission structures to capture multi-scale features. Second, we design a Feature Compare Enhancement Module to effectively fuse feature maps of different scales and distinguish feature intensity, enhancing feature expression capability. Finally, we construct a Mixture of Experts Feature Enhancement Module to capture specific details of vehicle features and obtain precise localization effects.

The main contributions of this paper are summarized as follows:

- We propose a novel vehicle type recognition framework integrating feature comparison and the Mixture of Experts mechanism. The proposed framework overcomes the limitations of existing methods in feature fusion and target localization. It systematically integrates multi-scale feature extraction, dynamic feature enhancement, and adaptive expert selection mechanisms. This work provides a new technical pathway for high-precision, real-time vehicle type recognition in complex traffic scenarios.
- We propose a Multi-scale Interleaving Fusion Module (MSIFM). By utilizing channel partitioning and interleaving transmission mechanisms, it effectively captures multi-scale features while reducing computational complexity. In this way, it solves the problem of insufficient multi-scale information fusion in existing methods.
- We design a Feature Compare Enhancement Module (FCEM). This module introduces a discrimination mechanism for strongly and weakly correlated features. As a result, it can dynamically strengthen key features, which effectively alleviates the low information utilization in traditional fusion strategies such as simple concatenation or element-wise addition.
- We construct a Mixture of Experts Feature Enhancement Module (MOEFEM). For the first time, the Mixture of Experts model is introduced into the vehicle type recognition task. Multiple expert units are leveraged to adaptively extract key detail features, significantly improving the localization capability for vehicle targets.

The rest of this paper is organized as follows. Section 2 reviews related work on vehicle type recognition and Mixture of Experts (MoE) models. Section 3 elaborates the overall framework and detailed designs of the proposed modules. Section 4 presents experimental settings, ablation studies, comparative results, and visualization analysis. Section 5 concludes the whole work and discusses future directions.

2. Related Work

2.1. Vehicle Type Recognition

Vehicle type recognition is a foundational technology in intelligent transportation systems. With the advancement of deep learning, numerous scholars have conducted research in this field.

During the early stages of research, vehicle type recognition primarily relied on traditional methods. These methods typically involved manually extracting vehicle features from video sequences and then classifying and identifying the extracted features. Jheng et al. [21] proposed utilizing the symmetric shadow characteristics of vehicles for type recognition. Fung et al. [22] extracted curvature and length information from vehicle movement trajectories, as well as grayscale and edge features from vehicle images, to approximate and model vehicle shapes for type identification. Lim et al. [23] proposed a Gabor filtering method to extract vehicle features, combining it with Support Vector Machines (SVMs) for vehicle type classification and recognition. Wen et al. [24] combined sample

features with label features to propose a fast feature selection method based on Adaboost. An improved normalization algorithm was also designed to process the selected feature values, thereby reducing intra-class variance and increasing inter-class variance. Hsieh et al. [25] first divided vehicles into multiple grids, then extracted SURF features from the grid images, and combined different weak classifiers trained on these grids. Through this approach, they constructed a strong classifier to achieve accurate vehicle type recognition.

With the widespread application of deep learning, feature extraction methods for vehicle images have undergone significant changes. Unlike the early stages, deep learning-based methods do not require manual feature selection; they can autonomously extract features and learn from them [26,27]. Vehicle type recognition algorithms are generally divided into one-stage and two-stage algorithms. Compared to two-stage algorithms, one-stage algorithms achieve a better balance between accuracy and speed, making them more widely applied. YOLO series detectors have been widely used in real-time vehicle detection due to their excellent balance between speed and accuracy. YOLOv5, YOLOv8, YOLOv10, and YOLOv11 represent typical and advanced one-stage detectors in recent years, and thus are selected as the main baseline methods in this paper for fair comparison. Kang et al. [28] proposed a novel YOLO detector based on fuzzy attention, enhancing vehicle type recognition under rainy and nighttime conditions. Bie et al. [29] designed a lightweight detection network based on YOLOv5, increasing recognition speed. Shi et al. [30] fully integrated temporal information and spatial features, leveraging the complementary nature of feature information from historical frames and the current frame. This approach achieved excellent performance on the nuScenes dataset for battery electric vehicle (BEV) object detection.

2.2. Application of Mixture of Experts

The application of Mixture of Experts (MoE) in the image domain has achieved significant progress in recent years, particularly in Vision Transformer architectures, multimodal learning, and efficient computing. In 2021, Riquelme et al. [31] proposed V-MoE, a pioneering work for vision MoE. It integrates sparse MoE layers into Vision Transformer (ViT). This model dynamically assigns image patches (tokens) to different expert networks through a learnable routing mechanism. It reduces inference computation by approximately 50% while maintaining comparable performance to dense models. ViMoE [32] presents an empirical study of MoE fine-tuning on the DINOv2 model. It explores expert placement strategies and introduces a Shared Expert mechanism to capture general knowledge and improve convergence stability. Soft MoE [33] proposes a soft MoE mechanism. It achieves better performance and training speed than traditional MoE on billion-scale datasets. M2Restore [34] proposes an MoE-based Mamba-CNN fusion framework for all-in-one image restoration. Inspired by the above insights, we introduce the Mixture of Experts (MoE) model into the model architecture, and design dedicated expert units to accurately extract key features. These features provide high-quality representations for vehicle type recognition.

3. Methodology

The proposed model consists of an encoder, a decoder, and detection heads. The overall architecture is illustrated in Figure 1. The proposed framework is developed based on the YOLOv8 detection pipeline. We choose YOLOv5, YOLOv8, YOLOv10, and YOLOv11 as representative benchmarks to verify the advancement of our method.

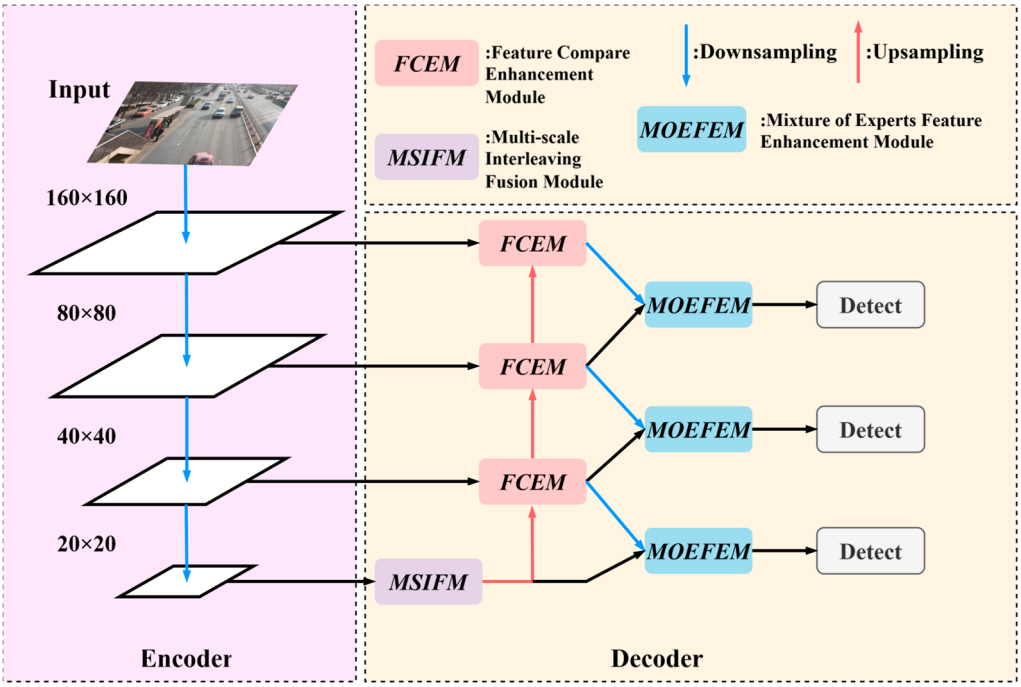


Figure 1. A vehicle type recognition network based on feature comparison and Mixture of Experts model.

In the encoder stage, we utilize the medium version of MobileNetV4 [35], which strikes a favorable balance between feature extraction capability and computational efficiency. From this backbone, we extract four multi-scale feature maps (160×160, 80×80, 40×40, and 20×20) as inputs to the decoder. These features provide a rich source of semantic information to guide the subsequent decoding process.

In the decoder stage, we first apply the Feature Compare Enhancement Module (FCEM) to fuse multi-scale features, mine deep semantic information, and restore image resolution. Then, the MOEFEM aggregates features from different hierarchical levels. It highlights vehicle targets through multi-level residual connections and feeds the features into the detection heads. Finally, the detection heads of YOLOv8 are utilized to output the recognition results.

3.1. Multi-Scale Interleaving Fusion Module

In real-world traffic environments, vehicle type recognition faces numerous challenges. Target vehicles appear at varying distances from cameras, and vehicles differ in inherent size. Consequently, vehicles of the same or different categories exhibit significantly different scales and appearance features. This diversity increases the difficulty of multi-scale feature extraction and fusion, adversely affecting recognition performance.

To address this issue, we design a Multi-scale Interleaving Fusion Module (MSIFM), as illustrated in Figure 2. Unlike traditional SPPF or FPN structures, this module achieves cross-scale information transmission through a channel interleaving mechanism. This design avoids feature redundancy and computational stacking.

First, the input feature map is split equally into four branches along the channel dimension to reduce computational complexity. Meanwhile, a gradient structure is employed to propagate the feature flow, ensuring that each branch contains sufficient information. This process is formulated as follows:

$$x_1, x_2, x_3, x_4 = Spilt(X) \tag{1}$$

$$\begin{aligned} x_2 &= x_2 + x_1 \\ x_3 &= x_3 + x_2 \\ x_4 &= x_4 + x_3 \end{aligned} \tag{2}$$

where $Spilt$ denotes the channel partitioning operation, and X represents the initial input feature.

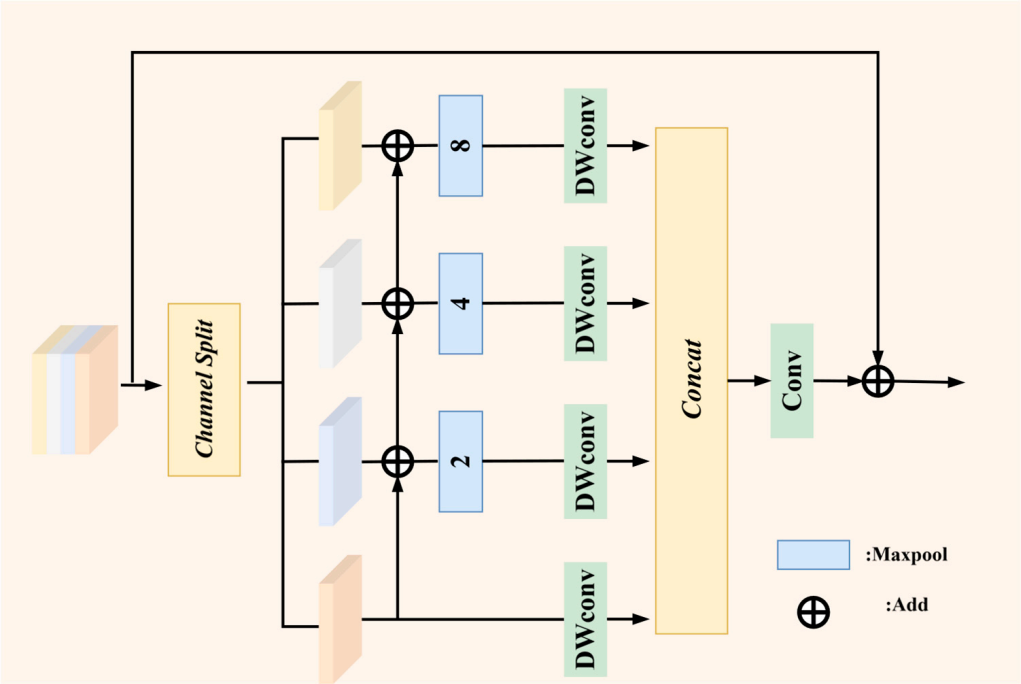


Figure 2. Structure of Multi-scale Interleaving Fusion Module.

Second, to effectively identify and select discriminative features that capture non-local interactions, adaptive max pooling is applied to the input features on three branches to generate multi-scale features. Depthwise separable convolution is then employed on each branch to further capture refined vehicle features. This process can be expressed as follows:

$$\begin{aligned} x_1 &= f_{DW}(x_1) \\ x_2 &= f_{DW}(f_{Maxpool}(x_2)) \\ x_3 &= f_{DW}(f_{Maxpool}(x_3)) \\ x_4 &= f_{DW}(f_{Maxpool}(x_4)) \end{aligned} \tag{3}$$

where f_{DW} denotes the depthwise separable convolution operation, and $f_{Maxpool}$ represents the max pooling operation.

Finally, the four branches are concatenated along the channel dimension to enhance feature representation across channels. The result is added to the input feature to strengthen the output and improve feature transmission efficiency. This enables the network to better learn complex feature representations, as shown in Equations (4) and (5):

$$K = f_{conv} f_{Concat}(x_1, x_2, x_3, x_4) \tag{4}$$

$$Z = X + K \tag{5}$$

where f_{Concat} denotes channel concatenation, f_{Conv} represents the 1×1 convolution operation for channel adjustment, and Z indicates the final output.

3.2. Feature Compare Enhancement Module

Typically, after obtaining the initial feature maps, deep semantic features are propagated to shallow layers and fused with them. This process enables the network to construct rich and expressive feature representations. However, most existing object detection algorithms merely concatenate or add features during fusion without further refinement. This shallow fusion strategy produces weakly correlated features, thereby hindering recognition accuracy.

To enhance feature fusion effectiveness, we propose a Feature Compare Enhancement Module (FCEM), as shown in Figure 3. Unlike traditional attention mechanisms such as CBAM and SE, our module adopts a feature comparison strategy that can clearly distinguish between strong and weak features, thereby achieving more precise feature enhancement.

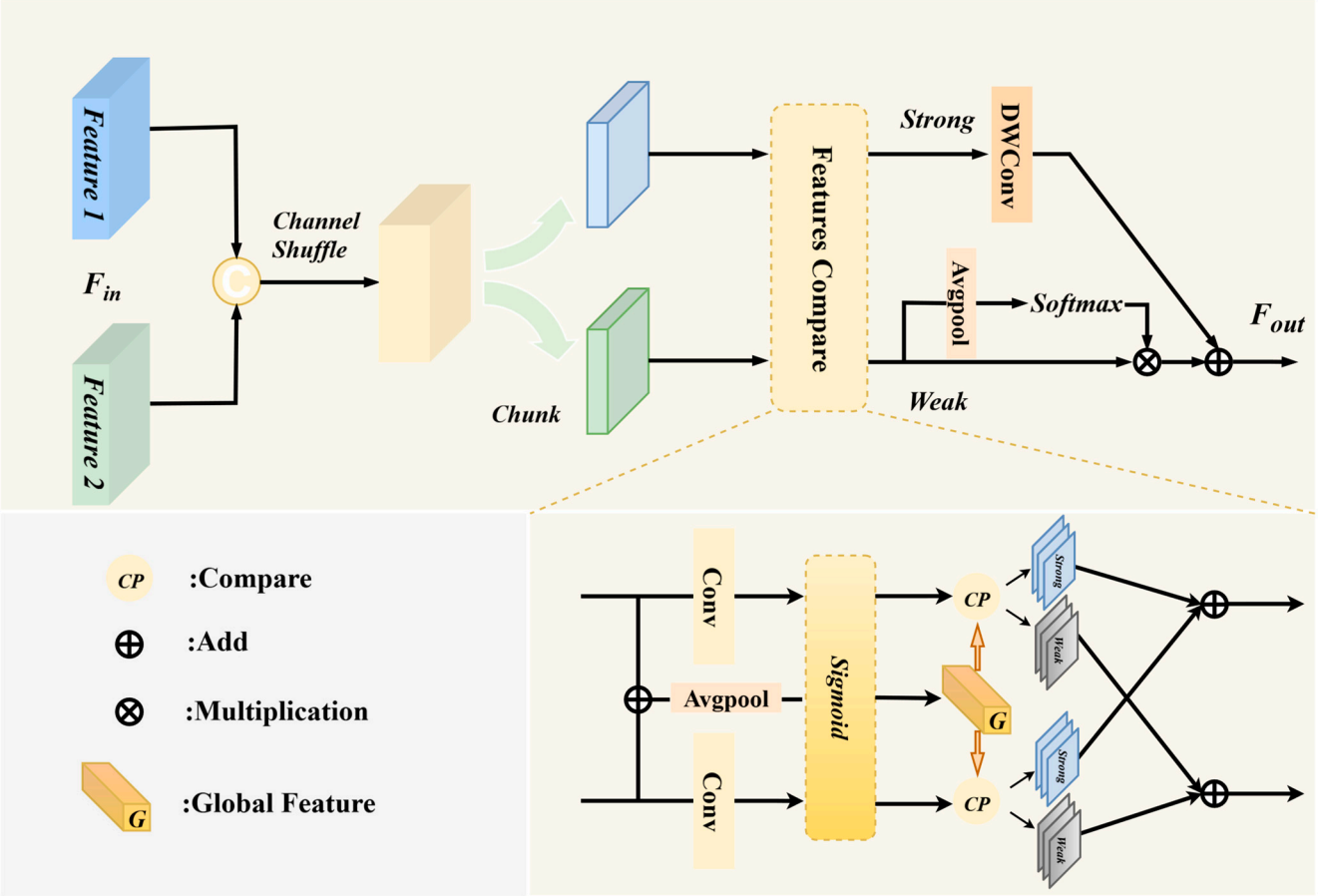


Figure 3. Structure of Feature Compare Enhancement Module.

First, features from the previous layer and encoder features are concatenated at the channel level, enriching the channel information. The concatenated features then undergo channel shuffling. Different channels can thus interact and fuse more thoroughly, breaking down information isolation for more effective utilization of multi-channel features. This process can be expressed as follows:

$$Y = f_{Suffle}(f_{concat}(x_1, x_2)) \tag{6}$$

where f_{Suffle} denotes the channel shuffling operation.

Second, the feature map with rich multi-channel features is evenly divided along the channel dimension and fed into the Feature Compare module. By doing so, the two branch features are combined to form global features. Subsequently, global average pooling and

sigmoid function are applied to generate threshold weights. This process can be expressed as follows:

$$y_1, y_2 = f_{Chunk}(Y) \tag{7}$$

$$T = f_{sigmoid}(f_{Avgpool}(y_1 + y_2)) \tag{8}$$

where $f_{sigmoid}$ denotes the *sigmoid* operation, y_1 and y_2 represent the evenly divided sub-tensors, and T denotes the threshold weights.

Then, 3×3 convolution is applied to both branches to obtain local features, followed by sigmoid function to acquire branch-specific weights. These weights are then compared with the threshold weights. Specifically, features exceeding the threshold are classified as strongly correlated, while those below are weakly correlated, yielding two distinct feature sets. These feature sets demonstrate the model's dynamic capability to identify and process key features, thereby improving the discrimination and effectiveness of vehicle feature expression. This process can be expressed as follows:

$$\begin{aligned} y_1 &= f_{sigmoid}(f_{conv}(y_1)) \\ y_2 &= f_{sigmoid}(f_{conv}(y_2)) \end{aligned} \tag{9}$$

$$\begin{aligned} y_1^{Strong} &= y_1 \geq T \\ y_1^{Weak} &= y_1 < T \\ y_2^{Strong} &= y_2 \geq T \\ y_2^{Weak} &= y_2 < T \end{aligned} \tag{10}$$

$$\begin{aligned} y^{Strong} &= y_1^{Strong} + y_2^{Strong} \\ y^{Weak} &= y_1^{Weak} + y_2^{Weak} \end{aligned} \tag{11}$$

where y^{Strong} denotes strongly correlated features, and y^{Weak} denotes weakly correlated features.

Finally, for strongly correlated features, depthwise convolution is employed to further capture local features of each channel while reducing redundant computation. For weakly correlated features, a self-gating mechanism is utilized to dynamically adjust input features, helping the model select more relevant vehicle target information. Both types of features are then added to obtain enhanced feature representations. This process can be expressed as follows:

$$y^{Strong} = f_{DW}(y^{Strong}) \tag{12}$$

$$y^{Weak} = y^{Weak} \times f_{Sm}(f_{Avgpool}(y^{Weak})) \tag{13}$$

$$Z = y^{Strong} + y^{Weak} \tag{14}$$

where f_{Sm} denotes the $SoftMax$ operation.

3.3. Mixture of Experts Feature Enhancement Module

During network execution, shallow feature maps contain more fine-grained information, which makes them suitable for detecting smaller objects. In contrast, deeper layers encompass richer global context and higher-level semantic information, which are better suited for processing large targets. Therefore, effectively fusing multi-level information is crucial for accurately identifying vehicles of varying sizes.

Currently, mainstream YOLO series models typically adopt direct concatenation for multi-scale feature fusion to improve computational efficiency. This approach ignores the differences in importance among features at different levels and cannot adaptively adjust according to the scale, pose, and occlusion status of vehicle targets, resulting in limited feature utilization efficiency. In contrast, the proposed Mixture of Experts Feature Enhancement Module (MOEFEM) introduces a dynamic gating mechanism to learn adaptive

weights and select the optimal expert units automatically. Different experts are specialized in capturing edge information, contour structure, and multi-scale details. This dynamic design effectively overcomes the limitations of fixed feature concatenation, enabling the network to focus on key vehicle regions and greatly improving feature representation and localization accuracy.

The structure of the Mixture of Experts Feature Enhancement Module (MOEFEM) is shown in Figure 4. We apply the MOEFEM to three feature maps of different scales. Each module receives features from two adjacent hierarchical levels and captures the most critical features and select optimal feature paths, enhancing the generalization capability for complex vehicle types.

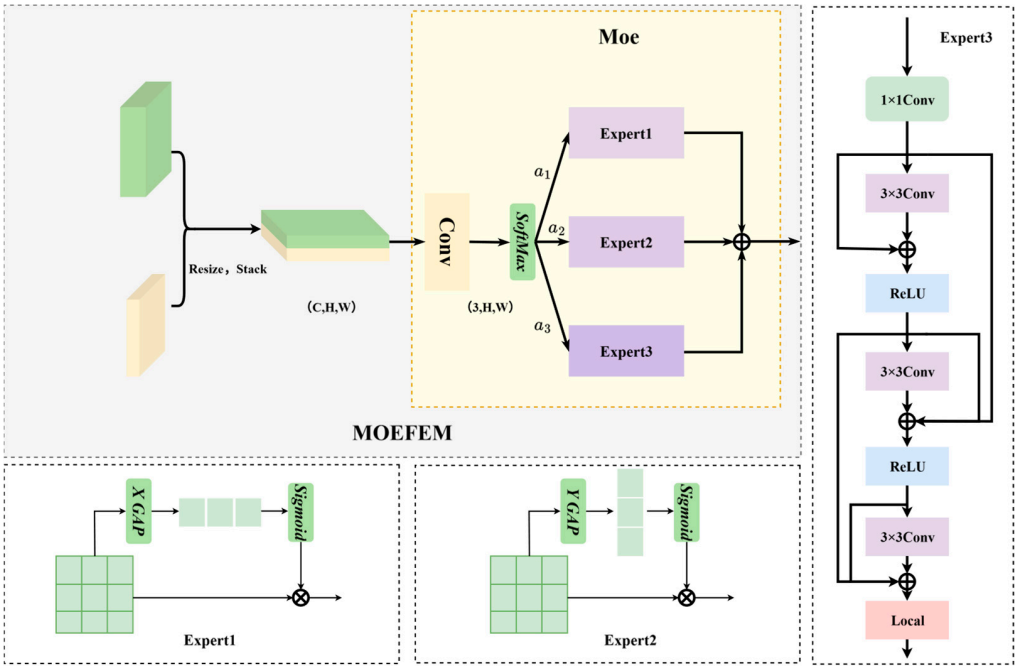


Figure 4. Mixture of Experts Feature Enhancement Module.

First, the MOEFEM upsamples low-resolution feature maps and concatenates them with high-resolution feature maps. Through this operation, features from different hierarchical levels are fused more effectively, enhancing recognition capability for multi-scale vehicle targets.

Second, the fused features are fed into the Mixture of Experts (MoE) model. A 1×1 convolution is first employed to adjust the channel dimension of input features to 3, which is consistent with the number of expert units. Then, the Softmax activation function is applied along the channel dimension to generate three adaptive scalar weights, denoted as a_1, a_2, and a_3. These weights represent the relative importance of the three expert branches and are automatically learned in an end-to-end manner.

Subsequently, the input features are forwarded into three parallel expert units. Expert 1 and Expert 2 are constructed as vertical and horizontal attention units, respectively, which strengthen the capture of vehicle edge information along spatial dimensions. Expert 3 utilizes convolutions with diverse kernel sizes to extract multi-scale details, while residual connections and activation functions enhance feature propagation and alleviate gradient vanishing.

To fuse the expert outputs, the three scalar weights a_1, a_2, and a_3 are broadcasted along the channel, height, and width dimensions to match the complete spatial and channel dimensions of the expert output features. The broadcasted weights are then multiplied element-wise with the corresponding expert output features. Finally, the three weighted fea-

ture maps are summed element-wise to obtain the final enhanced feature. The computation is formulated as:

$$out = a_1 F_{\exp ert1} + a_2 F_{\exp ert2} + a_3 F_{\exp ert3} \quad (15)$$

where a_1, a_2, and a_3 denote the generated adjustment weights, and $F_{\exp ert1}$, $F_{\exp ert2}$, and $F_{\exp ert3}$ represent the output results of the three expert units.

Through the above weight broadcasting and weighted fusion mechanism, the model can adaptively emphasize valuable expert features and suppress trivial information, significantly improving the representation ability and localization accuracy of vehicle targets.

3.4. Datasets and Experimental Settings

3.4.1. Datasets

The datasets used in this paper are the large-scale vehicle datasets UA-DETRAC [36] and BDD100K [37] for traffic surveillance scenarios. The UA-DETRAC dataset is complex in terms of its scene content. It consists of four vehicle categories, with a total of over 140,000 images. The training set contains 82,085 images, while the testing set contains 56,167 images.

The UA-DETRAC dataset is constructed by extracting individual frames from captured video data to form an image dataset. It is divided into four categories based on weather conditions: cloudy, sunny, rainy, and nighttime. The UA-DETRAC dataset is shown in Figure 5.

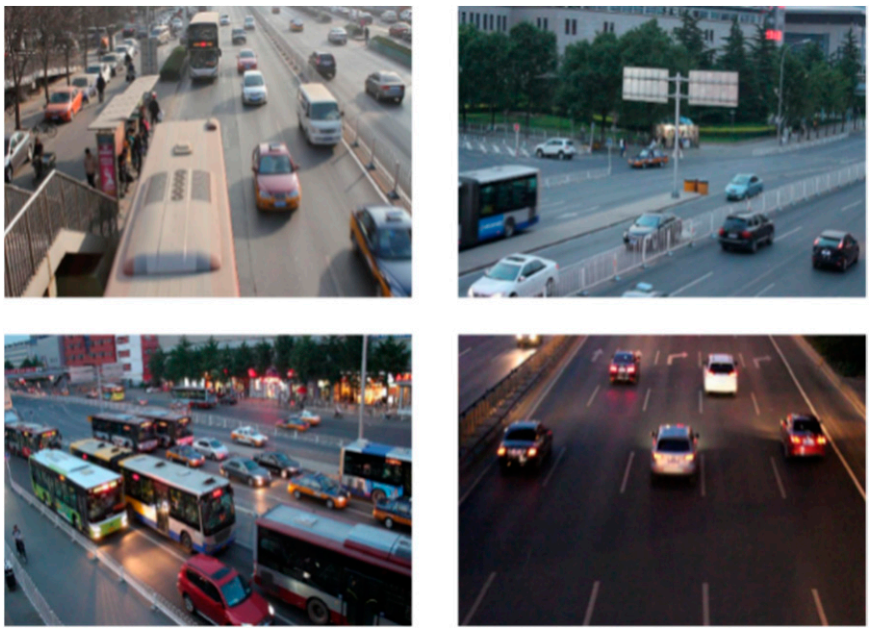


Figure 5. UA-DETRAC dataset.

The BDD100K dataset comprises ten object categories: Person, Rider, Car, Bus, Truck, Bike, Motor, Train, Traffic light, and Traffic sign. Since our research focuses on vehicle type recognition, we manually eliminated the labels for Person, Rider, Train, Traffic light, and Traffic sign. Only vehicle target labels were retained. The original BDD100K dataset consists of 70,000 images for training, 10,000 images for validation, and 20,000 images for testing. However, as the test set lacks annotations, these 20,000 unlabelled images are excluded to enable more accurate evaluation of the model. The remaining annotated images are then repartitioned into new training, validation, and test sets at a ratio of 7:1:2. The BDD100K dataset is shown in Figure 6.

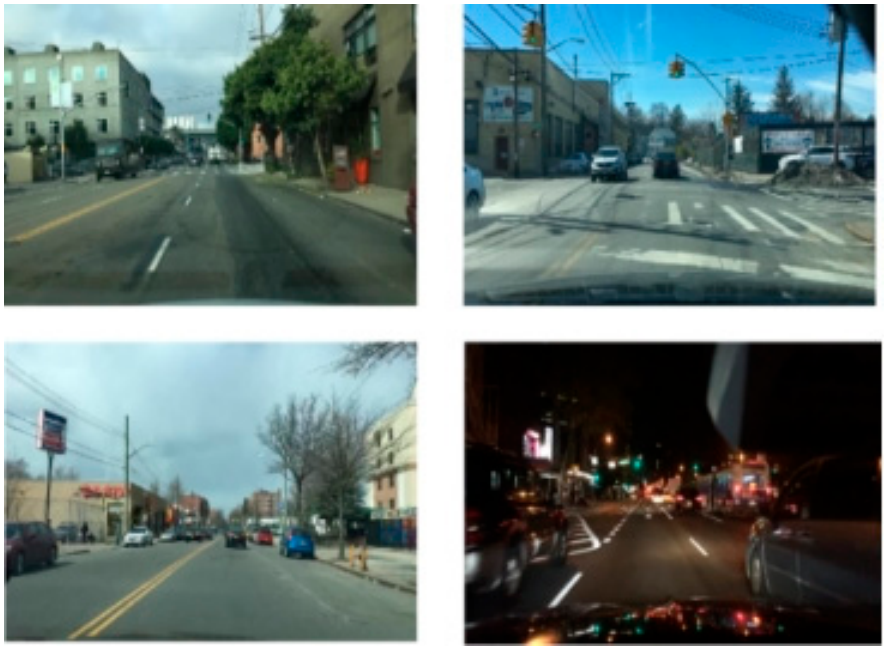


Figure 6. BDD100K dataset.

3.4.2. Training Configuration

The hardware environment of the experimental platform is shown in Table 1.

Table 1. Training Configuration Details.

CPU	GPU	Epochs	Batch Size
i9-14900k	RTX4090	200	16

The initial learning rate is set to 0.001. SGD is selected as the optimizer. CIoU loss is employed as the loss function, which is widely used in the object detection field.

4. Experiments

4.1. Evaluation Metrics

In practical applications, vehicle type recognition must meet the dual requirements of recognition accuracy and processing speed. This ensures efficient and precise identification. Accordingly, *GFLOPs*, *Parameters*, *FPS*, and *mAP* are selected as comparison metrics to demonstrate the application value and performance advantages of the proposed algorithm.

GFLOPs denote the number of floating-point operations executed by the model in a single forward pass. The unit is billions of floating-point operations (10^9 FLOPs).

Parameters indicate the number of parameters during the model training process. They reflect the complexity of the model and the extent of resource consumption.

FPS (Frames Per Second) represents the number of images processed per unit time. A higher relative *FPS* value indicates better processing speed.

mAP (mean Average Precision) is obtained by summing the *AP* values for each class and averaging them. The calculation is shown in Equation (16), where M is the number of recognized classes.

$$mAP = \frac{\sum_{m=1}^{M} AP(q)}{M} \tag{16}$$

4.2. Ablation Experiment

To verify the impact of each module on the overall detection performance, ablation studies are conducted on the UA-DETRAC and BDD100K datasets. The MSIFM, FCEM, and MOEFEM are replaced with 3×3 convolutions, using the combination of MobileNetV4 and 3×3 convolutions as the baseline. Specifically, each 3×3 convolution is a sequential stack of 3×3 convolution, BatchNorm, and SiLU activation, with a stride of 1. The channel numbers of the four hierarchical stages are set to 512, 256, 128, and 64 from bottom to top. The MSIFM, FCEM, and MOEFEM are then incrementally added. The experimental results are presented in Tables 2 and 3.

Table 2. Ablation experiment on UA-DETRAC dataset.

Baseline	MSIFM	FCEM	MOEFEM	mAP	GFLOPs	Params/10^6	FPS
√				55.4	15.8	7.9	111
√	√			57.5	15.4	8.2	117
√		√		61.2	16.8	8.8	108
√			√	59.1	16.4	8.7	105
√	√	√		62.4	16.7	9.1	114
√		√	√	62.8	18.4	9.4	102
√	√		√	63.4	18.3	9.2	105
√	√	√	√	63.9	19.7	9.8	109

Table 3. Ablation experiment on BDD100K dataset.

Baseline	MSIFM	FCEM	MOEFEM	mAP	GFLOPs	Params/10^6	FPS
√				54.4	15.8	7.9	111
√	√			55.9	15.4	8.2	115
√		√		60.5	16.8	8.8	108
√			√	59.3	16.4	8.7	105
√	√	√		60.3	16.7	9.1	114
√		√	√	61.5	18.4	9.4	102
√	√		√	60.8	18.3	9.2	105
√	√	√	√	62.6	19.7	9.8	107

As shown in Tables 2 and 3, all three modules contribute positive improvements to the model, with the FCEM achieving the most significant performance boost for the baseline. This verifies that efficient feature fusion plays a crucial role in enhancing vehicle type recognition performance. Notably, the MSIFM not only improves detection accuracy but also reduces computational complexity, which benefits from its effective capture of multi-scale features and the reduction in computation brought by channel-wise calculation. In addition, the design of the MOEFEM endows the model with a better localization capability for vehicle targets prior to final detection. Overall, the experimental results in Tables 2 and 3 demonstrate that the proposed vehicle type recognition algorithm achieves a significant improvement in recognition accuracy compared with the baseline model, and all the designed modules exert an effective role in the proposed algorithm.

4.3. Comparative Experiments

To demonstrate the vehicle type recognition performance of the proposed algorithm in traffic scenes with different complexities, comparative experiments are conducted on the UA-DETRAC dataset and the BDD100K dataset. General object detection algorithms such as YOLOv5s, RT-DETR-l, YOLOv8s, YOLOv10s, and YOLOv11s are selected. Recent state-of-the-art improved models are also included for comparison.

As shown in Table 4, the designed vehicle type recognition algorithm outperforms other comparative algorithms in recognition performance. Compared with state-of-the-art models including RT-DETR-L, YOLOv3, YOLOv5s, YOLOv8s, YOLOv10s and YOLOv11s, the proposed algorithm achieves accuracy improvements of 4.3%, 0.9%, 3.9%, 3.1%, 3.5% and 2.2%, respectively. It also yields accuracy gains of 2.4%, 1.6%, 5.1%, 2.7% and 2.1% in comparison with the studies of international scholars reported in [17,27,38–40], while demonstrating certain advantages in computational efficiency. As illustrated by the heatmap visualization results in Figure 7, compared with the state-of-the-art YOLOv11s detector, our proposed method demonstrates more precise and comprehensive attention to the complete edge contours of vehicle objects. In contrast, the YOLOv11 detector is susceptible to environmental disturbances, leading to inaccurate and unstable focus on vehicle regions. These improvements are attributed to the superior vehicle center-localization capability of the MOEFEM, as well as the ability of the FCEM to enhance strongly correlated features while suppressing irrelevant background interference.

Table 4. Comparison of detection results of different models on the UA-DETRAC dataset.

Method	mAP	GFLOPs	Params/10^6	FPS	Car	Bus	Van	Others
RT-DETR	59.6	108.0	32.8	74.6	74.1	71.2	47.9	45.2
YOLOv3	63.0	283.0	103.6	43.10	76.7	78.6	49.2	47.6
YOLOv5s	60.0	16.0	7.02	95.3	71.5	75.3	48.4	44.6
YOLOv8s	60.8	25.8	11.1	110.2	73.3	76.3	46.8	46.6
YOLOv10s	60.4	24.8	8.1	111.0	73.7	75.6	47.8	44.4
YOLOv11s	61.7	21.5	9.4	113.5	74.5	78.3	48.2	45.6
Dong et al. (2022) [27]	61.5	13.5		-	-	-	-	-
Zhang et al. (2023) [38]	62.3	-	-	-	-	-	-	
Song et al. (2024) [17]	58.8	6.1	1.8	-	-	-	-	-
FFCA-YOLO (2024) [39]	61.2	51.5	7.1	48.6	73.2	79.6	49.3	42.7
Hyper-YOLO-S (2025) [40]	61.8	39.3	14.8	89.1	74.4	78.6	48.9	45.3
Ours	63.9	19.7	9.8	109.0	76.2	81.4	50.0	48.0

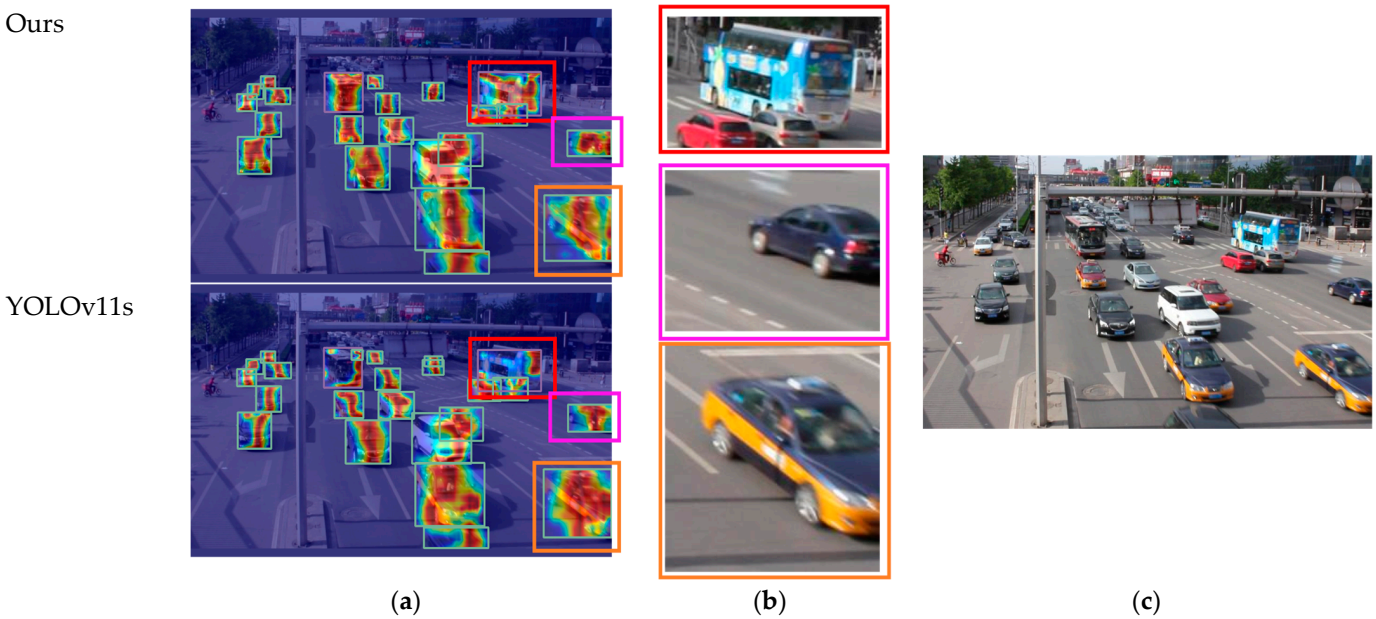


Figure 7. Comparison of YOLOV11s thermal map visualization. (**a**) Image Heatmap Recognition Result (**b**) Specific Car Model (**c**) Original Image.

Specifically, Dong et al. [27] introduced deep convolutional layers and various attention mechanisms, which significantly improved computational efficiency yet failed to achieve high detection accuracy. Zhang et al. [38] reduced the complexity of the detection model for better deployment on resource-constrained devices, but overlooked the impact of feature fusion on vehicle type recognition. Song et al. [17] integrated Mamba into the backbone network of YOLO, which greatly reduced the parameter count and enabled better information capture for vehicle targets. However, the feature extraction capability of the newly designed backbone network was compromised accordingly, leading to incomplete feature fusion of targets. In contrast, the proposed algorithm in this paper achieves a 5.1% improvement in recognition accuracy over the method in [17], making it more suitable for vehicle type recognition tasks.

Moreover, Zhang et al. [39] improved feature fusion to enhance the discrimination between background and targets, yet the weak feature extraction capability of the backbone network hindered the full utilization of feature information. On the contrary, this paper further strengthens feature representation by combining the strong feature extraction capability of the backbone network with the efficient feature fusion of the FCEM. Feng et al. [40] introduced hypergraphs into object detection and achieved promising recognition accuracy, but graph computation also drastically increased the computational burden. In comparison, the MSIFM maintains favorable computational efficiency while enhancing the multi-scale feature extraction capability, thus making it more applicable to vehicle type recognition tasks.

To further verify the generalization of the proposed method for vehicle type recognition across multiple scenarios, comparative analyses are conducted with various object detection algorithms on the BDD100K dataset. The BDD100K dataset features a larger data volume, more vehicle categories and more complex scenarios, thus imposing higher performance requirements on detection algorithms. The comparison results are presented in Table 5.

Table 5. Comparison of generalization performance of different models on BDD100K dataset.

Method	mAP	GFLOPs	Params/10^6	FPS	Car	Bus	Bike	Motor	Truck
RT-DETR	58.9	108.0	32.8	76.4	76.4	60.1	46.5	47.5	63.8
YOLOv3	58.2	283.0	103.6	42.6	76.3	61.2	42.0	46.5	65.1
YOLOv5s	56.4	16.0	7.02	95.3	70.5	58.4	44.6	44.8	63.7
YOLOv8s	60.2	25.8	11.1	105.2	79.6	63.1	46.8	46.6	64.8
YOLOv10s	58.3	21.6	7.2	106.0	79.0	60.0	45.2	43.8	63.7
YOLOv11s	60.2	21.5	9.4	110.5	80.3	62.1	47.5	48.5	62.6
FFCA-YOLO (2024) [39]	57.6	51.5	7.1	48.6	74.6	61.5	43.5	46.2	62.3
DTNet (2025) [41]	57.3	-	-	-	-	-	-	-	-
Ours	62.6	19.7	9.8	114.0	80.5	63.0	49.1	48.3	66.7

Table 5 demonstrates that the proposed method outperforms all comparative algorithms in both recognition accuracy and inference speed, achieving a favorable trade-off between real-time performance and detection efficacy. This result confirms the significant advantages of the proposed approach in terms of enhancing vehicle localization, preserving fine details and improving model robustness. Furthermore, the method maintains strong performance in complex scenarios involving diverse scenes and vehicle types, validating its excellent generalization capability.

4.4. Visualization Analysis

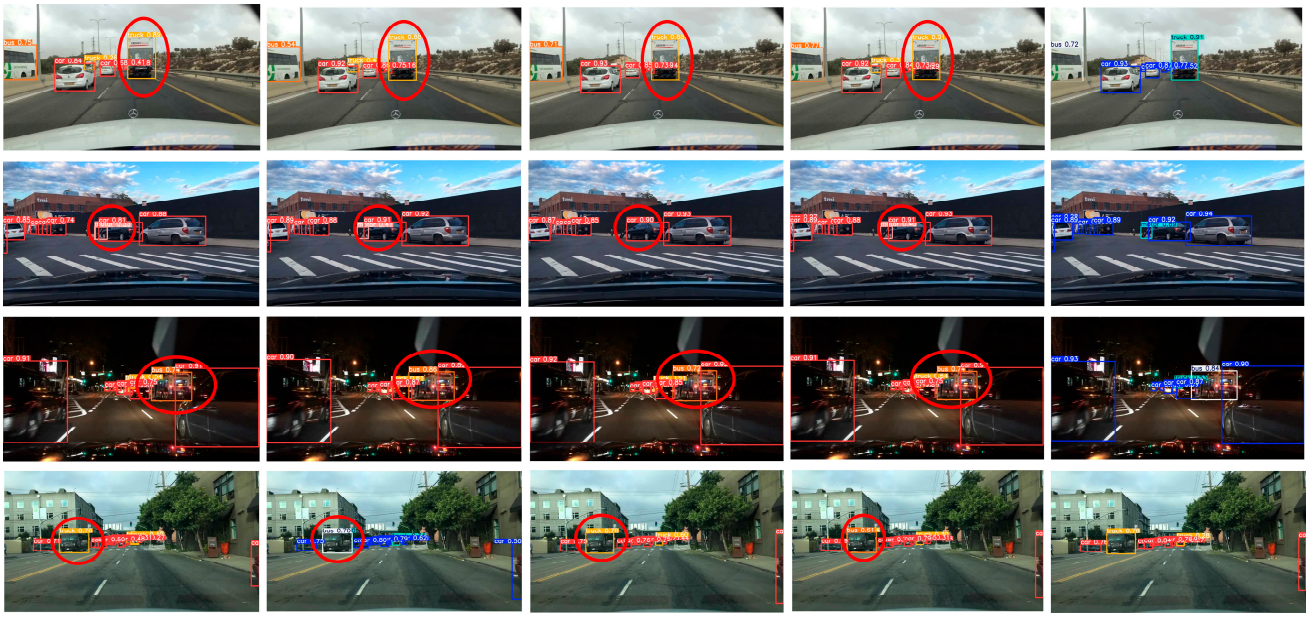
To demonstrate the vehicle type recognition performance of the proposed algorithm under various scenarios, comparative visualization results are presented under diverse

scene conditions. Figure 8 shows the recognition results of recent state-of-the-art algorithms and our method on UA-DETRAC. Figure 9 shows the recognition results on the BDD100K dataset.



(**a**) RT-Detr (**b**) YOLOv11s (**c**) Hyper-YOLO (**d**) FFCA-YOLO (**e**) Ours

Figure 8. Comparison of recognition performance on UA-DERAC dataset.



(**a**) RT-Detr (**b**) YOLOv10s (**c**) YOLOv11s (**d**) FFCA-YOLO (**e**) Ours

Figure 9. Comparison of recognition performance on BDD100K dataset.

In terms of detection performance, the proposed algorithm achieves superior recognition of distant vehicles, as shown in the first and third rows of Figure 8e. Despite the segmented targets and varying target sizes in these areas, the proposed algorithm still accurately identifies the vehicle targets, verifying its effective capture of multi-scale features. Meanwhile, our algorithm yields higher confidence scores for vehicle targets facing the camera and incomplete vehicle images at the image edges. These results demonstrate the strong capability of the proposed algorithm for vehicle type recognition in complex scenarios.

As can be seen from Figure 8, in terms of detection performance, the proposed algorithm achieves superior recognition of distant vehicles. Specifically, as indicated by the

red-circled regions in the first and third rows of Figure 8e, despite the segmented targets and varying target sizes in these areas, the proposed algorithm still accurately identifies the vehicle targets, which verifies its effective capture of multi-scale features. Meanwhile, as shown in the second and third rows of Figure 8e, our algorithm yields higher confidence scores for vehicle targets facing the camera and incomplete vehicle images at the image edges. The above results demonstrate the strong capability of the proposed algorithm for vehicle type recognition in complex scenarios.

According to Figure 9, YOLOv11s suffers from missed detections in the case of overlapping vehicle targets, while other comparative algorithms also face the issue of low recognition accuracy. In contrast, the proposed method accomplishes vehicle type recognition tasks more effectively, which verifies its favorable generalization performance and great potential for practical application.

4.5. Heatmap Comparison Analysis

To more intuitively highlight the attention degree of the algorithm to key regions, detection heatmaps are generated using HiResCAM technology. Red areas represent high-attention regions, while yellow areas represent secondary-attention regions.

Figure 10 shows the recognition heatmaps of different algorithms on the UA-DETRAC dataset. It can be observed from the enlarged heatmaps that the proposed algorithm pays higher attention to the edge contours of vehicles and focuses more on the global features of vehicles compared with other comparison algorithms. Meanwhile, it is less disturbed by the background environment and can focus more on distant targets, demonstrating favorable recognition accuracy and outstanding target perception capability.

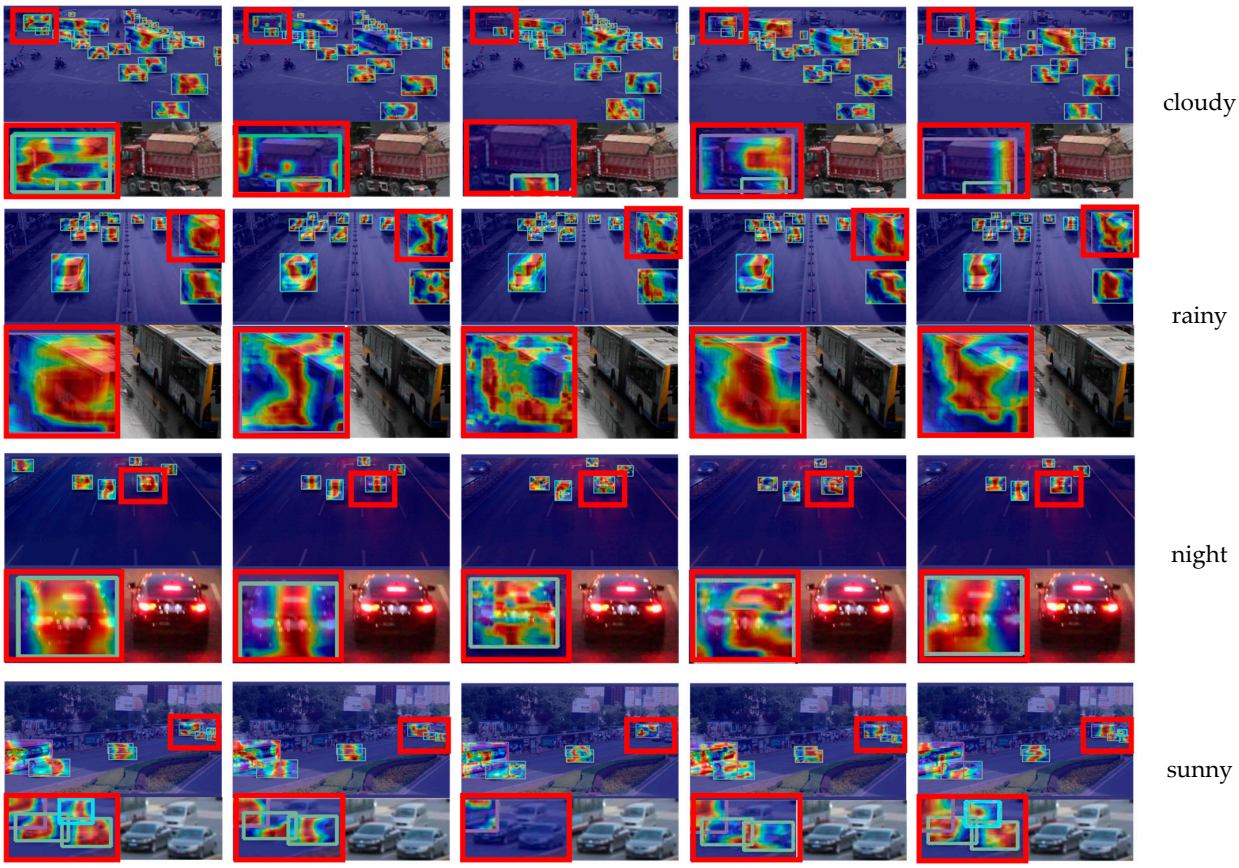


Figure 10. Comparison of heatmap recognition on UA-DETRAC dataset.

4.6. Limitations

Although the proposed vehicle type recognition network achieves competitive performance in terms of accuracy and efficiency on the UA-DETRAC and BDD100K datasets, several limitations still exist under extremely challenging conditions.

(1) First, the model experiences a significant performance drop under extreme weather conditions such as heavy snow and heavy rain. Severe environmental interference impairs feature extraction and reduces the discrimination between vehicle targets and the background. As a result, the FCEM cannot obtain sufficient effective features for contrast enhancement, and the MOEFEM also struggles to fully aggregate global and local features, leading to false detections or inaccurate bounding box regression, as illustrated in Figure 11.

(2) Second, although the model is lightweight, there is still room for optimization for edge deployment on low-cost embedded devices with constrained computing power and memory. The real-time inference speed and power consumption need to be further improved to better meet the requirements of real-world intelligent transportation edge devices.

In future work, we will address these limitations through the following approaches: (1) We will attempt to deploy the model on edge devices for real-world scenario testing. (2) We intend to prune the model to reduce the number of parameters. (3) We plan to collect and annotate data in more diverse environments, such as snowy and foggy conditions, to expand the dataset and improve the generalization ability of the model under various weather conditions.

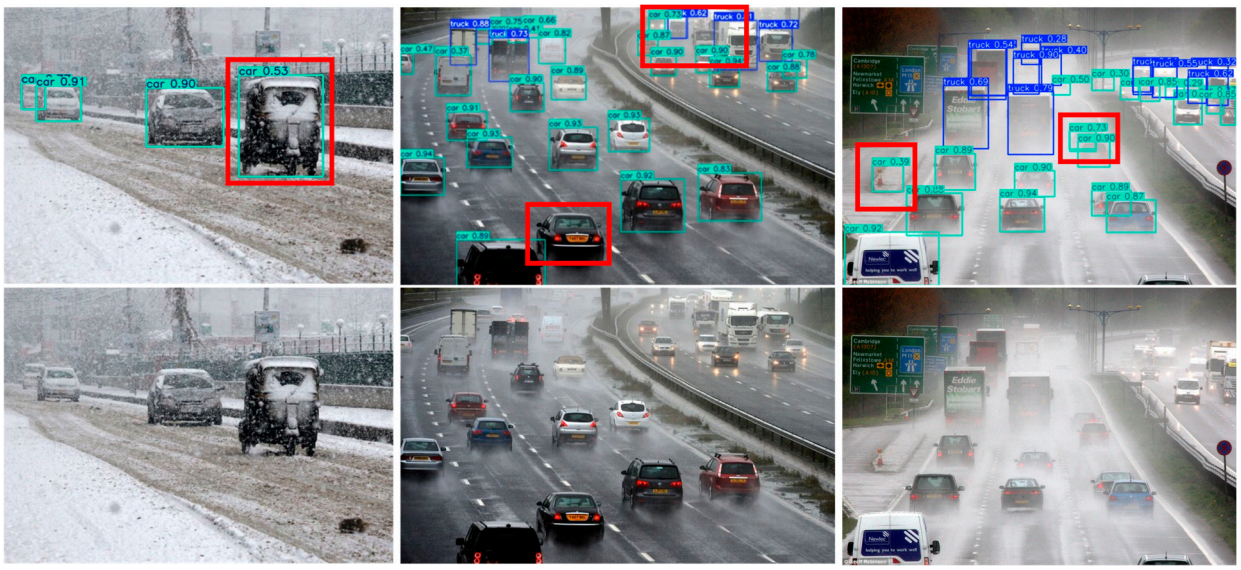


Figure 11. Identification results under extreme weather conditions.

5. Conclusions

In this paper, we propose a vehicle type recognition network based on feature comparison and the Mixture of Experts model. The Multi-scale Interleaving Fusion Module (MSIFM) efficiently extracts multi-scale features and reduces computation. The Feature Compare Enhancement Module (FCEM) strengthens critical feature representation. The Mixture of Experts Feature Enhancement Module (MOEFEM) adaptively improves vehicle localization accuracy. Experimental results on the UA-DETRAC and BDD100K datasets show that our method improves mAP by 2.2% and 2.4% over YOLOv11, respectively, while maintaining real-time inference speed, achieving a good balance between accuracy

and efficiency. Ablation studies verify the effectiveness of each module, and visualization results show that the model focuses more accurately on vehicle edge contours and is more robust to background interference. However, performance degrades under extreme weather such as heavy snow and rain, and there is still room for optimization for low-cost embedded edge deployment. Future work will improve the model from three aspects, namely, edge deployment, model lightweighting, and dataset expansion under extreme weather to further enhance practicality and generalization.

Author Contributions: T.H.—Methodology, writing—original draft. X.Z.—Writing—review and editing. L.Y.—Methodology, validation, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Shanxi Provincial Key Research and Development Program of China, grant number 202102010101008, and the Shanxi Province Basic Research Program (Free Exploration) Project, grant number 202403021222276.

Data Availability Statement: The original contribution of this research is included in the paper. For further inquiries, please contact the first author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xiang, Y.; Fu, Y.; Huang, H. Global topology constraint network for fine-grained vehicle recognition. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 2918–2929. [CrossRef]
2. Barth, M.; Sanchez, J.J. Guest Editorial Special Issue: The 21st IEEE International Conference on Intelligent Transportation Systems (ITSC 2018). *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 3929–3930. [CrossRef]
3. Sliwa, B.; Piatkowski, N.; Wietfeld, C. The Channel as a Traffic Sensor: Vehicle Detection and Classification based on Radio Fingerprinting. *IEEE Internet Things J.* **2020**, *7*, 7392–7406. [CrossRef]
4. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
5. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition, San Diego, CA, USA, 20–25 June 2025*; IEEE: New York, NY, USA, 2025; Volume 1, pp. 886–893.
6. Odat, E.; Shamma, J.S.; Claudel, C. Vehicle Classification and Speed Estimation Using Combined Passive Infrared/Ultrasonic Sensors. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 1593–1606. [CrossRef]
7. Nguyen, T.M.; Wu, Q.M.J. Gaussian-Mixture-Model-Based Spatial Neighborhood Relationships for Pixel Labeling Problem. *IEEE Trans. Syst. Man Cybern. Part B* **2012**, *42*, 193–202. [CrossRef]
8. Li, F.; Lv, Z. Reliable vehicle type recognition based on information fusion in multiple sensor networks. *Comput. Netw.* **2017**, *117*, 76–84. [CrossRef]
9. Macioszek, E.; Kurek, A. Road traffic distribution on public holidays and workdays on selected road transport network elements. *Transp. Probl.* **2021**, *16*, 127–138. [CrossRef]
10. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference On Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
11. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
12. 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.* **2016**, *39*, 1137–1149. [CrossRef]
13. Ke, X.; Zhang, Y. Fine-grained vehicle type detection and recognition based on dense attention network. *Neurocomputing* **2020**, *399*, 247–257. [CrossRef]
14. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017*; IEEE: New York, NY, USA, 2017; pp. 7263–7271.
15. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. *SSD: Single Shot Multibox Detector*; Springer: Cham, Switzerland, 2016. [CrossRef]
16. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019*; IEEE: New York, NY, USA, 2020; pp. 9627–9636.
17. Song, Z.; Wang, Y.; Xu, S.; Wang, P.; Liu, L. Lightweight Vehicle Detection Based on Mamba_ViT. *Sensors* **2024**, *24*, 7138. [CrossRef] [PubMed]

18. Kasper-Eulaers, M.; Hahn, N.; Berger, S.; Sebulonsen, T.; Myrland, Ø.; Kummervold, P.E. Short Communication: Detecting heavy goods vehicles in rest areas in winter conditions using YOLOv5. *Algorithms* **2021**, *14*, 114. [CrossRef]
19. Sun, W.; Zhang, G.; Zhang, X.; Zhang, X.; Ge, N. Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning strategy. *Multimed. Tools Appl.* **2021**, *80*, 30803–30816. [CrossRef]
20. Cao, C.-Y.; Zheng, J.-C.; Huang, Y.-Q.; Liu, J.; Yang, C.-F. Investigation of a promoted you only look once algorithm and its application in traffic flow monitoring. *Appl. Sci.* **2019**, *9*, 3619. [CrossRef]
21. Jheng, Y.-J.; Yen, Y.-H.; Sun, T.-Y. A symmetry-based forward vehicle detection and collision warning system on Android smartphone. In *Proceedings of the 2015 IEEE International Conference on Consumer Electronics-Taiwan*; IEEE: New York. NY, USA, 2015; pp. 212–213.
22. Fung, G.S.K.; Yung, N.H.C.; Pang, G.K.H. Vehicle shape approximation from motion for visual traffic surveillance. In *Proceedings of the ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings, Oakland, CA, USA, 25–29 August 2001*; IEEE: New York. NY, USA, 2002; pp. 608–613.
23. Lim, T.R.; Guntoro, A.T. Car recognition using Gabor filter feature extraction. In *Asia-Pacific Conference on Circuits and Systems*; IEEE: New York, NY, USA, 2002; Volume 2, pp. 451–455.
24. Wen, X.; Shao, L.; Fang, W.; Xue, Y. Efficient feature selection and classification for vehicle detection. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *25*, 508–517. [CrossRef]
25. Hsieh, J.W.; Chen, L.C.; Chen, D.Y. Symmetrical SURF and its applications to vehicle detection and vehicle make and model recognition. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 6–20. [CrossRef]
26. Yin, S.; Li, H.; Teng, L. Airport detection based on improved faster RCNN in large scale remote sensing images. *Sens. Imaging* **2020**, *21*, 49. [CrossRef]
27. Dong, X.; Yan, S.; Duan, C. A lightweight vehicles detection network model based on YOLOv5. *Eng. Appl. Artif. Intell.* **2022**, *113*, 104914. [CrossRef]
28. Kang, L.; Lu, Z.; Meng, L.; Gao, Z. YOLO-FA: Type-1 fuzzy attention based YOLO detector for vehicle detection. *Expert Syst. Appl.* **2024**, *237*, 121209. [CrossRef]
29. Bie, M.; Liu, Y.; Li, G.; Hong, J.; Li, J. Real-time vehicle detection algorithm based on a lightweight You-Only-Look-Once (YOLOv5n-L) approach. *Expert Syst. Appl.* **2023**, *213*, 119108. [CrossRef]
30. Dong, X.; Shi, P.; Qi, H.; Yang, A.; Liang, T. TS-BEV: BEV object detection algorithm based on temporal-spatial feature fusion. *Displays* **2024**, *84*, 102814. [CrossRef]
31. Riquelme, C.; Puigcerver, J.; Mustafa, B.; Neumann, M.; Jenatton, R.; Pinto, A.S.; Keysers, D.; Houlsby, N. Scaling vision with sparse mixture of experts. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 8583–8595.
32. Han, X.; Wei, L.; Dou, Z.; Sun, Y.; Han, Z.; Tian, Q. Vimoe: An empirical study of designing vision mixture-of-experts. *IEEE Trans. Image Process.* **2025**, *34*, 7209–7221. [CrossRef] [PubMed]
33. Videau, M.; Leite, A.; Schoenauer, M.; Teytaud, O. Mixture of Experts in Image Classification: What's the Sweet Spot? *arXiv* **2024**, arXiv:2411.18322. [CrossRef]
34. Liu, P.; Zhang, H.; Zhang, C.; Jiang, F. Hybrid-Frequency-Aware Mixture-of-Experts Method for CT Metal Artifact Reduction. *Mathematics* **2026**, *14*, 494. [CrossRef]
35. Qin, D.; Leichner, C.; Delakis, M.; Fornoni, M.; Luo, S.; Yang, F.; Wang, W.; Banbury, C.; Ye, C.; Akin, B.; et al. MobileNetV4: Universal models for the mobile ecosystem. In *European Conference on Computer Vision*; Springer Nature: Cham, Switzerland, 2024; pp. 78–96.
36. Wen, L.; Du, D.; Cai, Z.; Lei, Z.; Chang, M.C.; Qi, H.; Lim, J.; Yang, M.H.; Lyu, S. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Comput. Vis. Image Underst.* **2020**, *193*, 102907. [CrossRef]
37. Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; Darrell, T. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020*; IEEE: New York, NY, USA, 2020; pp. 2636–2645.
38. Zhang, Z.; Xu, H.; Lin, S. Quantizing yolov5 for real-time vehicle detection. *IEEE Access* **2023**, *11*, 145601–145611. [CrossRef]
39. Zhang, Y.; Ye, M.; Zhu, G.; Liu, Y.; Guo, P.; Yan, J. FFCA-YOLO for small object detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5611215. [CrossRef]
40. Feng, Y.; Huang, J.; Du, S.; Ying, S.; Yong, J.-H.; Li, Y.; Ding, G.; Ji, R.; Gao, Y. Hyper-YOLO: When Visual Object Detection Meets Hypergraph Computation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 2388–2401. [CrossRef]
41. Tian, C.; Liu, K.; Zhang, B.; Huang, Z.; Lin, C.-W.; Zhang, D. A Dynamic Transformer Network for Vehicle Detection. *IEEE Trans. Consum. Electron.* **2025**, *71*, 2387–2394. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.