SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2024, Vol. 67, Iss. 11, 212208:1–212208:16
https://doi.org/10.1007/s11432-023-4116-5

All-day perception for intelligent vehicles: switching perception algorithms based on WBCNet

Hongbin XIE[1], Haiyan ZHAO[1*], Chengcheng XU[1] & Hong CHEN[2]

[1]*College of Communication Engineering, Jilin University, Changchun* 130025*, China;*
[2]*College of Electronic and Information Engineering, Tongji University, Shanghai* 201804*, China*

Received 9 October 2023/Revised 22 April 2024/Accepted 8 May 2024/Published online 22 October 2024

Abstract A weather- and brightness-based classification network (WBCNet) is proposed for driving scene classification to address the decreased accuracy in perception caused by weather and environment changes. To facilitate its applicability in vehicles and minimize computational demands on vehicle chips, WBCNet has been designed with special modules, including attention mechanisms and dilated convolutions. Dilated convolutions combined with residual connections empower WBCNet to concurrently handle information at various scales. This aids in simplifying the training and optimization of deep networks, consequently enhancing the model's performance and mitigating the risk of overfitting. The outstanding feature association capability originating from the fusion of channel attention and spatial attention enables WBCNet to focus more on the sky, lanes, and other traffic information features within the image. This design enables WBCNet to use only images as input, making it highly suitable for engineering applications. The output of WBCNet provides the basis for the downstream perception model selection algorithm, allowing it to choose the appropriate perception model for different scenes accurately. A dataset with complex scenes based on Carla is constructed for comparison to verify WBCNet's performance. Finally, a real-world driving dataset is used to validate the effectiveness and real-time performance of WBCNet.

Keywords driving scene classification, intelligent vehicle, safe driving, convolutional neural network, attention mechanism

1 Introduction

Intelligent vehicles (IVs) offer new possibilities for the future of transportation [1, 2] and can greatly improve traffic congestion, save energy, and decrease other transportation issues [3–5]. Autonomous driving systems face significant challenges due to varying weather conditions and different times of the day. In this respect, improving the perception accuracy is one of the important tasks in the field of IV research [6, 7]. Accurate perception provides an important basis for downstream tasks in a timely manner to ensure the safe operation of IVs, such as decision-making and path planning [8]. Most perception algorithms for IVs are subject to specific scenes. The weather and the time of the day during driving have various negative impacts on transportation. When the weather and environment change, the accuracy of the perception algorithm is reduced to varying degrees and results in certain safety risks. By accurately identifying driving scenes, IVs can dynamically adjust their perception algorithms based on the characteristics of the scene to adapt to the current environment. Therefore, a framework that can achieve accurate perception around the clock is particularly important for IVs.

Weather classification is a new task derived from image classification. Changes in weather and environment brightness can cause changes in visibility. The problem of classification first arose in the field of machine learning. Several solutions have emerged, such as K-nearest neighbor algorithms [9, 10], decision trees [11, 12], and support vector machines [13, 14]. With the increase in computing power of computers and the development of deep learning, VGG and ResNet were also proposed for image classification [15,16]. ResNet has a wide range of applications in the field of classification. Meanwhile, DropCNN was proposed to classify vehicles based on ResNet, and a method named JF was explored to improve the accuracy

* Corresponding author (email: zhao_hy@jlu.edu.cn)

of classification [17]. ResNet was also used to classify malicious software [18]. The pretrained weights on an ImageNet dataset are further trained using transfer learning. Classification of malicious software can be achieved efficiently. There have also been ResNet applications in the medical field [19]. Dilated convolution (DC) and pyramid structures were combined to achieve an accurate classification of glioma types. A dataset of common weather conditions consisting of a total of six classes was proposed previously [20]. The authors discovered that a subset of image regions was sufficient for weather recognition, whereas an excessive amount of image information could even pose challenges for the classification model. A model that combined region selection and coexistence modules was constructed for the task based on the observation. Meanwhile, the phenomenon of edge degradation was introduced previously [21] based on the scattering effect of aerosols on light. Then, it was combined with a convolutional neural network (CNN) for weather classification. Future, a network training process was proposed by combining mask region-based CNN [22] and VGG-16 [15], where the former was utilized to extract the region of interest, and the latter was employed for their classification. Transfer learning is also used in weather classification. A classification model based on transfer learning was proposed and achieved good accuracy on a proposed dataset [23]. To further increase the correctness of weather recognition, some scholars began to consider weather-related cues. A network architecture called CNN-based multitask weather recognition network, which combined weather classification and weather clue segmentation tasks, was proposed previously [24]. Because of the network's requirement to accomplish both tasks, an adaptive loss function with weighted schemes for each task was designed. The weight coefficients could be trained according to task requirements. Meanwhile, with the development of object detection, weather classification models have undergone new changes. For instance, a previous study [25] improved the YOLOv5 [26] model for weather recognition by incorporating nonforgetting learning techniques. EfficientNet was used for image classification using the compound scaling method to balance model size and performance [27]. Transformers originate from natural language processing. When the training dataset is large enough, they can achieve about the same results as those of common classification networks in ImageNet. Swin is a classification network based on vision transformer (ViT) and demonstrates the importance of transformers in the computer vision field [28]. ViT further eliminates some of the difficult decisions of encoding in a convolutional architecture. To better advance the evolution of less hard-coded networks, ResMLP has been proposed [29]. Then, a neural network structure was redesigned to propose ConvNeXt that reached a new high level in image classification [30].

Object detection is one of the crucial tasks for ensuring the safe navigation of IVs [26, 31, 32]. An accurate perception of the surrounding environment is essential for vehicles to make informed decisions regarding route planning. Thus, effective detection of vehicles in various weather and lighting conditions is of paramount importance. In this regard, a previous work [32] proposed a real-time object detection method called YOLOv3, which used Darknet53 as the backbone network. Progress has also been made in object detection at night. A method for nighttime vehicle detection and tracking using the localization of vehicle lights was proposed previously [33]. However, the object detection methods described above are limited to specific light intensities. An object detection algorithm specifically designed for daytime may not be able to achieve better results at night, and vice versa. Therefore, adaptive object detection algorithms that adapt to various luminance levels have been proposed. For instance, a previous study [34] proposed an adaptive object detection network that could operate under varying brightness conditions. The network initially extracted image features, which were then adaptively adjusted before conducting object detection. Experimental results demonstrated the network's excellent performance metrics in both low and normal brightness environments. Research has also been gradually conducted in different weather conditions, such as rain and snow with poor visibility. For instance, another previous study [35] proposed a method to improve the detection accuracy of lightweight object detection networks. The method was composed of four subnetworks and verified on a proposed iRAIN dataset, which achieved accurate object detection without consuming too much time. The idea of adaptiveness was then applied to various weather conditions. An enhancement module to the YOLO framework was proposed for adaptive object detection, enabling image augmentation and thereby improving detection accuracy under adverse conditions [36]. In addition, several researchers investigated algorithms for object detection with robustness in various weather conditions. Novel improvements have been presented on the YOLO framework by incorporating adversarial learning techniques, enabling the network to adapt to images captured in adverse weather conditions [37]. The researchers can primarily be categorized into two groups. One group of researchers focused on scene classification, whereas the another group specialized in designing adaptive object detection algorithms. IVs can benefit from selecting different perception

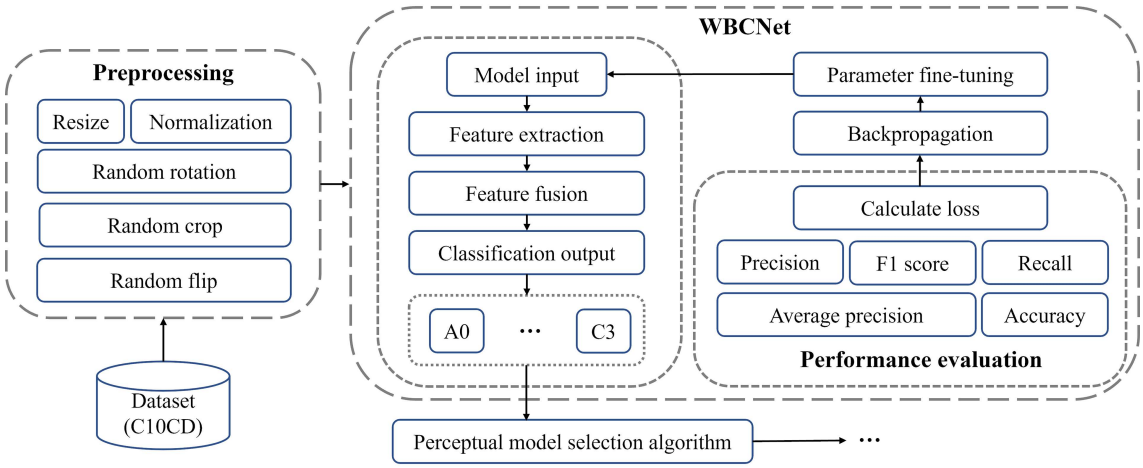

Figure 1 (Color online) Solution framework for this paper.

algorithms based on their current surroundings or scenes, which can lead to more effective problem-solving in diverse driving situations. Figure 1 illustrates the proposed solution framework in this paper, which is elaborated upon in subsequent sections.

This paper proposes a novel driving scene classification network (weather- and brightness-based classification network, WBCNet) that integrates multilevel feature streams, attention mechanisms, and DCs to overcome the insufficient perception accuracy of a single algorithm in different scenes. It only uses images captured by a camera to classify the current environment of a vehicle. In the WBCNet network, various new plug-and-play feature extraction modules are specially designed. The incorporation of DCs, residual connections, and attention mechanisms within WBCNet brings several advantages. These design elements enable WBCNet to focus on important aspects of the image, such as the sky, lanes, and other traffic-related features, enhancing its ability to extract relevant information. Furthermore, the utilization of DCs in conjunction with residual connections allows WBCNet to effectively handle information at multiple scales simultaneously. This unique feature not only simplifies the training and optimization of deep networks but also bolsters the model's performance. Additionally, it helps reduce the risk of overfitting. Because of the difficulty of collecting data in certain scenarios in real-world settings, a dataset called Carla10Classes dataset (C10CD) based on a Carla simulator that encompasses a wide range of common driving scenes is constructed. C10CD offers a finer-grained classification, setting it different from other datasets. Finally, a real-world driving dataset is used to test the generalization performance and real-time capabilities of WBCNet.

2 Network structure

WBCNet is designed to avoid taking up too much computational resources. For example, the model features cheap and easy-to-obtain inputs, clever integration of attention mechanisms, and better use of DCs. For a more comprehensive understanding of WBCNet, this section shows the detailed architecture of plug-and-play network modules and WBCNet.

2.1 WBCNet

Weather and environment brightness are critical influencing factors in driving scene classification. As shown in Figure 2, WBCNet only needs images as inputs. Cameras are the cheapest among various vehicle sensors. Identifying weather conditions through images can save computing resources and the cost of sensors for vehicles, making it easy to deploy to real vehicles. The images captured by a camera are used as input for the feature extraction part. These images are extracted and abstracted layer by layer using specially designed modules. The original data are gradually mapped into a high-level feature space through nonlinear transformations. The feature fusion part fuses the two feature streams of the feature extraction part. The high-level features extracted from the first two parts are integrated and transformed by the classification output part of WBCNet to produce the final output.

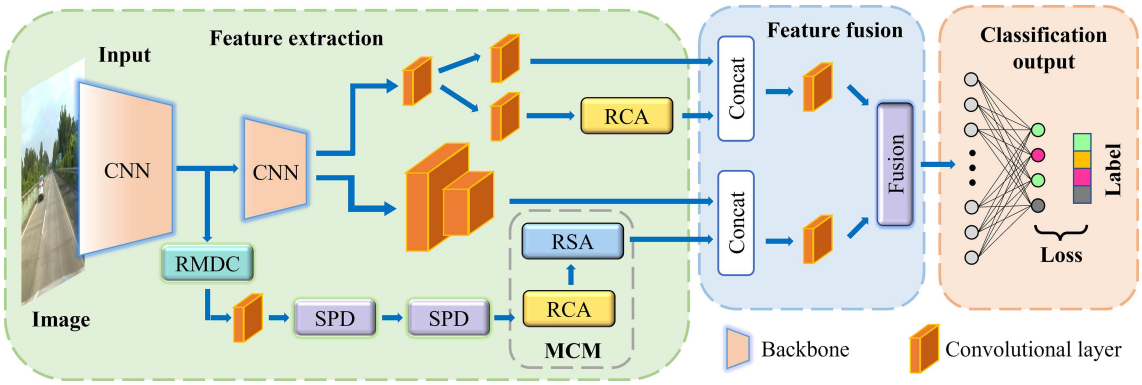

Figure 2 (Color online) Architecture diagram of the WBCNet. WBCNet mainly consists of three parts: feature extraction, feature fusion, and classification output.

2.1.1 *Feature extraction part*

ResNet is selected as the backbone network for WBCNet. As shown in Figures 2 and 3, ResNet contains a total of four stages. Branch A contains features at the end of Stage2 to input low-level features into the residual multiscale DC (RMDC). Branch B contains the total features extracted by ResNet, which is used for subsequent network modules to extract more detailed features and prepare for blending low-level features from Branch A.

The feature extraction part contains several feature extraction modules. The space-to-depth (SPD) module, originating from YOLO, is a feature transformation operation used to change the shape of a tensor. It rearranges the input tensor according to rules that convert spatial dimension features into depth dimension. The ability of WBCNet to perceive multiscale objects is increased by SPD without losing spatial information. Compared to existing modules, WBCNet simultaneously retains feature representations at different scales. SPD is used in WBCNet instead of pooling layers to reduce the size of feature maps while preserving useful features as much as possible. It does not introduce additional parameters, which helps to reduce the complexity of the model and improve the computational efficiency. Additionally, SPD assists neural networks in extracting intricate features. Our proposed RMDC, multi-attention convolutional module (MCM), and residual channel attention (RCA) module are described in detail in the following section.

RMDC. DC allows a larger range of features to be extracted without the use of pooling layers. Compared with the traditional convolution layer, DC has unique advantages in extracting multilevel features, especially in expanding the receptive field while keeping the number of parameters unchanged. In traditional convolution operations, the size of the receptive field is determined by the size of the convolution kernel. However, larger convolution kernels result in larger parameters, which evidently increase the complexity and computational cost of the model. DC has an important parameter called the dilated rate. It can effectively expand the range of the receptive fields without increasing the number of parameters. DC allows a 3×3 convolution kernel to have the field of view (FOV) of a 5×5 convolution kernel or an even larger receptive field by adjusting the dilated rate. As shown in Figure 4, multiscale feature information is simultaneously obtained by combining DC with different dilated rates. RMDC combines the advantages of DC and pyramid structure. This design can capture the multilevel features of an image, allowing the model to comprehensively understand the image in greater detail. In addition, a pyramid-shaped feature group is stacked with low-level feature maps by residual connection. High-level abstract features are learned better by residual connection while preserving the details of low-level features and avoiding learning bad features. Convolutional layers with convolutional kernel size of 1×1 are added to transform the input feature map. The increase and decrease in feature dimension are achieved by the 1×1 convolution layer while maintaining the connectivity of the network.

MCM. Attention mechanisms are increasingly applied to CNNs [38–40]. They have a good effect on the network performance and are easy to add to the network because of the convenience of plug-and-play. In this respect, the convolutional block attention module (CBAM) shows that both channel and spatial information are very important. However, CBAM may have some limitations in certain scenes, as found in our experimental observations. Channel attention and spatial attention are two different ways of applying the attention mechanism in image processing. MCM is designed and proposed as inspired by CBAM.

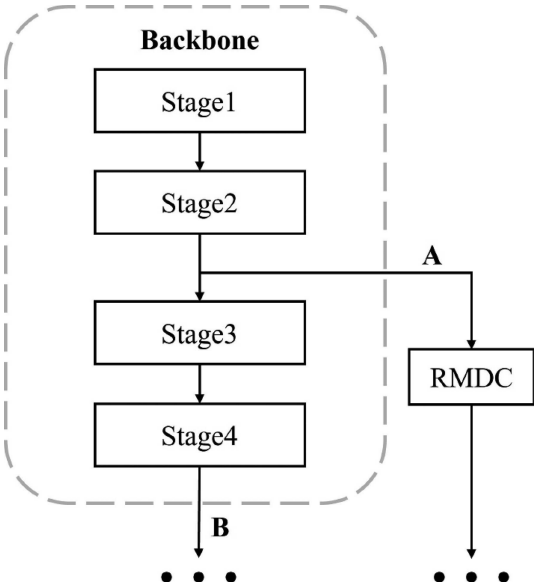

Figure 3 Detailed structure of the backbone network in the feature extraction part of WBCNet. Branch A contains low-level features, and Branch B contains high-level features. The two are fused in the feature fusion part.

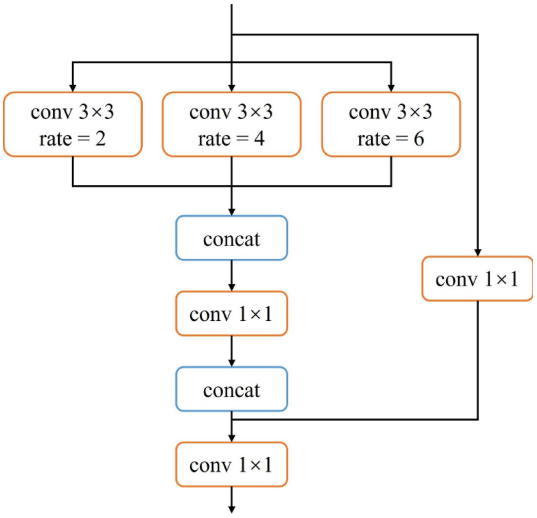

Figure 4 (Color online) Detailed structure of the RMDC. The number after conv represents the size of the convolution kernel. Herein, 2, 4, and 6 are chosen as the dilated rate, corresponding to smaller, medium, and larger receptive fields, respectively.

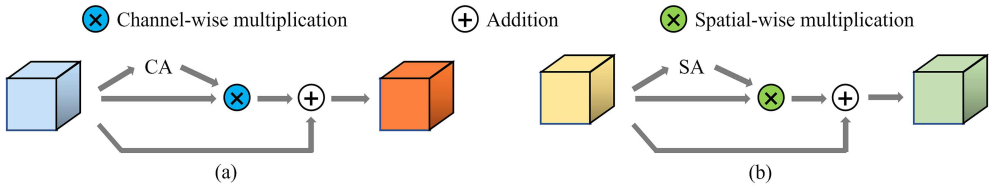

Figure 5 (Color online) Detailed structures of the RCA and RSA modules. (a) RCA; (b) RSA.

MCM consists of two parts: the RCA module and the residual spatial attention (RSA) module. The detailed structures of the two modules can be seen in Figure 5. The interrelations among various channels within an image are highlighted by RCA, empowering the model to discern the varying significance of each channel. For instance, attributes like color, texture, shape, and other features are typically represented across distinct channels. The main task of the RCA module is to calculate the weight of each channel in the feature map to evaluate its importance in the entire feature representation. WBCNet focuses on information from different channels adaptively, strengthens useful features, weakens useless features, and improves its expressive and generalization abilities. Meanwhile, RSA tends to extract important information in the spatial dimension. WBCNet's comprehension of the structure and spatial correlations of the image is enhanced by RSA. The RSA module makes the network focus on a specific spatial position by weighting each spatial position of the feature map. It aids WBCNet in comprehending the subject's position and size within the image, enhancing the model's balance between local and global features. MCM fully considers the feature information of different channels and different locations by combining RCA and RSA. The multichannel and multiscale information in the image is fully utilized by MCM. The design motivation of MCM is to improve the performance and generalization of the model. At the same time, a more powerful and efficient feature extraction capability is provided for the feature extraction part.

2.1.2 *Feature fusion part*

This part merges low-level features with high-level features.

$$y_{\text{cat/sum/max/avg}} = f^{\text{cat/sum/max/avg}}(x_l, x_h), \tag{1}$$

where x_{l} represents low-level features and x_h represents high-level features, f^* stands for the fusion method for low- and high-level features.

As shown in (1), four different feature fusion methods are proposed: concatenate (concat), maximum (max), summation (sum), and average (avg).

The concat fusion method stacks low- and high-level feature maps in the channel dimension. Concat can provide a rich and comprehensive representation of information, thereby enhancing the model's understanding of images. By combining features from multiple dimensions, the risk of information loss can be reduced, ultimately boosting the model's robustness. The fusion formula is as follows:

$$y_{\mathrm{cat}} = \mathrm{concat}(x_{\mathrm{l}}, x_h), \tag{2}$$

where $y_{\mathrm{cat}} \in \mathbb{R}^{2C\times W\times H}$, C represents the number of channels in the feature map, W and H represent the width and height of the feature map, respectively.

$$y_{\max\_c,i,j} = \max\{x_{a\_c,i,j}, x_{b\_c,i,j}\}, \tag{3}$$

$$y_{\mathrm{sum}\_c,i,j} = x_{a\_c,i,j} + x_{b\_c,i,j}, \tag{4}$$

$$y_{\mathrm{avg}\_c,i,j} = \frac{x_{a\_c,i,j} + x_{b\_c,i,j}}{2}, \tag{5}$$

where $c \in \{1, 2, \ldots, C\}, i \in \{1, 2, \ldots, W\}, j \in \{1, 2, \ldots, H\}$, $y_{\mathrm{sum}}, y_{\max}, y_{\mathrm{avg}} \in \mathbb{R}^{C\times W\times H}$.

The other three fusion methods are shown in (3)–(5). These three fusion methods show the max, sum, and average of the corresponding positions of low- and high-level feature maps, respectively. These three methods differ from the stacking operation of concatenation. These methods can reduce model complexity and computational costs, thereby enhancing both training and inference efficiency. These approaches are more conducive to focusing on critical information and reducing the impact of noise.

Because the fusion function only fuses the two feature streams using the four methods proposed above, it does not extract features. Therefore, after using the fusion function, a convolution layer is used to further select features. The effects of four fusion methods on network performance are compared in Section 3.

2.1.3 *Classification output part*

After a series of convolution operations, the feature extraction and feature fusion parts gradually abstract the image into a set of high-level feature maps. These feature maps capture important features at different locations and levels of the image, such as edges, textures, and shapes. The representations of these feature maps are not suitable for input into a classifier directly because the classifier requires a fixed-length vector rather than a two-dimensional feature map.

Fully connected neural networks (FCNNs) are important structures for completing classification tasks. After obtaining the convolutional layer in the feature fusion part, a series of high-level feature maps are obtained. To perform image classification tasks, these feature maps are flattened and input into an FCNN. FCNN integrates and transforms features extracted from the convolutional layers to generate final classification results, accurately determining the scene to which the input image belongs. The classification output part finally contains 10 neurons, each representing a different driving scene.

2.2 Implementation

The following shows the details of the network training process:

$$\mathcal{L}(x, l) = -\log\left(\frac{\mathrm{e}^{x_{[l]}}}{\sum_j \mathrm{e}^{x_j}}\right) = \log\left(\sum_j \mathrm{e}^{x_j}\right) - x_{[l]}, \tag{6}$$

where x is the predicted label and l represents the actual label.

The cross entropy loss function in (6) is selected for training, which is effective for multiclass classification tasks. The cross entropy loss function is based on probabilistic modeling, allowing it to capture the probability distribution between different categories. Moreover, it operates on a logarithmic scale,

amplifying the loss and thus accelerating the training speed. The prediction probability of the model is as close as possible to the real label by reducing the cross entropy loss.

To balance the training speed and the convergence of the model, we choose a learning rate of 6.5×10^{-4} for the network based on our training experience. A smaller learning rate can make the model converge more stably. In addition, an early stopping strategy is used in training to avoid overfitting. Early stopping is a method to prevent overfitting. In the process of model training, whether or not to stop training is judged according to the performance of the model on the validation set. Specifically, the value of the model's loss function on the validation set is monitored. If the validation set loss does not decrease significantly in successive training iterations, the training is terminated to avoid overfitting the model on the training set. WBCNet is trained on an NVIDIA GeForce RTX 3090 GPU.

2.3 Evaluation metrics

To comprehensively evaluate the classification performance of the network, the following metrics are chosen as the evaluation metrics of the network:

$$P(\text{macro}) = \frac{1}{n}\sum_{i=1}^{n} P_i', \tag{7}$$

where n represents the total number of classes in the dataset, and P' represents precision in each class.

P (precision). It measures how many of the samples predicted by the classifier as positive classes are true positive samples. As shown in (7), a higher P means a lower false positive.

$$R(\text{macro}) = \frac{1}{n}\sum_{i=1}^{n} R_i', \tag{8}$$

where R' represents recall in each class.

R (recall). It measures the classifier's ability to correctly recognize all positive samples. In (8), the higher the recall, the lower the probability of the classifier failing to report, which means that more positive samples can be found.

$$\begin{aligned} \text{F1}(\text{macro}) &= \frac{1}{n}\sum_{i=1}^{n} \text{F1}_i \\ &= \frac{1}{n}\sum_{i=1}^{n} 2 \times \frac{R_i' \times P_i'}{R_i' + P_i'}. \end{aligned} \tag{9}$$

F1 (macro-F1 score). The F1 score considers both precision and recall comprehensively. In (9), macro-F1 treats all classes equally. It can reflect the comprehensive performance of the model.

Acc (accuracy). A high accuracy indicates that the classification model has a high accuracy in predicting the overall sample.

AP (average precision), mAP (mean average precision). AP is obtained by calculating the area under the precision-recall curve. It is a measure of a classifier's performance on a single class. The higher the value of AP, the better the performance of the classifier. Meanwhile, the mAP is an indicator obtained by averaging all classes of APs and integrating the performance of classifiers across all classes, providing a more comprehensive evaluation of the overall performance of the classification model.

3 Experiments

To prove the classification performance of WBCNet, a classification experiment is launched. The validity of the proposed three modules is verified in this section. Results show the importance of driving scene classification for autonomous driving.

3.1 Dataset preparation

Carla is an open source simulator for autonomous driving systems. It provides various scenes, including highways, city roads, and tunnels, which can simulate various situations. At the same time, Carla allows us to easily adjust the weather and sunlight conditions of the simulated scene, providing a richer data

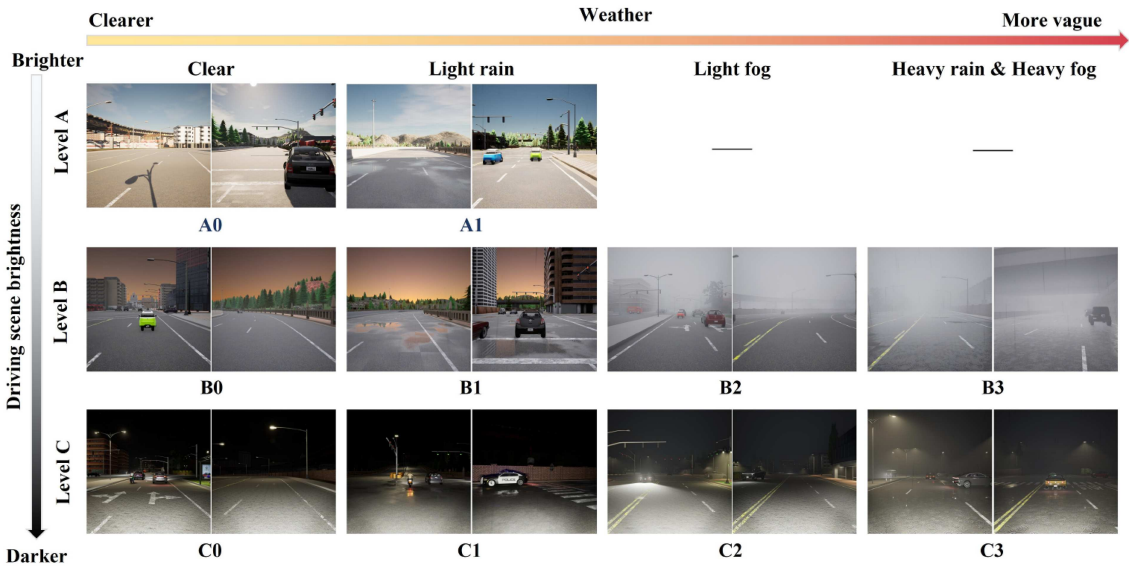

Figure 6 (Color online) Presentation of all classes of images in the C10CD dataset. C10CD contains a total of 10 classes.

Table 1 Comparison of parameters in each class of the C10CD dataset

Class No.	Class	Brightness	Weather	Number of images
1	A0	Light (A)	Clear	400
2	A1	Light (A)	Light rain	400
3	B0	Medium (B)	Clear	400
4	B1	Medium (B)	Light rain	400
5	B2	Medium (B)	Light fog	400
6	B3	Medium (B)	Severe	400
7	C0	Dark (C)	Clear	400
8	C1	Dark (C)	Light rain	400
9	C2	Dark (C)	Light fog	400
10	C3	Dark (C)	Severe	400

Table 2 Transformation and augmentation methods

Transformation and augmentation	Value
Resize	(340, 340)
Random crop	(224, 224)
Random rotation	−10–10°
Random horizontal flip	True
Random vertical flip	False

sample for training and testing autonomous driving systems. During the data collection process, an RGB color camera was chosen as the sensor to capture the dataset. The camera's horizontal FOV was set to 60°. This ensured that the camera image covered important information in front of the vehicle. RGB color images can provide rich color information, which is important for the classification of driving scenes. All classes of images in the C10CD dataset are shown in Figure 6.

The composition of the dataset is shown in Table 1. According to the environment brightness, three levels were classified: Levels A, B, and C. According to the weather conditions, C10CD was divided into four classes, represented by numbers 0 to 3 for simplicity. Through this classification, it could comprehensively consider the impact of different brightness and weather conditions on the driving scene to better simulate the diversity and complexity of actual driving scenes. Each class in C10CD contained 400 images, making a total of 4000 images in the dataset. Such a sample size ensured the adequacy and representativeness of the data set, which could effectively support the training and evaluation of the driving scene classification network.

The images were preprocessed as shown in Table 2 before being input into WBCNet to enhance the

Table 3 WBCNet experimental results on the C10CD dataset[a)]

Arch.	Architecture	A0	A1	B0	B1	B2	B3	C0	C1	C2	C3	mAP	F1
VG	VGG19	97.68	98.81	97.51	62.40	46.03	10.19	90.02	89.33	54.18	11.94	65.81	0.5619
RN	ResNet152	69.24	87.91	72.58	58.38	80.93	81.52	26.63	79.83	40.94	62.13	66.01	0.6083
DN	Darknet53	99.10	99.13	91.14	50.49	99.80	98.94	97.88	98.73	52.02	10.56	79.78	0.6966
EN	EfficientNet	98.78	99.27	99.52	98.02	99.06	99.43	96.81	99.76	99.46	99.65	98.98	0.9548
ST	Swin-T	99.74	99.83	99.24	98.83	86.58	94.58	99.26	97.37	97.16	97.31	96.99	0.9224
RM	ResMLP	99.37	99.68	98.96	99.33	90.78	97.45	92.05	95.40	99.66	99.04	97.17	0.9500
CT	ConvNeXt-T	98.67	99.20	99.13	99.20	93.59	96.13	93.24	95.30	98.99	98.95	97.24	0.9366
CS	ConvNeXt-S	**99.82**	99.83	99.21	99.14	98.16	98.66	93.99	92.32	97.74	97.24	97.61	0.9356
CB	ConvNeXt-B	99.40	99.79	99.53	99.51	97.76	99.41	95.31	95.64	97.48	97.92	98.18	0.9516
Wc	WBCNet (Ours)-concat	99.63	99.78	99.90	99.91	**99.94**	**99.96**	**99.98**	**99.98**	99.91	99.92	**99.89**	**0.9808**
Wm	WBCNet (Ours)-max	98.95	99.64	99.71	99.87	96.50	98.29	99.92	99.94	99.57	99.61	99.20	0.9472
Ws	WBCNet (Ours)-sum	99.69	99.78	99.30	99.62	99.89	99.88	99.93	99.89	**99.98**	**99.99**	99.80	0.9752
Wa	WBCNet (Ours)-avg	99.72	**99.90**	**99.92**	**99.94**	99.84	99.88	99.95	99.96	99.78	99.78	99.87	0.9802

a) The best result in each column is highlighted in bold.

model's robustness and generalization performance. The length and width of the captured images were relatively large, so they were resized to 340×340. A 224×224 image on the resized image was randomly cropped out to meet the input requirements of the network. Other data augmentation and transforming methods were also applied. These included randomly rotating the image within a range of 10° to increase image diversity and changes in perspective. In addition, the images were flipped horizontally with a probability of 50%, further increasing the diversity of the sample and the richness of the dataset. Finally, the images were normalized.

3.2 Experimental results

A series of classification experiments were conducted to test the performance of the network on the C10CD dataset. In the classification experiments of different networks, AP, mAP, and F1 were chosen as comprehensive evaluation metrics to evaluate the excellence of the model. To ensure the reliability of the experimental results, 60% of the dataset was designated as a training set for training and parameter optimization for WBCNet, 20% as a validation set for model selection and hyperparameter adjustment, and the remaining 20% as a test set for evaluating the final performance of the model. In this way, we can optimize the use of the information in the dataset to train and evaluate the network model and ensure the objectivity and reliability of the evaluation results.

Table 3 shows the experimental results of different network models. By comparing various metrics, we can comprehensively understand the classification performance of different network models on the C10CD dataset. In this subsection, we first performed classification performance experiments on the C10CD dataset using other existing network structures, as shown in the first half of Table 3. The APs of these network models were not the same in different classes. Some models performed better in some classes, whereas they performed worse in others. This shows that different network models have different adaptability and advantages for classification tasks in different scenes.

Then, a classification performance test was conducted on our proposed WBCNet on C10CD. Columns A0 to C3 show AP metrics for each class. The experimental results show that WBCNet achieved high AP in all classes except the A0 class. ConvNeXt-S achieved an AP of 99.82 in the A0 class, just 0.10 better than that of WBCNet-avg. This indicates that our model performed well in the classification of various driving scenes. Further analysis of the experimental results reveals that WBCNet with four fusion methods achieved good results using two comprehensive evaluation indicators (mAP and F1). Figure 7 more intuitively shows the comparison of each model in Table 3 in terms of mAP and F1. This shows the effectiveness of our proposed multilevel feature stream and attention mechanism modules in the task of driving scene classification. By taking full advantage of the information from the multilevel feature stream, WBCNet could more accurately classify different classes of driving scenes, thereby improving its overall classification performance.

In the comparison of WBCNet methods, concat had the best performance. The concat method of the four fusion methods achieved an mAP of 99.89 and F1 of 0.9808, which was very competitive among all the tested classification models. In the process of concat, there is no loss of features. This is because concat

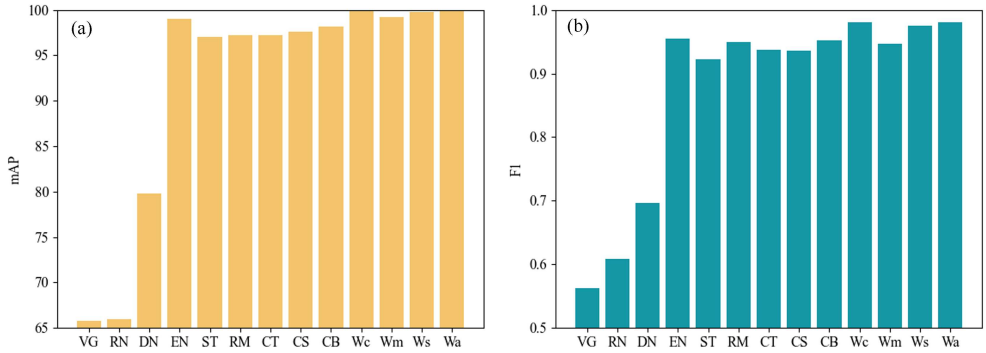

Figure 7 (Color online) Comparison of different models in terms of mAP and F1. The horizontal axis is the abbreviation of the network architectures shown in Table 3. (a) Comparison of different models in terms of the mAP metric; (b) comparison of different models in terms of the F1 metric.

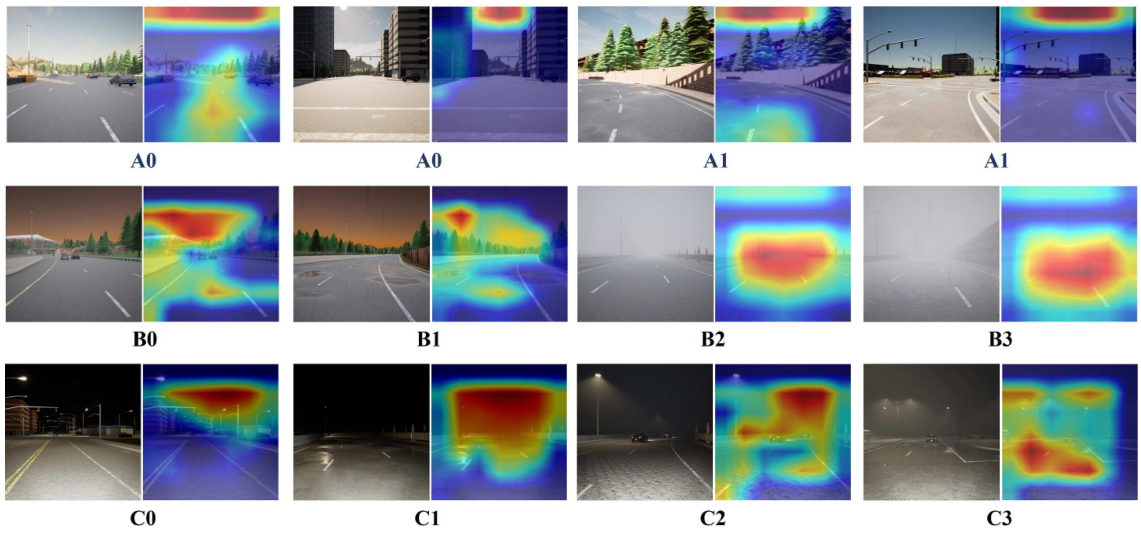

Figure 8 (Color online) WBCNet convolutional layer visualization results. In the Grad-CAM image, red regions denote the areas that WBCNet pays more attention to, while the blue ones denote those WBCNet does not care about.

retains more feature information during the fusion process. Through the validation of this experiment, we can fully affirm the excellent performance of the concat method in driving scene classification tasks and confirm its advantages in WBCNet. This lays the foundation for subsequent experiments. Through careful comparisons of different fusion methods, we obtained insights into the characteristics of the model and chose the fusion method that best suited our task. In the subsequent experiments, WBCNet refers to WBCNet-concat.

3.3 Attention region of WBCNet

This subsection explores what image features WBCNet has learned. Grad-CAM [41] was used to visualize the features learned at the last convolutional layer of WBCNet. The principle was to use the gradient information of the last convolutional layer to analyze the driving scene features learned by the network.

The output results of Grad-CAM could clearly identify the areas of concern of WBCNet. Figure 8 shows the original images of each class and the visualizations passing through Grad-CAM. As shown by the visualization results, WBCNet pays more attention to the sky, the distance, and the road in the image. These three parts are also the main basis for the classification of driving scenes. The sky area can determine whether the scene is currently in bad weather or whether it is in a poor lighting environment. The distance is mainly about visibility. When fog is present, visibility in front of the vehicle is reduced, resulting in poor visibility. Water on the road, reflections, and other details can help WBCNet determine the current weather conditions. In certain scenes, WBCNet may also pay attention to other auxiliary information, such as vehicles and plants. This results in a wider area of focus for WBCNet. These

Table 4 Ablation experiments on the C10CD dataset with different modules

Model	Architecture	mAP	P	R	Acc	F1
A	Baseline	99.58	96.37	96.26	96.25	0.9628
B	Baseline + MCM	+ 0.11	+ 0.05	+ 0.15	+ 0.13	+ 0.0011
C	Baseline + RCA	+ 0.10	+ 0.62	+ 0.79	+ 0.75	+ 0.0072
D	Baseline + RMDC	+ 0.10	+ 0.55	+ 0.66	+ 0.63	+ 0.0064
E	Baseline + RCA + MCM	+ 0.12	+ 0.90	+ 1.05	+ 1.00	+ 0.0098
F	Baseline + RMDC + MCM	+ 0.26	+ 1.29	+ 1.36	+ 1.38	+ 0.0133
G	Baseline + RMDC + RCA	+ 0.22	+ 1.33	+ 1.48	+ 1.38	+ 0.0143
H	Baseline + RMDC + RCA + MCM (WBCNet)	+ 0.31	+ 1.72	+ 1.82	+ 1.75	+ 0.0180

visualization results provide important clues for us to explain the classification performance of WBCNet and help us further optimize network design and improve algorithms.

3.4 Ablation experiments

The proposed plug-and-play modules MCM, RCA, and RMDC were used in WBCNet. The experiments in this subsection verified how much the above three modules contributed to the classification task. Eight models, named Models A to H, respectively, were established to complete the experiment. The impact of each module on classification tasks was evaluated by training and testing these eight models on the C10CD dataset. By comparing the experimental results, we could quantitatively analyze the effectiveness of each module in improving classification performance.

In the ablation experiment, mAP, P, R, Acc, and F1 were selected as metrics to test the function of MCM, RCA, and RMDC. As shown in Table 4, all three modules were removed to get the baseline model, which was Model A. Then MCM, RCA, and RMDC modules were added respectively on the basis of Model A to get Models B, C, and D. This verified the performance of individual modules. Then, the contributions of the combination of modules to the network were explored. Models E, F, and G were obtained by combining the three modules. The three modules were combined to get Model H, which was our proposed WBCNet model. Through this series of experiments, we can gradually analyze the impact of the independent and combined effects of each module on network performance. Meanwhile, by comparing the experimental results of Model H with those of other models, the effectiveness and superiority of WBCNet were verified.

Figure 9 compares the eight models in terms of each metric. The dotted lines of each subplot represent the value of a certain metric in the baseline model. If the values of the other models exceeded this dotted line, it represented an improvement over the baseline model. Table 4 shows the metrics for the baseline model and the increments for the seven improved models. Combined with the results in Figure 9 and Table 4, in the comparison experiment of single modules, MCM, RCA, and RMDC all had an effect on improving the baseline performance. Among them, MCM had the greatest impact on the mAP of the baseline, whereas RCA had the greatest impact on the F1 of the baseline. In the experiment where two modules were combined, RMDC improved the baseline even more. Models E and F performed similarly, but model D lagged behind them. After synthesizing the three modules, Model H reached the highest level of performance. The comprehensive experimental results show that our WBCNet model had excellent performance in each evaluation metric. At the same time, the ablation experiment also confirmed the effectiveness of the MCM, RCA, and RMDC modules, and their combined use further improved the network performance.

3.5 Object detection test

Figure 10 shows the object detection results of the YOLO [26] algorithm in two different scenes. Figure 10(a) shows visible precise object detection results under clear weather. Meanwhile, the detection accuracies in other driving scenes were not ideal. For example, there was an undetected vehicle in the red dotted box in Figure 10(b).

WBCNet was designed to classify as many driving scenes as possible, even though subsequent perceptions may combine similar scenes. Some perception algorithms are needed, which can include single-sensor and multisensor fusion perception algorithms. These algorithms make up the perceptual algorithm set $\mathcal{D}$. One scene may correspond to one algorithm, or multiple scenes may correspond to one algorithm in $\mathcal{D}$. Then, the corresponding perception algorithm can be selected according to the output of the network.

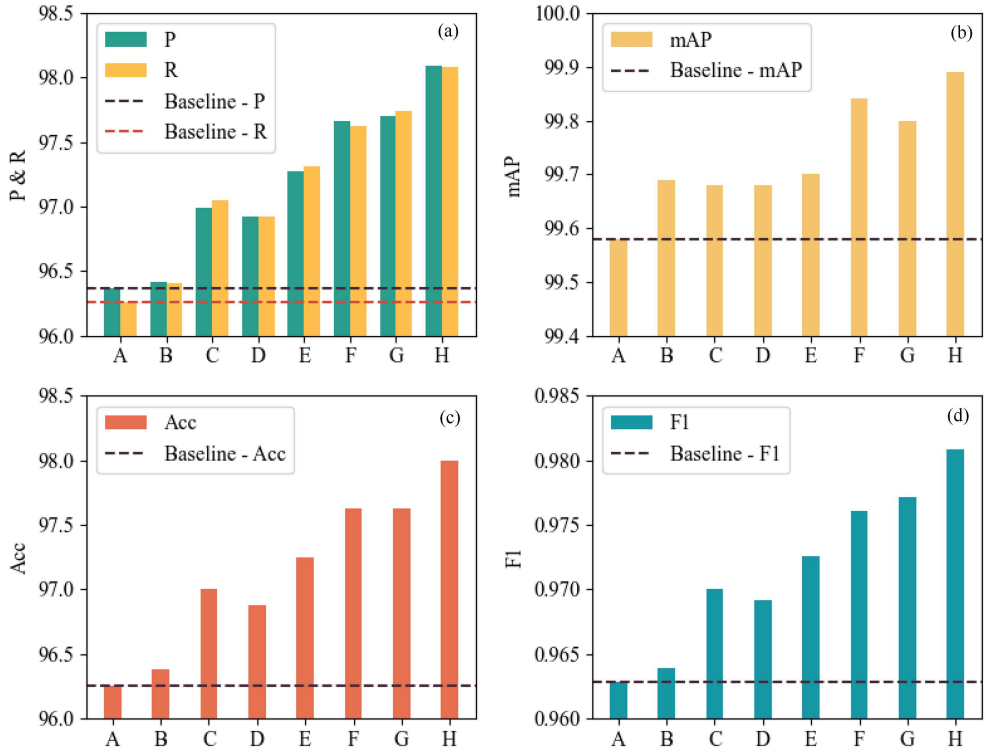

Figure 9 (Color online) Comparison of different modules in the ablation experiment. The horizontal axis is the abbreviation of the network architecture shown in Table 4. (a) Experimental results from Model A to Model H in terms of the P and R metrics; (b) experimental results from Model A to Model H in terms of the mAP metric; (c) experimental results from Model A to Model H in terms of the Acc metric; (d) experimental results from Model A to Model H in terms of the F1 metric.

Figure 10 (Color online) Examples of the same detection algorithm detecting failures in different scenes. (a) Detection results in a clear day; (b) detection results in a rainy and foggy night.

For IVs, perception algorithms are important. They extract useful information from raw data collected by various sensors, such as cameras, radars, and lidars. Then, related perceptual tasks, such as target detection and driving area division, are conducted. Therefore, constructing an effective set of perception algorithms $\mathcal{D}$ is crucial for achieving reliable autonomous driving.

Algorithm 1 is a preliminary demonstration algorithm because it only includes the processing of three

Algorithm 1 Framework for object detection according to WBCNet (Only B2, C0, and C3 are included)

Input: Image data I captured by the camera; time constant τ; total set of perceptual models $\mathcal{D}$;

Load WBCNet weights;
Initialize the vehicle driving environment E = A0;
Set the program run time $T = 0$;
while $I \neq \emptyset$ **do**
  if T % τ == 0 **then**
    Input image I to WBCNet;
    Update E based on WBCNet output;
  end if
  if E == B2 **then**
    Perform dehazing on image I;
  else if E == C0 **then**
    Perform low-light enhancement on image I;
  else if E == C3 **then**
    Perform dehazing and low-light enhancement on image I;
  end if
  Input image I into the object detection network;
end while

Output: Object detection results.

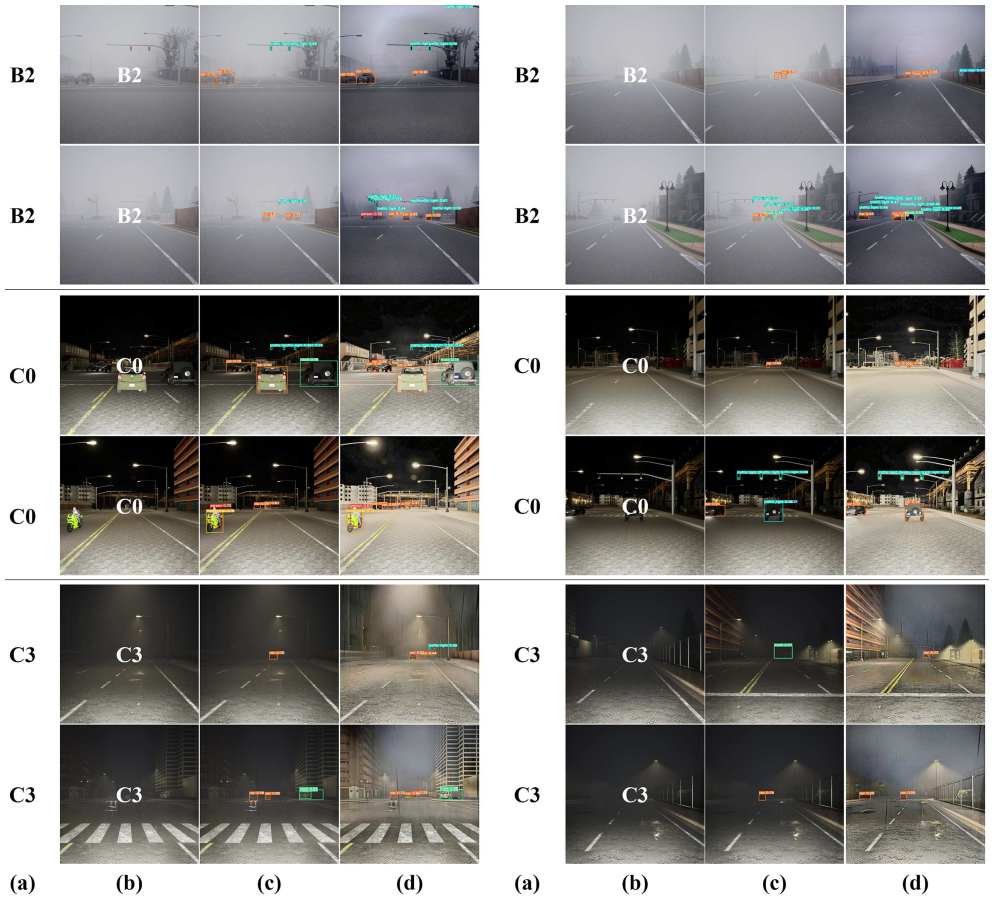

Figure 11 (Color online) Comparision of object detection results. (a) Ground truth label; (b) scene image and label predicted by WBCNet; (c) object detection results directly from the original image; (d) object detection results after the image is predicted by WBCNet and processed according to Algorithm 1.

scenes, and the processing algorithm is very simple. In this test, we simply design $\mathcal{D}$ as shown below:

$$\begin{aligned}\mathcal{D} = \{&\text{B2:MSBDN [42]};\\ &\text{C0:Bread [43]};\\ &\text{C3:MSBDN + Bread}\}.\end{aligned} \tag{10}$$

Figure 11 shows a comparison of object detection results. Three important scenes serve as a demonstration of the necessity of WBCNet's existence. B2, C0, and C3 are common scenes in daily life. Each

Table 5 Real-time performance analysis and experimental results on a real-world driving dataset

Architecture	FLOPs (G)	Params (M)	GPU inference time (ms)	CPU inference time (ms)	Acc	F1
ResNet	6.61	38.10	0.950	127.96	53.46	0.2979
Darknet	7.15	40.60	0.849	112.00	67.78	0.4691
EfficientNet	5.34	63.81	3.120	295.83	68.52	0.5096
ResMLP	3.01	14.94	0.398	36.80	73.33	0.6109
ConvNeXt-T	4.46	27.82	0.746	57.05	69.07	0.5492
ConvNeXt-S	8.70	49.45	1.253	96.84	72.78	0.5808
ConvNeXt-B	15.37	87.55	1.937	153.35	74.07	0.6041
WBCNet	11.93	39.47	1.315	112.75	85.25	0.8292

scene is illustrated with four sets of results. Visibility is reduced because of heavy rain, fog, and nighttime conditions. At this time, if the corresponding perception method is not used, it leads to missing detection and false detection. By comparing columns (c) and (d) of the same image, we can see that driving scenes must be classified because column (c) is more likely to have missed or false detection than column (d). This section briefly validates the need for WBCNet. $\mathcal{D}$ in real life is not so simple. It may contain multiple sensors working together for perception.

3.6 Real-time performance and test on real-world dataset

In this subsection, the performance and real-time capabilities of WBCNet were tested on a real-world driving dataset. BDD100K is a real-world dataset containing various weather and driving scenes [44]. A dataset comprising 8100 images, twice the size of C10CD, was built from BDD100K for testing. The dataset consisted of nine classes: clear day, clear dusk, clear night, foggy day, overcast day, rainy day, rainy dusk, rainy night, and snowy day. The dataset was divided into training, validation, and testing sets in a ratio of 6:2:2. Floating point operations (FLOPs) are commonly used to measure the computational complexity of a computer program or algorithm. FLOPs usually refer to the total number of FLOPs performed by a neural network model during inference or training. Model parameters represent the number of adjustable parameters to be learned in the model. The more model parameters there are, the larger the capacity of the model and the higher the complexity that can be represented. However, an excessive number of parameters may also lead to overfitting and waste of computational resources. The inference time represents the time required for the model to perform inference on the given input data. The shorter the inference time, the better the real-time performance of the model. The model parameters and inference time represent the complexity and efficiency of the model. FLOPs, model parameters, and inference time were selected as metrics for real-time performance. The real-time performance and classification performance of WBCNet and other models using the real-world driving dataset are shown in Table 5. Apparently, WBCNet had a moderate number of parameters from Table 5. The RTX 3090 graphics card (GPU) was used to test the inference time of the models. The classification process of an image took only 1.315 ms using GPU in WBCNet. The time met the real-time performance requirements of IVs. WBCNet achieved the leading accuracy of 85.25% while maintaining real-time performance using the dataset comprising real-world driving scenes. We also conducted tests on low-power platforms, such as an i7-10700K central processing unit (CPU), and the results showed an increase in inference time. Considering that the CPU has limited computing power in image processing, the inference speed can be further improved when deploying it on a vehicle chip, such as a mass-produced Qualcomm Snapdragon 8295 (SA8295) automotive processor.

4 Conclusion

This paper proposes WBCNet as an innovative solution to overcome the challenge of reduced perception accuracy resulting from the variability of driving scenes. First, WBCNet combines the outstanding feature association capability of attention mechanisms with the excellent multilevel feature fusion capability of DCs. Because of the integration of the aforementioned specially designed modules, WBCNet can accept images as input, making it more suitable for engineering applications. This helps mitigate issues related to insufficient computational capacity on vehicle chips. To verify the performance of the WBCNet network, a dataset based on the Carla simulator is constructed. Furthermore, we conduct ablation experiments

on three proposed modules. By adding and removing these modules, we demonstrate that all three modules are crucial for WBCNet to achieve its leading level of performance. We validate the necessity and effectiveness of WBCNet by designing a simple object detection algorithm library. The detection results show that, following precise scene recognition by WBCNet, selecting the appropriate perception algorithm based on the scene can significantly enhance detection accuracy. WBCNet is compared with other classification models, and the results demonstrate that WBCNet achieved a leading level. Finally, the real-time performance and generalization capabilities of WBCNet are tested using a real-world driving dataset. The results demonstrate that WBCNet achieves more accurate classification of driving scenes in real time compared with other models.

Acknowledgements This work was supported by National Nature Science Foundation of China (Grant No. 52472430) and Jilin Province Science and Technology Plan Program (Grant No. 20230201123GX).

References

1 Chen H, Yuan K, Huang Y J, et al. Feedback is all you need: from ChatGPT to autonomous driving. Sci China Inf Sci, 2023, 66: 166201
2 Cao D, Wang X, Li L, et al. Future directions of intelligent vehicles: potentials, possibilities, and perspectives. IEEE Trans Intell Veh, 2022, 7: 7–10
3 Gao H, Zhu J, Zhang T, et al. Situational assessment for intelligent vehicles based on stochastic model and Gaussian distributions in typical traffic scenarios. IEEE Trans Syst Man Cybern Syst, 2022, 52: 1426–1436
4 Chen S T, Jian Z Q, Huang Y H, et al. Autonomous driving: cognitive construction and situation understanding. Sci China Inf Sci, 2019, 62: 081101
5 Long T, Liang Z N, Liu Q H. Advanced technology of high-resolution radar: target detection, tracking, imaging, and recognition. Sci China Inf Sci, 2019, 62: 040301
6 He Z, Chen Y, Zhang H, et al. WKN-OC: a new deep learning method for anomaly detection in intelligent vehicles. IEEE Trans Intell Veh, 2023, 8: 2162–2172
7 Wang K, Zhou T, Li X, et al. Performance and challenges of 3D object detection methods in complex scenes for autonomous driving. IEEE Trans Intell Veh, 2023, 8: 1699–1716
8 Cui Q M, Hu X X, Ni W, et al. Vehicular mobility patterns and their applications to internet-of-vehicles: a comprehensive survey. Sci China Inf Sci, 2022, 65: 211301
9 Zhang S, Li X, Zong M, et al. Learning k for KNN classification. ACM Trans Intell Syst Technol, 2017, 8: 1–19
10 Taunk K, De S, Verma S, et al. A brief review of nearest neighbor algorithm for learning and classification. In: Proceedings of the International Conference on Intelligent Computing and Control Systems (ICCS), 2019. 1255–1260
11 Vens C, Struyf J, Schietgat L, et al. Decision trees for hierarchical multi-label classification. Mach Learn, 2008, 73: 185–214
12 Charbuty B, Abdulazeez A. Classification based on decision tree algorithm for machine learning. J Appl Sci Technol Trends, 2021, 2: 20–28
13 Zhang Y L. Support vector machine classification algorithm and its application. In: Proceedings of the International Conference on Information Computing and Applications (ICICA), 2012. 179–186
14 Wang Z W, Zhang C L, Su C, et al. On modeling of atmospheric visibility classification forecast with nonlinear support vector machine. In: Proceedings of the 5th International Conference on Natural Computation, 2009. 240–244
15 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. ArXiv:1409.1556
16 He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 770–778
17 Jung H, Choi M K, Jung J, et al. ResNet-based vehicle classification and localization in traffic surveillance systems. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 61–67
18 Rezende E, Ruppert G, Carvalho T, et al. Malicious software classification using transfer learning of ResNet-50 deep neural network. In: Proceedings of the 16th IEEE International Conference on Machine Learning and Applications (ICMLA), 2017. 1011–1014
19 Lu Z, Bai Y, Chen Y, et al. The classification of gliomas based on a pyramid dilated convolution resnet model. Pattern Recogn Lett, 2020, 133: 173–179
20 Lin D, Lu C, Huang H, et al. RSCM: region selection and concurrency model for multi-class weather recognition. IEEE Trans Image Process, 2017, 26: 4154–4167
21 Shi Y Z, Li Y X, Liu J W, et al. Weather recognition based on edge deterioration and convolutional neural networks. In: Proceedings of the 24th International Conference on Pattern Recognition (ICPR), 2018. 2438–2443
22 He K M, Gkioxari G, Dollr P, et al. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017. 2980–2988
23 Kukreja V, Solanki V, Baliyan A, et al. WeatherNet: transfer learning-based weather recognition model. In: Proceedings of the International Conference on Emerging Smart Computing and Informatics (ESCI), 2022. 1–5
24 Xie K, Huang L, Zhang W, et al. A CNN-based multi-task framework for weather recognition with multi-scale weather cues. Expert Syst Appl, 2022, 198: 116689
25 Dalal S, Seth B, Radulescu M, et al. Optimized deep learning with learning without forgetting (LwF) for weather classification for sustainable transportation and traffic safety. Sustainability, 2023, 15: 6070
26 Jocher G, Stoken A, Borovec J, et al. Ultralytics/yolov5: v3.1 — bug fixes and performance improvements. Zenodo, 2020
27 Tan M X, Le Q V. EfficientNet: rethinking model scaling for convolutional neural networks. 2019. ArXiv:1905.11946
28 Liu Z, Lin Y T, Cao Y, et al. Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2021. 9992–10002
29 Touvron H, Bojanowski P, Caron M, et al. ResMLP: feedforward networks for image classification with data-efficient training. 2021. ArXiv:2105.03404
30 Liu Z, Mao H Z, Wu C Y, et al. A ConvNet for the 2020s. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 11966–11976

31 Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell, 2017, 39: 1137–1149
32 Redmon J, Farhadi A. Yolov3: an incremental improvement. 2018. ArXiv:1804.02767
33 Chen Y L, Wu B F, Huang H Y, et al. A real-time vision system for nighttime vehicle detection and traffic surveillance. IEEE Trans Ind Electron, 2011, 58: 2030–2044
34 Huang S C, Hoang Q V, Jaw D W. Self-adaptive feature transformation networks for object detection in low luminance images. ACM Trans Intell Syst Technol, 2022, 13: 1–11
35 Huang S C, Jaw D W, Hoang Q V, et al. 3FL-Net: an efficient approach for improving performance of lightweight detectors in rainy weather conditions. IEEE Trans Intell Transp Syst, 2023, 24: 4293–4305
36 Liu W Y, Ren G F, Yu R S, et al. Image-adaptive YOLO for object detection in adverse weather conditions. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2022. 1792–1800
37 Wang L, Qin H, Zhou X, et al. R-YOLO: a robust object detector in adverse weather. IEEE Trans Instrum Meas, 2022, 72: 1–11
38 Hu J, Shen L, Albanie S, et al. Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 7132–7141
39 Li X, Wang W H, Hu X L, et al. Selective kernel networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 510–519
40 Woo S, Park J, Lee J Y, et al. CBAM: convolutional block attention module. 2018. ArXiv:1807.06521
41 Ramprasaath R, Selvaraju, Cogswell M. Grad-CAM: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017. 618–626
42 Dong H, Pan J S, Xiang L, et al. Multi-scale boosted dehazing network with dense feature fusion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 2154–2164
43 Guo X, Hu Q. Low-light image enhancement via breaking down the darkness. Int J Comput Vis, 2023, 131: 48–66
44 Yu F, Chen H F, Wang X, et al. BDD100K: a diverse driving dataset for heterogeneous multitask learning. 2018. ArXiv:1805.04687