Dual-aggregation feature compilation network for urban traffic object detection and pedestrian pose estimation

Huang Xiao[1], Hanqing Jian[2]
[1, 2]School of Traffic Management and Engineering, Guangxi Police College, Nanning, China
[2]School of Mechanical Engineering, Guangxi University, Nanning, China
[2]Corresponding author
E-mail: [1]*gxhuangxiao@126.com*, [2]*gxpcjianhanqing@163.com*

Received 7 April 2025; accepted 17 July 2025; published online 9 August 2025
DOI https://doi.org/10.21595/jme.2025.24953

Abstract. With the increasing complexity of urban transportation systems, object detection and pedestrian pose estimation play a crucial role in intelligent traffic management and autonomous driving technologies. However, existing feature compilation networks are often designed for single tasks and perform poorly in small object detection and high occlusion pedestrian pose estimation tasks. To address the above issues, this paper proposes an efficient feature compilation network with Dual-aggregation, compatible with both object detection and pedestrian pose estimation. This network adopts a transfer learning-like training strategy in the feature extraction network, using a micro-complex convolution structure during training to bring the training results as close as possible to global optimization. During inference, a single simple convolution is used to inherit the training results, improving the model performance while ensuring model lightweight. The feature fusion employs a global-local dual aggregation structure, simultaneously considering multi-scale global and local features. Additionally, we use multiple public datasets to create a hybrid dataset under various scenarios to validate the robustness of the network. The experiments show that the proposed method outperforms existing mainstream methods in detection accuracy for urban object detection and pedestrian pose estimation tasks, especially demonstrating better robustness in complex urban traffic scenarios.

Keywords: urban traffic, deep learning, object detection, pose estimation, vision transformer.

1. Introduction

With the rapid development of urbanization and the increasing demand for intelligent transportation systems (ITS), robust and efficient object detection and pedestrian pose estimation have become crucial for ensuring traffic safety and optimizing transportation management. Object detection algorithms play a vital role in identifying vehicles, pedestrians, and other traffic entities, while pose estimation enhances the understanding of human behavior and interactions in urban scenarios. Together, these technologies form the backbone of advanced driver-assistance systems (ADAS) and autonomous driving applications.

Currently, most mainstream approaches for urban traffic object detection and pedestrian pose estimation rely on deep learning [1-4]. These methods extract relevant features from images using feature compilation networks, which are then fed into different downstream detectors for either object detection or pose estimation. Therefore, an efficient and compatible feature compilation network plays a vital role in both object detection and pedestrian pose estimation tasks. The feature compilation network consists of a feature extraction network and a feature fusion network. In the feature extraction component, Krizhevsky et al. [5] were the first to propose using convolution for feature extraction. However, this approach is limited to capturing local features and lacks the ability to model long-range dependencies. Studies [6, 7] enhanced feature extraction performance by increasing network depth or introducing highly complex architectures. However, these approaches often overlooked the importance of lightweight design, resulting in excessive computational redundancy. Research efforts in [8, 9] focused on designing lightweight convolutional structures to reduce computational cost and model size. Nevertheless, these methods

often compromised accuracy in pursuit of efficiency, performing poorly in complex tasks. Studies [10-12] have explored combining Transformer architectures with convolutional networks to enable more efficient feature extraction. However, these hybrid methods require more intricate hyperparameter tuning, and often underperform traditional convolutional models on small-scale datasets. In the feature fusion component, Lin et al. [13] were the first to propose using a feature pyramid structure to achieve multi-scale feature fusion. This top-down fusion approach significantly improves the detection of small objects. However, during the fusion process, high-level features may overwhelm low-level detail information. Liu et al. [14] introduced a bottom-up path aggregation strategy, known as the Path Aggregation Network (PAN), to supplement missing low-level details. Nonetheless, this method handles redundant information inadequately, leading to suboptimal fusion efficiency. Ghiasi et al. [15] leveraged Neural Architecture Search (NAS) to automatically design an optimal feature pyramid structure. However, this approach requires significant computational resources during training, making rapid deployment challenging. Chen et al. [16] proposed a bidirectional feature pyramid fusion strategy based on PAN, achieving efficient multi-scale fusion with a more lightweight architecture. Although weighted fusion improves performance, it introduces additional computational overhead. Studies [17-19] enhanced feature fusion effectiveness by employing attention mechanisms and dynamically assigning weights to emphasize critical features. However, Transformer-based approaches often struggle to effectively capture local features.

In response to the specific feature requirements of urban traffic object detection and pedestrian pose estimation tasks, Li et al. [20] integrated spatial and channel attention mechanisms within the feature compilation network to enhance the network's focus on critical spatial locations and feature channels in the domain of object detection. They also employed dilated convolutions to expand the receptive field. However, the efficiency of attention mechanisms and dilated convolutions remains limited, particularly under complex environmental conditions, resulting in insufficient robustness. Han et al. [21] proposed a lightweight symmetric data fusion network, Epurate-Net, which merges spatial responses into visual features and aggregates contextual information to adaptively refine road contours. This approach improves the delineation accuracy of road boundaries. Nevertheless, it exhibits limitations in feature representation and generalization during boundary optimization and context aggregation. Furthermore, balancing lightweight design and model complexity remains a challenge. In pedestrian pose estimation tasks, Li et al. [22] adopted a Transformer-based approach, utilizing a self-attention mechanism to label each keypoint and learn the constraints among them within the image. Despite its effectiveness, the Transformer-based method suffers from high computational complexity, increased memory usage, and longer inference time due to the large number of parameters.

In summary, object detection and pedestrian pose estimation tasks place different emphases on image feature processing. Object detection focuses more on the fusion of multi-scale information and global contextual cues to facilitate the recognition of diverse objects and the complex relationships between scenes. In contrast, pedestrian pose estimation emphasizes fine-grained local features and the precise localization of keypoints. As a result, current feature compilation networks struggle to balance the distinct feature requirements of these two tasks. To address the aforementioned issues, this paper proposes a Dual-Aggregation Feature Compilation Network (DAFCN) based on YOLOv8s, which is suitable for both urban traffic object detection and pedestrian pose estimation. DAFCN comprises a Multi-Branch Feature Learning Network (MBLNet) and a multidimensional spatial feature aggregation network (MS-FAN). MBLNet employs a micro-complex convolution structure during training to bring the model's convergence closer to global optimization. During inference, a single simple convolution inherits the trained parameters, achieving more accurate feature extraction in a lightweight manner. The feature extraction network effectively reduces model complexity through a class-transfer training strategy, thereby accelerating feature compilation to meet the demands of real-time systems. The MS-FAN incorporates our proposed global and local aggregation module (GLAM), which focuses on global and local features of the multi-scale features extracted by MBLNet and filters out noisy

features through a parallel fusion strategy, achieving the integration of global and local image features.

The main contributions of the paper are as follows:

(1) An efficient feature compilation network architecture, DAFCN, is proposed, which balances multi-scale, global, and local detailed features. It demonstrates excellent performance in both urban traffic object detection and pedestrian pose estimation tasks.

(2) A Multi-Branch Feature Learning Network (MBLNet) is proposed for the feature extraction network. MBLNet adopts a strategy of complex training and lightweight inference, maximizing feature extraction performance while ensuring network efficiency.

(3) A multidimensional spatial feature aggregation network (MS-FAN) for feature fusion is proposed, which utilizes the Global and Local Aggregation Module (GLAM) to balance the global and local information within the extracted multi-scale features. A parallel fusion strategy is employed to achieve efficient feature fusion.

2. Materials and methods

2.1. Overview of DAFCN

The state-of-the-art feature compilation architecture, the YOLO series, achieves high feature compilation performance with relatively low parameter and computational costs. However, its performance in detecting distant small objects in object detection tasks is suboptimal. In action recognition tasks, it struggles to accurately estimate the pose of occluded pedestrians. Therefore, in this paper, DAFCN is developed based on the feature compilation architecture of YOLOv8s. By designing a feature input strategy and a global-local feature aggregation module, and optimizing the overall network structure, it effectively achieves multi-scale feature compilation, as well as the processing of detailed texture features and the capture of global semantic information.

The overall structure of DAFCN is shown in Fig. 1. DAFCN consists of a feature extraction network named MBLNet and a feature fusion network named MS-FAN. In MBLNet, for the input RGB image, it is first converted into a grayscale image with 64 channels using a convolution layer. Then, a four-stage multi-scale feature extraction architecture is applied, resulting in features with channel dimensions of 128, 256, 512, and 512, respectively. The three lowest-resolution features, namely those with channel dimensions of 256, 512, and 512, are fed into the feature fusion network. The detailed structure of MBLNet will be elaborated in Section 3.2 of the paper.

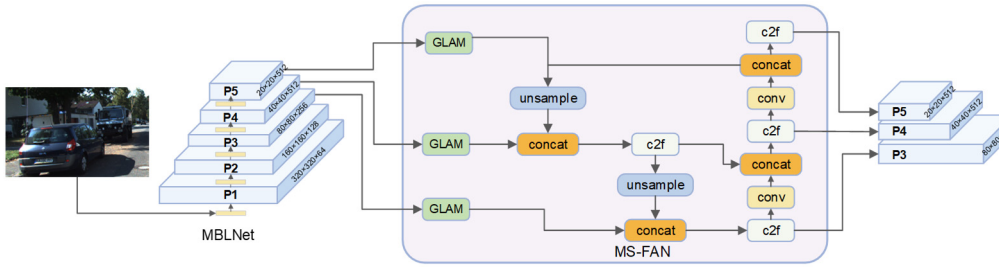


Fig. 1. The overall structure of DAFCN

In MS-FAN, the three input feature layers are first aggregated for their global and local features using the GLAM module. Then, the Feature Pyramid Network (FPN) structure and the Path Aggregation Network (PAN) structure are used, respectively, to perform a second round of feature fusion, from bottom to top and top to bottom, for the three feature layers. Finally, three fused feature maps with channel dimensions of 256, 512, and 512 are output. The detailed structure of MS-FAN will be elaborated in Section 3.3 of the paper.

2.2. Multi-branch feature learning network

The multi-branch feature learning network, as shown in Fig. 2(a), consists of four feature learning stages with varying dimensions.

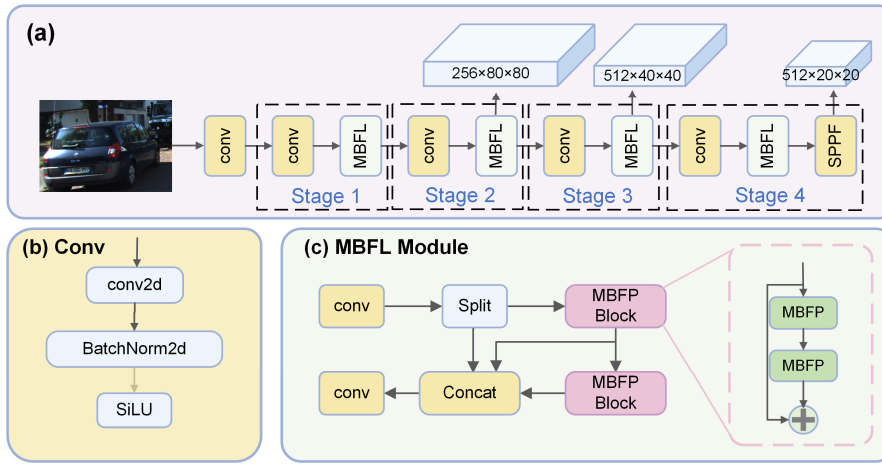


Fig. 2. a) Structure of multi-branch feature learning network,
b) Conv operation, c) multi-branch feature learning (MBFL) module

In each stage, a Conv operation is first applied to downsample the feature map. Then, a multi-branch feature learning (MBFL) module is employed to learn diverse feature representations or perform specific tasks. The Conv operation, as shown in Fig. 2(b), consists of a 3×3 convolution, a Batch Normalization layer, and a SiLU activation function. The specific calculation method is as follows:

$$Conv(\bullet) = SiLU\left(BN\left(conv2d(\bullet)\right)\right), \tag{1}$$

where BN refers to the Batch Normalization operation, $conv2d$denotes the two-dimensional convolution operation.

The MBFL structure, as shown in Fig. 2(c), adopts a residual network architecture. for the input feature f_i. It is first split into f_i^1 and f_i^2 along the channel dimension. The f_i^2 is then fed into the multi-branch feature processing block (MBFP) for feature compilation. Finally, the features are concatenated and processed with a Conv operation to obtain the output O_i. The specific calculation method is as follows:

$$\begin{aligned} &f_i^1, f_i^2 = Split\left(Conv(f_i)\right), \\ &O_i = Conv(Concat(f_i^1, MBFP(f_i^2), MBFP\left(MBPF(f_i^2)\right), \end{aligned} \tag{2}$$

where each MBFP Block consists of two layers of multi-branch feature learning operations, connected in series via a residual structure, as illustrated in Fig. 2(c). Each multi-branch feature learning (MBFP) operation, shown in Fig. 3, adopts a multi-branch training and single-path inference strategy to enhance testing performance while reducing model complexity.

We generally aim for the model to achieve global optimization during training to ensure maximum effectiveness during inference. Simultaneously, we desire the model to be lightweight enough to enhance inference speed. Inspired by study [23], MBFP introduces additional convolution types during the training phase to bring the training performance closer to the global optimum. During inference, it employs only a single $k \times k$ convolution to inherit the training

ISSN PRINT 2335-2124, ISSN ONLINE 2424-4635

parameters, thereby reducing inference complexity while improving training performance. As shown in Fig. 3(a), MBFP utilizes six types of convolutions during training, namely conv-BN, sequential convolutions, average pooling and three types of multi-scale convolutions. During inference, all these convolutions can be replaced by a single $k \times k$ convolution. The underlying principle is as follows:

(1) Conv-BN.

For conv-BN, a convolution is typically followed by a batch normalization operation. For an input I and a convolution operation F, the computation process is as follows:

$$O_j = \frac{(I * F)_j - \mu_j}{\delta_j}\gamma_j + \beta_j, \tag{3}$$

where $*$ represents the standard convolution operation. μ_j and δ_j denote the normalization mean and standard deviation for the j channel, while γ_j and β_j represent the batch normalization scaling factor and bias. By merging these, the convolution kernel F and BN parameters are recalculated into F' and b', allowing the BN layer to be removed during inference while achieving the same functionality with a single convolution. The specific computation method is as follows:

$$F'_j = \frac{\gamma_j}{\delta_j}F_j, \quad b'_j = -\frac{\mu_j\gamma_j}{\delta_j} + \beta_j. \tag{4}$$

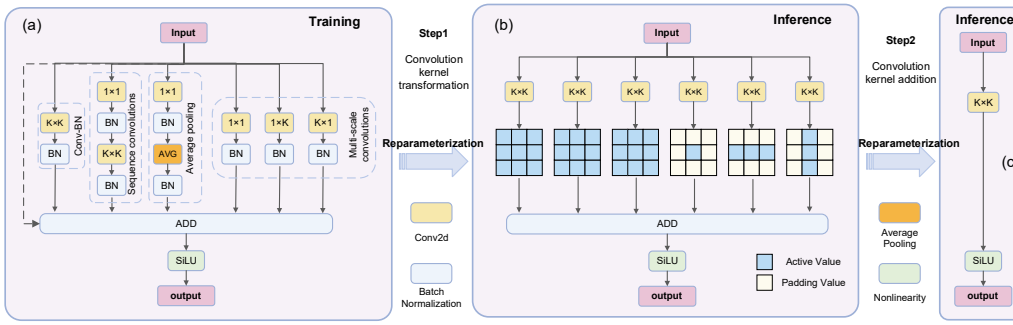


Fig. 3. Multi-branch feature processing operation (MBFP): a) the MBFP structure during training, b) equivalent substitution diagram of each branch, c) MBFP structure during inference

(2) Sequential convolutions.

The sequential convolutions structure consists of a 1×1 convolution kernel for channel adjustment and a $K \times K$ convolution kernel for feature extraction. The computation process is as follows:

$$O = (I * F_1 + \mathrm{b}_1) * F_2 + b_2, \tag{5}$$

where F_1, F_2, b_1, and b_2 represent the 1×1 convolution, $K \times K$ convolution, and their respective biases. The combined effect of these two convolution layers can be represented by an equivalent $K \times K$ convolution kernel and bias. The calculation methods for the new convolution kernel F' and bias b' are as follows:

$$F' = F_2 * TRANS(F_1), \quad b' = b_1 * F_2 + b_2, \tag{6}$$

where $TRANS(F_1)$ represents the transposed version of the 1×1 convolution kernel, used for channel combination.

(3) Average pooling.

The average pooling operation is equivalent to a special convolution. When the pooling kernel

is K, the replaced convolution kernel becomes a smoothed version of the identity matrix, represented by $1/K^2$ for averaging. This allows average pooling operations to be replaced with convolutions, unifying the computational framework as follows:

$$F'_{d,c,u,v} = \begin{cases} \frac{1}{K^2}, & d = c, \\ 0, & d \neq c, \end{cases} \tag{7}$$

where d and c represent the output and input channel indices, while u and v denote the row and column indices of the convolution kernel.

(4) Multi-scale convolutions.

For multi-scale convolutions, smaller kernels can be expanded into larger ones via zero padding, enabling unified representation of multi-scale features. The calculation for the new convolution kernel is as follows:

$$F'_{d,c,u,v} = \begin{cases} F_{d,c,u,v}, & u < k_h, \quad v < k_w, \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

where k_h and k_w represents the height and width of the original convolution kernel.

During inference, as shown in Fig. 3(b), the series of convolutions in the six branches can all be replaced with $K \times K$convolutions. Since multiple $K \times K$ convolutions in parallel are equivalent to a single-channel $K \times K$ convolution, the final inference adopts the $K \times K$ Conv-SiLU structure shown in Fig. 3(c).

2.3. Multidimensional spatial feature aggregation network

As shown in Fig. 4, the multidimensional spatial feature aggregation network adopts an FPN-PAN feature fusion architecture. For the three smallest-resolution feature layers input by the feature extraction network, the Global and Local Aggregation Module (GLAM) is applied to fuse their global and local information. The fused three feature layers are then further processed for multi-scale fusion using a bottom-up Feature Pyramid Network (FPN) [13] structure. To prevent information loss during the FPN cascading process and to retain more detailed information, a Path Aggregation Network (PAN) [14] structure is employed to cascade and concatenate the features in a top-down manner.

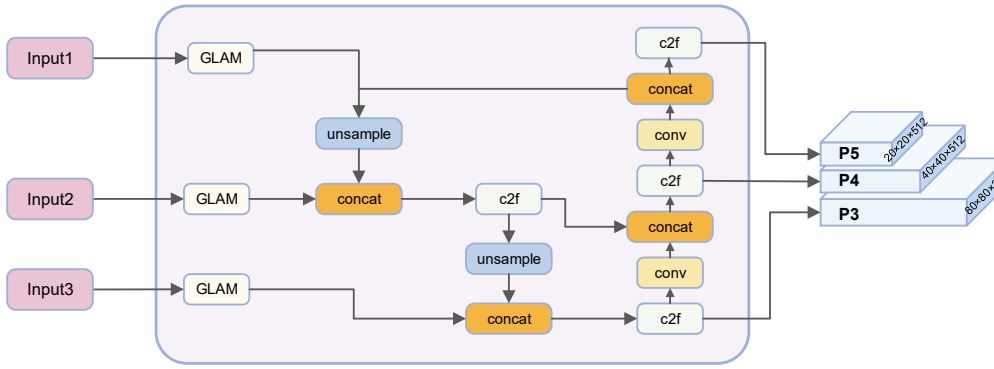


Fig. 4. Multidimensional spatial feature aggregation network (MS-FAN)

The GLAM module, as shown in Fig. 5, divides the input feature $X \in R^{C \times H \times W}$ evenly along the channel dimension into sub-features X_1, $X_2 \in R^{\frac{C}{2} \times H \times W}$. These sub-features are then fed into the global feature extraction module (GFE) and local feature extraction module (LFE),

respectively, for feature compilation. The compiled features X_1', $X_2' \in R^{\frac{C}{2}\times H\times W}$ are obtained. They are concatenated and further fused using a 1×1 convolution. The specific calculation method is as follows:

$$X_1, X_2 = Split(X), \tag{9}$$
$$Y = Conv(Concat(GFE(X_1), LFE(X_2))), \tag{10}$$

where Y represents the output features after global and local aggregation. $GFE()$ and $LFE()$ refer to the global feature extraction module and local feature extraction module, respectively.

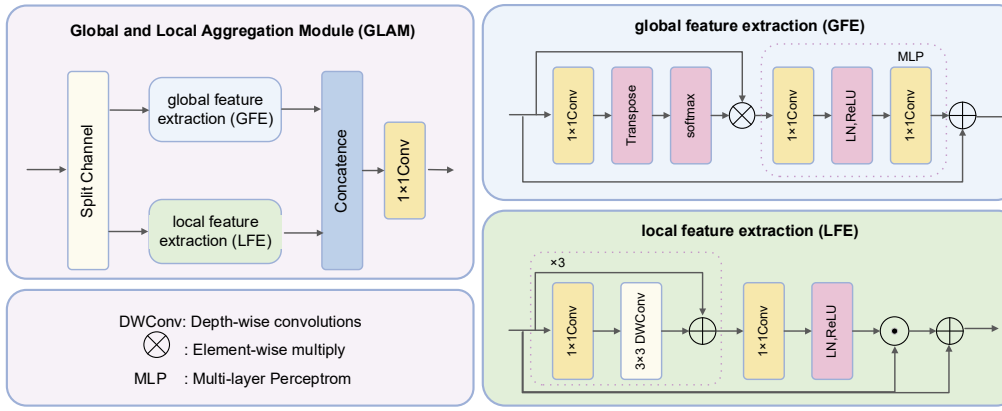


Fig. 5. Global and local aggregation module (GLAM)

Since the Vision Transformer architecture focuses more on the global semantic information of the feature map, while convolution emphasizes the detailed texture information of the feature map. The GFE adopts an attention-based computation approach, and the LFE utilizes depth wise convolution to extract local semantic information. The specific calculation method for GFE is as follows:

$$X_1^{Att} = X_1 Soft\,max(Transpose(Conv(X_1))),$$
$$GLE(X_1) = X_1 + MLP\left(R\,E\,LU\left(LN\left(Conv(X_1^{Att})\right)\right)\right), \tag{11}$$

where X_1^{Att} is the feature vector after the attention operation. Transpose represents the transposition operation. MLP corresponds to a 1×1 convolution. LN refers to the linear layer operation. $RELU$ is the activation function. The LFE effectively extracts texture details from the feature map through three layers of depthwise convolution. The specific calculation method is as follows:

$$LFE(X_2) = X_2 + X_2\delta\left(Conv\left(F_{DWC}^3(X_2)\right)\right), \tag{12}$$

where δ is the activation function. F_{DWC} represents the depthwise convolution module operation. The specific calculation method is as follows:

$$F_{DWC}(X_2) = X_2 + Conv\left(DWC(X_2)\right), \tag{13}$$

where DWC represents a 3×3 depthwise convolution operation.

3. Experimentation and analysis

3.1. Data processing

The dataset used in this experiment focuses on urban traffic and includes 8,000 images captured at various intersection scenes. Samples from the COCO and KITTI datasets were incorporated to enhance the robustness and generalization of the model. This ensured the diversity and representativeness of the dataset. Object annotations were performed using the LabelImg software, and targets were classified into six categories: pedestrian, bicycle, bus, car, motorcycle, and truck. The dataset was divided into training and validation sets in a 7:3 ratio. A solid foundation was provided for training and evaluating the model in diverse real-world scenarios.

The COCO-pose dataset was designed for human pose estimation tasks, containing over 250,000 labeled person instances. Images were collected from diverse environments and activities to ensure variability and robustness. Key points were annotated for 17 body parts, including the head, shoulders, elbows, and knees. The dataset was split into training, validation, and testing sets to facilitate comprehensive model evaluation.

3.2. Experimental parameter details

The model training was conducted on the AutoDL server platform. The hardware environment comprised an AMD EPYC 9754 128-core processor as the CPU and an NVIDIA GeForce RTX 4090D GPU with 24GB of memory. The software environment was configured with the Ubuntu 20.04 operating system, CUDA 11.8 toolkit, and PyTorch 2.0 deep learning framework. The detailed training parameters are summarized in Table 1. The input resolution was set to 640×640, with a batch size of 32. The SGD optimizer was employed and configured with an initial learning rate and final learning rate of 0.01. To ensure adequate model fitting, the number of training epochs was set to 200.

Table 1. Learning parameters

Parameter					
Input size	Batch size	Epochs	Optimizer	Lr0	Lrf
640×640	32	200	SGD	0.01	0.01

3.3. Object detection comparison experiment

To evaluate the performance of the algorithm, a series of evaluation metrics are introduced. First, mean average precision (mAP) is used as the main metric to measure the detection capability. Higher mAP indicates better detection performance across multiple object classes. Second, floating-point operations (FLOPS) are used to quantify the computational requirements during the inference process, representing the total number of floating-point operations performed. lower FLOPS values indicate lower hardware requirements. In addition, Params are used to describe the total number of weights and biases in the model, with fewer parameters indicating lower computational cost and facilitating deployment on lightweight devices. Frames Per Second (FPS) is introduced to assess the model's real-time processing capability. FPS denotes the number of image frames the model can process per second, directly reflecting its inference speed. Finally, the Average Precision (AP) per class is added to evaluate the detection accuracy of the model for each object class.

To evaluate the performance of the DAFCN network in the object detection task, this experiment combines DAFCN with the YOLOv8s detection head and compares it with several state-of-the-art object detection models, including YOLOv6 [24], YOLOv7 [25], rt-detr [26], and EfficientDet [27]. The best results for each metric are highlighted in bold, as shown in Table 2. The results indicate that YOLOv6 achieves a good balance between inference speed and detection accuracy due to its reduced parameter count and computational requirements, but it struggles with

detecting densely packed pedestrians and small vehicles. While YOLOv7 offers faster inference, its detection accuracy still has room for improvement. Rt-detr achieves high detection accuracy through its anchor-free design; however, its large parameter counts and computational cost result in an inference speed of only 64.44 FPS, falling short of real-time detection requirements. EfficientDet provides a relatively balanced performance with a smaller parameter count, yet its detection accuracy lags significantly behind other models. In contrast, the proposed DAFCN model demonstrates outstanding performance, particularly in detecting densely packed pedestrians and small vehicles. It achieves the highest detection accuracy and fastest inference speed while maintaining significantly lower parameter and computational costs compared to other models. Furthermore, DAFCN excels in terms of average precision across all categories, showcasing superior detection capabilities.

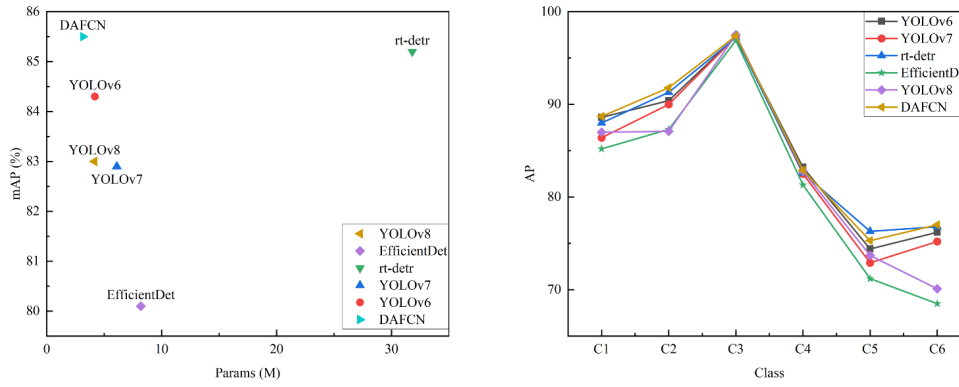
Table 2. Comparative experimental results

Comparison of system models											
Model	mAP (%)	Params (M)	FLOPs (G)	FPS (f/s)	AP of Classes						
					C1	C2	C3	C4	C5	C6	
YOLOv6	84.3	4.2	11.8	108.28	88.6	90.4	97.3	83.2	74.4	76.2	
YOLOv7	82.9	6.1	13	97.43	86.4	90	97.3	82.5	72.9	75.2	
rt-detr	85.2	31.8	125.6	64.44	88	91.3	97.2	82.6	76.3	76.8	
EfficientDet	80.1	8.2	17.5	100.26	85.2	87.3	96.9	81.3	71.2	68.5	
DAFCN (ours)	85.5	4.16	9.9	136.2	88.7	91.8	97.4	82.9	75.3	77	
DAFCN with different feature extraction networks											
Backbone	mAP (%)	Params (M)	FLOPs (G)	FPS (f/s)	AP of Classes						
					C1	C2	C3	C4	C5	C6	
efficientViT [12]	83.7	4.15	10.1	41,8	90.9	86.1	97.3	82.1	73.2	72.3	
Fasternet [23]	84.2	4.32	11.3	167.4	87.1	93.1	97.4	82.5	76.4	68.5	
convnextv2 [28]	81.6	5.81	14.7	102.4	86.8	91.1	97	80.1	68.4	66.3	
Vanillanet [9]	84.8	24.1	97.3	219.4	89.3	91.9	97.4	82.3	75.8	72.3	
MBLNet(ours)	84.1	4.16	9.9	138.5	85.3	90.7	97.5	83.4	75.1	72.4	
DAFCN with different feature fusion networks											
Necks	mAP (%)	Params (M)	FLOPs (G)	FPS (f/s)	AP of Classes						
					C1	C2	C3	C4	C5	C6	
EMBSFPN [11]	84.7	2.24	7.7	99.7	89.1	88.3	97.4	81.4	75.8	75	
ContextGuideFPN [17]	83.8	3.31	9	122.3	88.9	90.2	97.4	82.8	73.9	69.6	
CGRFPN[29]	83.9	3.58	8.9	120.1	88.4	92.6	97.2	81.9	72.2	71	
TransNeXt [18]	84.4	2.84	8.3	106	88.1	90.2	97.1	83.4	73.5	69.3	
MS-FAN (ours)	84.7	3.15	8.7	222.7	90.5	90.4	97.5	85.7	74	76.2	

In Table 2, the performance of mainstream feature extraction networks is compared with the proposed MBLNet. MBLNet achieves a balance between efficiency and accuracy, delivering 138.5 FPS inference speed and 84.1 % mAP with only 4.16M parameters and 9.9G FLOPs, outperforming EfficientViT, Fasternet, and ConvNeXtV2 in both aspects. In Table 2, the proposed MS-FAN is evaluated against FPN-based feature fusion networks. MS-FAN achieves 84.7 % mAP with 3.15M parameters, 8.7G FLOPs, and an inference speed of 222.7 FPS, significantly surpassing EMBSFPN, ContextGuideFPN, CGRFPN, and TransNeXt in both detection performance and computational efficiency. It can be observed that MS-FAN achieves higher inference speed and detection accuracy. This is accomplished while maintaining low computational complexity. Its performance in feature fusion is particularly outstanding, especially in complex scenarios.

As shown in Fig. 6(a), the horizontal axis represents the number of model parameters, while the vertical axis indicates the mAP. Models closer to the top-left corner are considered to have better performance. Benefiting from its lightweight design, DAFCN achieves the highest detection accuracy with minimal parameters and computational cost. A 4.7 % improvement in mAP is achieved compared to the baseline model YOLOv8. As shown in Fig. 6(b), DAFCN has

demonstrated excellent performance across all categories.



a) b)

Fig. 6. a) Comparison of different model mAPs with parameters, b) comparison of APs per category

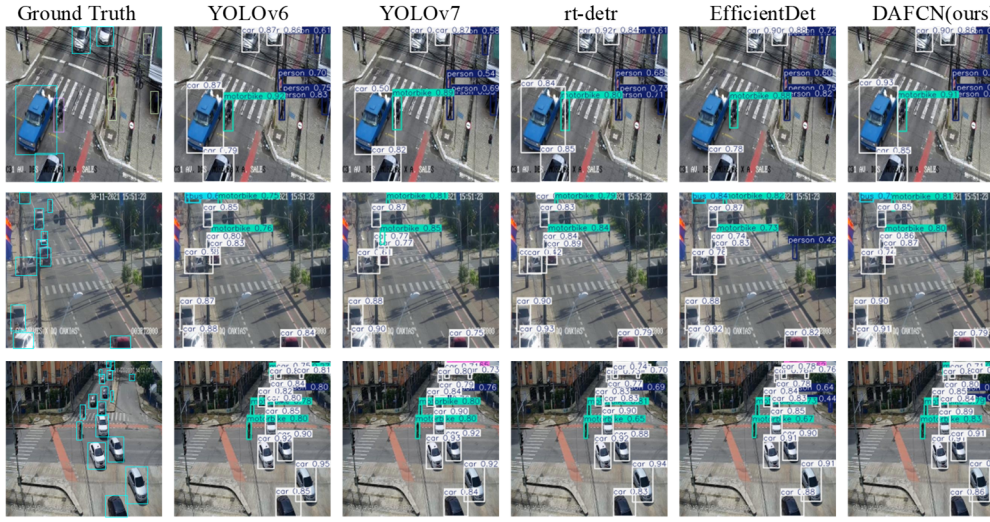


Fig. 7. Comparison of detection effectiveness of DAFCN with different obstacle detection algorithms.

The test results of YOLOv6, YOLOv7, rt-detr, EfficientDet, and DAFCN on an urban traffic dataset are compared in Fig. 7. YOLOv6 mainly relies on the basic up-sampling and down-sampling operations to realize the fusion of features at different scales, and when dealing with complex scenes, it is unable to adequately integrate semantic information at different levels, resulting in the occurrence of wrong detections. YOLOv7 adopts a relatively complex and inflexible multi-scale feature fusion strategy, which leads to the difficulty of effectively integrating different levels of feature information, and in turn affects the model's detection accuracy. Rt-detr feature fusion process is less adaptive to small sample data, leading to its missed detection. efficientDet's feature fusion is not flexible enough when dealing with complex scenes, leading to different degrees of missed detection of small targets. In contrast, DAFCN effectively detects small objects by fully integrating deep and shallow features. This proves the effectiveness and superiority of DAFCN.

3.4. Human pose estimation comparison experiment

To evaluate the performance of the DAFCN network in pedestrian pose estimation, this

ISSN PRINT 2335-2124, ISSN ONLINE 2424-4635

experiment combines DAFCN with the YOLOpose detection head and compares it with several pose estimation models, including EfficientHRNetH0 [30], OpenPose [31], and YoloPose [32]. The results are shown in Table 3.

Table 3. Comparative experimental results

Comparison of system models			
Model	mAP (%)	Params (M)	FLOPs (G)
EfficientHRNetH0	84.9	23.3	56.36
OpenPose	82.1	52.31	124.7
YoloPose	84.3	15.08	39.8
DAFCN (ours)	86.2	13.6	33.4
DAFCN with different feature extraction networks			
Backbone	mAP (%)	Params (M)	FLOPs (G)
efficientViT	84.7	4.96	11.16
Fasternet	86.2	5.05	12.49
convnextv2	82.5	6.8	16.24
Vanillanet	84.2	28.2	107.5
MBLNet (ours)	85.8	4.87	10.94
DAFCN with different feature fusion networks			
Necks	mAP (%)	Params (M)	FLOPs (G)
EMBSFPN	83.2	2.62	8.96
ContextGuideFPN	84.8	3.87	9.54
CGRFPN	84.9	4.19	9.67
TransNeXt	82.3	3.32	9.12
MS-FAN (ours)	85.2	3.65	8.82

Among them, EfficientHRNetH0 is ineffective in small target detection due to its high model complexity and insufficient feature fusion. OpenPose uses VGG as the backbone network, but its network structure is deeply hierarchical and prone to gradient vanishing, which leads to a large amount of computation and poor detection accuracy. YoloPose is based on the YOLOv3 algorithm, which is easily affected by background noise and occlusion. The highest mAP of 86.2 % is achieved by DAFCN. Meanwhile, its parameter count (13.6M) and computational cost (33.4G FLOPs) are the lowest. The ability to achieve high detection accuracy with minimal complexity is demonstrated by the proposed model. In Table 3, various feature extraction backbones are compared. Excellent performance is exhibited by the proposed MBLNet. Among them, EfficientViT has a complex model structure with multi-branch and multi-scale design, which makes the training and optimization process cumbersome and requires high computational resources. Fasternet focuses on model lightweighting and speed enhancement, and has limited feature extraction capability. ConvNeXtV2 is not robust enough in feature extraction, and cannot stably extract high-quality features in the face of complex and changing scenes. vanilanet has a simple model structure, which is insufficient in feature extraction when dealing with complex tasks, making it difficult to effectively capture important features in the data and affecting model performance. features in the data, which affect the model's performance. An mAP of 85.8 % is achieved by MBLNet with only 4.87M parameters and 10.94G FLOPs. It outperforms other backbone networks, including EfficientViT, Fasternet, and ConvNeXtV2, in both efficiency and accuracy. Finally, a comparison of feature fusion networks is shown in Table 3. Among them, EMBSFPN mainly relies on basic feature splicing and convolution operations, which easily leads to the loss of feature information and inadequate fusion, and thus affects the model's accurate target localization and classification effect. ContextGuideFPN has a limited range of context information capture when facing multi-scale targets and complex backgrounds, making it difficult to adequately cover target features at different scales. CGRFPN cannot finely adapt to the specific feature requirements of each pixel point, resulting in poorly targeted feature fusion. TransNeXt is prone to computational bottlenecks when processing high-resolution feature maps, resulting in

lower feature fusion efficiency and affecting the overall detection speed. The proposed MS-FAN achieves 85.2 % mAP with only 3.65M parameters and 8.82G FLOPs. It outperforms EMBSFPN, ContextGuideFPN, CGRFPN, and TransNeXt. Superior detection accuracy and computational efficiency are exhibited by MS-FAN.

The detection results of EfficientHRNetH0, OpenPose, YoloPose, and DAFCN are compared in Fig. 8. Missed and incorrect detections are observed in EfficientHRNetH0 due to insufficient feature representation. OpenPose fails to capture accurate pose structures in cluttered scenes, leading to errors. YoloPose, while faster, struggles with complex scenarios involving overlapping human poses and occlusions, which results in misdetections and incomplete pose estimations. In contrast, DAFCN achieves superior detection performance by leveraging its robust feature extraction and fusion capabilities. This highlights the effectiveness and reliability of DAFCN in challenging scenarios.

Fig. 8. Comparison of detection effectiveness of DAFCN with different obstacle detection algorithms

4. Conclusions

To address the distinct feature requirements of small object detection in urban traffic and pedestrian pose estimation tasks, this paper proposes the Dual-Aggregation Efficient Feature Compilation Network (DAFCN), a unified framework compatible with both tasks. DAFCN adopts a multi-channel complex convolution structure during training and a lightweight single-channel simple convolution for inference. Furthermore, it integrates a global-local dual-aggregation module to effectively fuse multi-scale global and local semantic features. In the object detection experiments on the hybrid urban traffic dataset, DAFCN achieved an mAP of 85.5 with only 4.16M parameters and 9.9 GFLOPs. In the human pose estimation experiments on the COCO-pose dataset, DAFCN reached an mAP of 86.2 with 13.6M parameters and 33.4 GFLOPs. Compared to current state-of-the-art object detection and pose estimation models, DAFCN achieves the highest accuracy while maintaining an ultra-lightweight design, meeting the performance demands of real-world applications under limited computational resources.

In future work, we plan to explore multi-sensor fusion strategies – such as combining visual and depth or thermal data – to further enhance the robustness and accuracy of both object detection and pose estimation under complex urban scenarios.

Acknowledgements

This research was funded by Guangxi Key R&D Programme Projects, Grant No. Guike AB24010143. Project on Enhancement of Basic Research Ability of Young and Middle-aged

Teachers in Guangxi Universities and Colleges, Grant No. 2023KY0911.

Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Author contributions

Huang Xiao: conceptualization, methodology, resources. Hanqing Jian: validation, writing-original draft, writing-review and editing.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] Q. Zhang, F. Yan, W. Song, R. Wang, and G. Li, "Automatic obstacle detection method for the train based on deep learning," *Sustainability*, Vol. 15, No. 2, p. 1184, Jan. 2023, https://doi.org/10.3390/su15021184

[2] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1653–1660, Jun. 2014, https://doi.org/10.1109/cvpr.2014.214

[3] H.-C. Nguyen, T.-H. Nguyen, J. Nowak, A. Byrski, A. Siwocha, and V.-H. Le, "Combined YOLOv5 and HRNet for high accuracy 2D keypoint and human pose estimation," *Journal of Artificial Intelligence and Soft Computing Research*, Vol. 12, No. 4, pp. 281–298, Oct. 2022, https://doi.org/10.2478/jaiscr-2022-0019

[4] J. Ding, S. Niu, Z. Nie, and W. Zhu, "Research on Human posture estimation algorithm based on YOLO-Pose," *Sensors*, Vol. 24, No. 10, p. 3036, May 2024, https://doi.org/10.3390/s24103036

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, Vol. 60, No. 6, pp. 84–90, May 2017, https://doi.org/10.1145/3065386

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, Jan. 2014, https://doi.org/10.48550/arxiv.1409.1556

[7] C. Szegedy et al., "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, Jun. 2015, https://doi.org/10.1109/cvpr.2015.7298594

[8] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv:1704.04861*, Jan. 2017, https://doi.org/10.48550/arxiv.1704.04861

[9] H. Chen, Y. Wang, J. Guo, and D. Tao, "Vanillanet: the power of minimalism in deep learning," *Advances in Neural Information Processing Systems*, Vol. 36, pp. 7050–7064, 2023.

[10] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10002, Oct. 2021, https://doi.org/10.1109/iccv48922.2021.00986

[11] D. Shi, "Transnext: Robust foveal visual perception for vision transformers," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17773–17783, Jun. 2024, https://doi.org/10.1109/cvpr52733.2024.01683

[12] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, and Y. Yuan, "Efficientvit: memory efficient vision transformer with cascaded group attention," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14420–14430, Jun. 2023, https://doi.org/10.1109/cvpr52729.2023.01386

[13] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2125, Jul. 2017, https://doi.org/10.1109/cvpr.2017.106

[14] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8759–8768, Jun. 2018, https://doi.org/10.1109/cvpr.2018.00913

[15] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Nas-fpn: Learning scalable feature pyramid architecture for object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7036–7045, Jun. 2019, https://doi.org/10.1109/cvpr.2019.00720

[16] J. Chen, H. Mai, L. Luo, X. Chen, and K. Wu, "Effective feature fusion network in BIFPN for small object detection," in *IEEE International Conference on Image Processing (ICIP)*, pp. 699–703, Sep. 2021, https://doi.org/10.1109/icip42928.2021.9506347

[17] S. Feng et al., "CPFNet: context pyramid fusion network for medical image segmentation," *IEEE Transactions on Medical Imaging*, Vol. 39, No. 10, pp. 3008–3018, Oct. 2020, https://doi.org/10.1109/tmi.2020.2983721

[18] M. M. Rahman, M. Munir, and R. Marculescu, "Emcad: Efficient multi-scale convolutional attention decoding for medical image segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11769–11779, Jun. 2024, https://doi.org/10.1109/cvpr52733.2024.01118

[19] X. Ding, X. Zhang, J. Han, and G. Ding, "Diverse branch block: building a convolution as an inception-like unit," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10881–10890, Jun. 2021, https://doi.org/10.1109/cvpr46437.2021.01074

[20] M. Li, X. Pan, and C. Liu, "ASOD: an atrous object detection model using multiple attention mechanisms for obstacle detection in intelligent connected vehicles," *IEEE Internet of Things Journal*, Vol. 11, No. 20, pp. 33193–33203, Oct. 2024.

[21] T. Han et al., "Epurate-net: efficient progressive uncertainty refinement analysis for traffic environment urban road detection," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 25, No. 7, pp. 6617–6632, Jul. 2024, https://doi.org/10.1109/tits.2023.3345901

[22] Y. Li et al., "Tokenpose: Learning keypoint tokens for human pose estimation," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11293–11302, Oct. 2021, https://doi.org/10.1109/iccv48922.2021.01112

[23] J. Chen et al., "Run, don't walk: chasing higher FLOPS for faster neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12021–12031, Jun. 2023, https://doi.org/10.1109/cvpr52729.2023.01157

[24] C. Li et al., "YOLOv6: a single-stage object detection framework for industrial applications," *arXiv:2209.02976*, Jan. 2022, https://doi.org/10.48550/arxiv.2209.02976

[25] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7464–7475, Jun. 2023, https://doi.org/10.1109/cvpr52729.2023.00721

[26] Y. Zhao et al., "Detrs beat yolos on real-time object detection," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16965–16974, Jun. 2024, https://doi.org/10.1109/cvpr52733.2024.01605

[27] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10781–10790, Jun. 2020, https://doi.org/10.1109/cvpr42600.2020.01079

[28] S. Woo et al., "Convnext v2: Co-designing and scaling convnets with masked autoencoders," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16133–16142, Jun. 2023, https://doi.org/10.1109/cvpr52729.2023.01548

[29] Z. Ni, X. Chen, Y. Zhai, Y. Tang, and Y. Wang, "Context-guided spatial feature reconstruction for efficient semantic segmentation," in *Lecture Notes in Computer Science*, Cham: Springer Nature Switzerland, 2024, pp. 239–255, https://doi.org/10.1007/978-3-031-72943-0_14

[30] C. Neff, A. Sheth, S. Furgurson, and H. Tabkhi, "EfficientHRNet: efficient scaling for lightweight high-resolution multi-person pose estimation," *arXiv:2007.08090*, Jan. 2020, https://doi.org/10.48550/arxiv.2007.08090

[31] S. Qiao, Y. Wang, and J. Li, "Real-time human gesture grading based on OpenPose," in *10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–6, Oct. 2017, https://doi.org/10.1109/cisp-bmei.2017.8301910

[32] D. Maji, S. Nagori, M. Mathew, and D. Poddar, "Yolo-pose: enhancing yolo for multi person pose estimation using object keypoint similarity loss," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2636–2645, Jun. 2022, https://doi.org/10.1109/cvprw56347.2022.00297

Xiao Huang received M.S. degree in transportation engineering from Wuhan University of Technology, Wuhan, China in 2006. He obtained the qualification of Senior Engineer in Traffic Engineering in 2009. Currently, he serves as the Vice Dean of the School of Traffic Management Engineering at Guangxi Police College. His research interests included road traffic management and control, traffic engineering, vehicle engineering, and communication command.

Jian Hanqing received M.S. degree in mechanical engineering from Guangxi University, Nanning, China in 2020. He currently works as a teacher at Guangxi Police College. His research interests included computer vision and intelligent transportation.