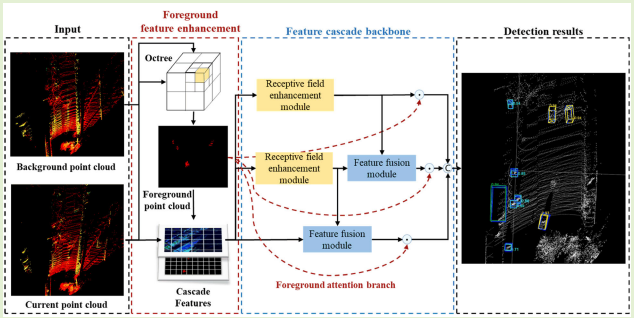
FecNet: A Feature Enhancement and Cascade Network for Object Detection Using Roadside LiDAR

Ziren Gong, Zhangyu Wang, Guizhen Yu, Wentao Liu, Songyue Yang, and Bin Zhou

***Abstract*—Roadside light detection and ranging (LiDAR) is commonly used to record the traffic data of the whole intersection scene or road segment in intelligent transportation systems (ITSs). However, general deep-learning object detection methods do not adequately consider the static background captured by roadside LiDAR. Moreover, critical issues remain to be overcome in object detection using roadside LiDAR: false alarms caused by complex background interference and multiscale objects with limited characteristics. To this end, a feature enhancement and cascade network (FecNet) is proposed to alleviate the problems. From the perspective of feature enhancement, FecNet improves foreground feature discrimination by extracting foreground information and fusing it with feature maps of multiple stages. Also, from the perspective of feature cascade, a feature cascade backbone is proposed to enhance the localization and contextual information of multiscale objects with limited characteristics. Comprehensive experiments are conducted using a roadside LiDAR dataset. The experimental results suggest that FecNet is superior to the benchmark detectors and achieves better computational efficiency and detection accuracy.**



***Index Terms*— Feature fusion, foreground feature enhancement (FFE), object detection, roadside light detection and ranging (LiDAR).**

I. Introduction

ROAD traffic safety is the key to realizing autonomous driving [1], [2], [3], [4], and sensing technology serves as a basis [5], [6]. However, the surrounding environment of autonomous vehicles is complex. For example, due to the limited perception range around the ego vehicle, onboard sensors have limited performance in preventing collisions at sharp bends or intersections. Different from onboard sensors, roadside sensors are installed at specific positions to obtain traffic data of the whole intersection or road segment. The fixed position enables them to avoid violent vibrations, thereby reducing data noise. Thus, using roadside sensors helps improve the safety of autonomous driving [7], [8].

As a typical sensor, light detection and ranging (LiDAR) has numerous advantages in intelligent perception [9], [10], [11], e.g., robust performance, high-dimensional information, and a wide sensing range. All these bring a detailed description of the environment and make it an indispensable sensor to be implemented in various object detection methods. Therefore, LiDAR plays an increasingly significant role in object detection in roadside units. However, roadside LiDAR-based object detection still confronts challenges in complex traffic scenarios [12], [13], [14]: 1) background point clouds with similar appearance characteristics usually interfere with object detection, as shown in Fig. 1, leading to false alarms; 2) objects with truncated or sparse point clouds are common in traffic scenarios and their limited features cause difficulties in object detection; and 3) the diversity of object scales puts further requirements on the detection accuracy of the network.

The above issues limit the performance of object detection using roadside LiDAR in complex traffic scenarios. To tackle the problems, this study exploits the nature of roadside LiDAR.

Manuscript received 11 July 2023; revised 9 August 2023; accepted 9 August 2023. Date of publication 17 August 2023; date of current version 2 October 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 52102448 and in part by the National Key Technologies Research and Development Program of China under Grant 2020YFB1600301. The associate editor coordinating the review of this article and approving it for publication was Dr. Guofa Li. *(Corresponding author: Zhangyu Wang.)*

Ziren Gong, Guizhen Yu, Wentao Liu, and Songyue Yang are with the School of Transportation Science and Engineering and the State Key Laboratory, Beihang University, Beijing 100191, China, and also with the Hefei Innovation Research Institute, Beihang University, Hefei 230012, China (e-mail: zirengong@buaa.edu.cn; yugz@buaa.edu.cn; zyliuwt@buaa.edu.cn; yangsy@buaa.edu.cn).

Zhangyu Wang and Bin Zhou are with the Research Institute for Frontier Science and the State Key Laboratory of Intelligent Transportation System, Beihang University, Beijing 100191, China (e-mail: zywang@buaa.edu.cn; binzhou@buaa.edu.cn).

Digital Object Identifier 10.1109/JSEN.2023.3304623

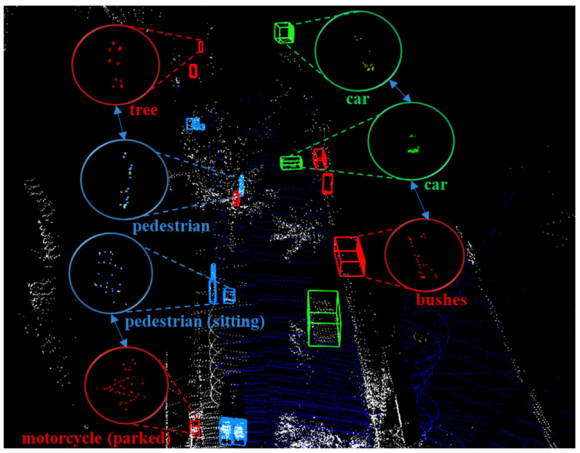


Fig. 1. Main challenges of object detection using roadside LiDAR. Blue arrows indicate that two objects share similar appearances. Red boxes refer to background obstacles, blue boxes refer to pedestrians, and green boxes refer to vehicles.

Specifically, a feature enhancement and cascade network (FecNet) are proposed for more accurate and robust object detection by exploring solutions from the perspectives of feature enhancement and feature cascade. FecNet can be divided into two parts: foreground feature enhancement (FFE) and feature cascade backbone.

FFE is designed to decrease the false alarms caused by the complex background point cloud. FFE enables the network to focus more on the foreground, which naturally preserves the precise position and basic profile information of objects. The module exploits the nature of the static background captured by the roadside LiDAR to extract the foreground features. By further fusing the foreground information with feature maps from multiple stages, FFE enables FecNet to enhance object feature discrimination in complex backgrounds with similar appearance characteristics, thus reducing false alarms.

The feature cascade backbone aims to improve the feature extraction for multiscale objects with limited features. When trying to recognize objects without environmental information, point clouds with limited characteristics are generally difficult to recognize, even for humans. To tackle this challenge, a receptive field-based enhancement module (RFEM) is devised to cascade object-correlated surroundings while extracting features with limited characteristics. To achieve accurate multiscale object detection, a feature fusion module (FFM) is proposed to enhance detailed and contextual information of multiscale objects by fusing high-level features with low-level features. In addition, the FFM refines effective features from high-level feature maps, where features that contribute to the final results can be preserved.

The main contributions of this study are summarized as follows.

1) A FecNet is proposed for object detection using roadside LiDAR in complex traffic scenarios.
2) To overcome complex background interference caused by similar appearance characteristics between objects and the background, an FFE module is proposed to extract foreground information and fuse it with feature maps from multiple stages.
3) To enhance the feature representation of objects with limited characteristics, an RFEM is devised to associate objects with their correlated surroundings.
4) To improve the accuracy of multiscale object detection, the FFM leverages the complementary features at both high- and low-level layers. Then, features without any contribution are suppressed by refining the high-level features.

The remainder of the study is structured as follows. Section II briefly reviews the related work on object detection using roadside LiDAR. Section III describes the details of the proposed FecNet framework. Section IV shows the experiment results and comparison analysis. Finally, a conclusion and future work are included in Section V.

II. Related Work

This section will briefly review closely related work in object detection using roadside LiDAR. Generally, the methods can be divided into two categories: traditional and deep-learning-based.

A. Traditional Approach

Research on object detection using roadside LiDAR is the key to realizing intelligent transportation systems (ITSs), and traditional methods are mainly adopted [15], [16], [17]. The traditional approach is a collective term for algorithms that use nondeep-learning methods (clustering, grid maps, machine learning, and so on) for point cloud structuring and object localization.

For instance, Chen et al. [18] proposed a roadside LiDAR object detection algorithm based on machine learning, which adopts a density-based clustering algorithm and uses naive Bayes, random forest, and k-nearest neighbor algorithms to perform object recognition. The result of each classification method is presented in the study. Wu et al. [19] proposed a roadside perception architecture based on 3DDSF and DBSCAN under severe conditions such as rain and snow, improving object detection accuracy in extreme weather conditions. In the literature [20], a rule-based roadside LiDAR object detection algorithm was proposed, which clusters and classifies the objects through artificial features such as geometric features, speed, and point cloud density to identify different objects on the road. Their approach analyzed the effectiveness of the selected features in classification tasks. Zhao et al. [21] first used background filtering and a DBSCAN algorithm to complete the point cloud clustering. They then used an artificial neural network to classify and track objects at an intersection. Wu et al. [22] proposed a 3D-SDBSCAN algorithm to distinguish the object points from the noise generated in rain and snow conditions. This algorithm can overcome weather occlusion problems. Considering that traditional methods have better generalization ability in mine areas, Wang et al. [23] proposed a preprocessing method to organize point clouds for vehicle detection with roadside LiDAR, including a voxel-based background filtering and a multivariate Gaussian loss (MGL).

Traditional methods can meet the efficiency requirements and achieve object detection in most scenarios when

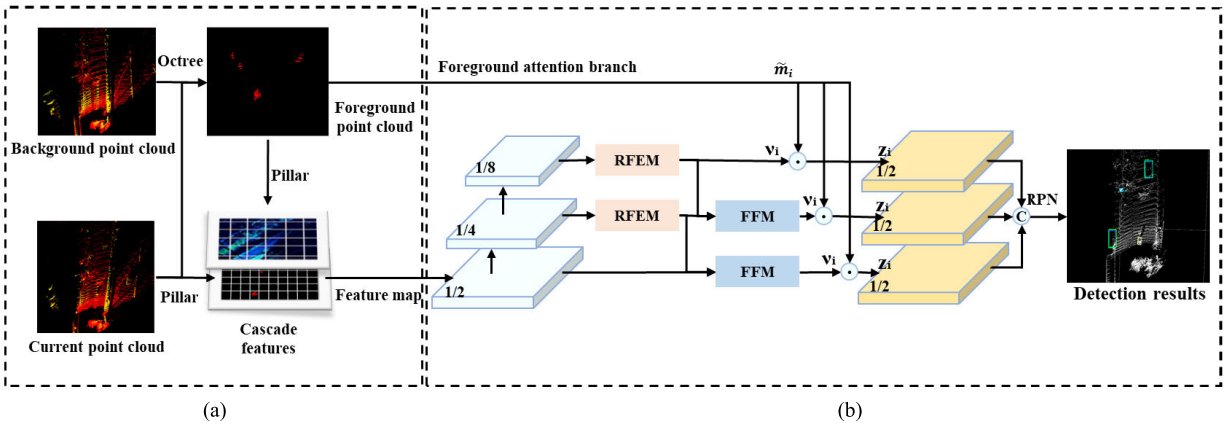


Fig. 2. Overview of FecNet. (a) FFE for learning foreground features. (b) Feature cascade backbone. RFEM represents the receptive field enhancement module, FFM means the feature fusion module, and RPN [36] is the region proposal network. $\widetilde{\boldsymbol{m}}_i$, $\boldsymbol{v}_i$, and $\boldsymbol{z}_i$ will be explained in Fig. 3. $\odot$ represents the pointwise inner product.

deployed. However, due to the sparsity of point clouds and complex background interference, the accuracy of traditional approaches is limited. Moreover, most traditional methods adopt machine-learning algorithms relying on manual feature selection, which has low generalization ability and robustness. As a result, they perform poorly on accuracy in complex scenarios.

B. Deep-Learning-Based Approach

With the success of deep learning in 3-D object detection, related approaches have been widely used in autonomous driving [24], [25], [26]. Charles et al. [27], [28] analyzed point cloud features and proposed a network to encode the spatial features of point clouds effectively. This method improved the extraction efficiency of point cloud features, and the methods in their study are used in all research based on point-based approaches. Zhou and Tuzel [29] proposed a framework to encode point clouds into fixed-size voxel grids, which standardized disordered sparse point clouds and preserved the 3-D structure of the raw point clouds. Inspired by this work, SECOND was proposed by Yan et al. [30]. They adopted sparse convolution layers to replace the 3-D convolution layers in VoxelNet [29], and this structure ran faster on GPU. Their study proved the effectiveness of sparse convolution layers using voxel-based 3-D object detection. Lang et al. [31] proposed a new encoding method called Pointpillars. By normalizing point cloud features into the form of 2-D feature maps, the features could be processed by 2-D convolution layers, which significantly improved the model's efficiency. Shi et al. [32] proposed PV-RCNN, which combined the advantages of both voxel- and point-based methods to obtain more point cloud features. They used two-stage detection to extract features. First, they obtained high-quality proposals by using voxel-based methods. Second, the local spatial information was encoded by point-based methods. Zhou et al. [33] improved PointPillars by adding dense connections and fine-tuned the pretrained model trained on open datasets to achieve vehicle detection from the captured roadside LiDAR dataset. Shi et al. [26] proposed CetrRoad, a center-aware method with a transformer-based detector for object detection using roadside LiDAR. The accuracy of deep-learning-based detectors has been greatly enhanced on open datasets [34], [35]. In recent years, using deep-learning-based methods are becoming a trend in object detection using roadside LiDAR.

However, the abovementioned deep-learning methods have their drawbacks: 1) onboard or transfer-learning-based 3-D object detectors ignore the characteristics of roadside LiDAR and 2) large detectors with computationally expensive structures struggle to meet efficiency requirements in real-world scenarios. Besides, how to decrease false alarms caused by complex backgrounds and multiscale objects with limited characteristics is still an open issue. Hence, FecNet is proposed to tackle these problems from the perspective of feature enhancement and feature cascade by exploiting the nature of the roadside LiDAR.

III. Methods

As shown in Fig. 2, FecNet is devised based on the voxel-based method [31] and a region proposal network (RPN) [36]. The framework comprises includes FFE and a feature cascade backbone.

In FFE, the study formulates the false alarms problem as an issue of the lack of discriminative features in foreground information. Accordingly, by exploiting the nature of roadside LiDAR, the foreground feature cascade and foreground attention branch are introduced to improve the discrimination of foreground features.

The feature cascade backbone comprises a receptive field enhancement module (RFEM) and an FFM. The former is used in an object-correlated context to improve the feature representation with limited characteristics. The latter refines the effective high-level features and improves the semantic features from multiple levels to detect multiscale objects.

A. Foreground Feature Enhancement

Due to the complexity of background information, the background point cloud shares similar appearance characteristics with objects. Learning an appropriate feature distribution between the background and objects with limited

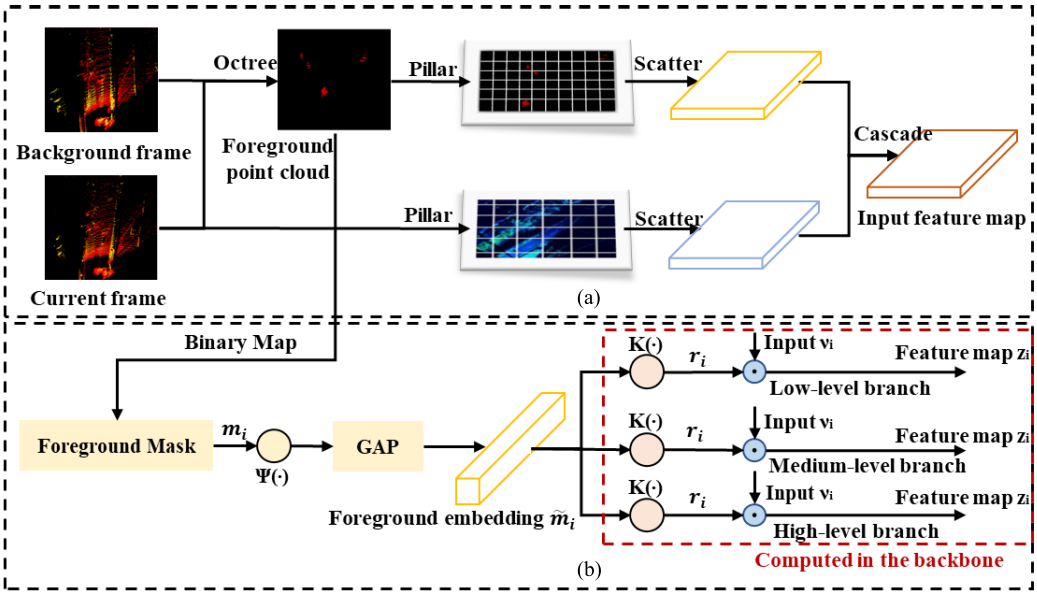


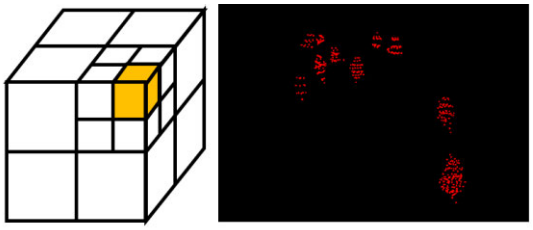
Fig. 3. FFE. (a) Foreground feature cascade for fusing foreground regions into the feature map. (b) Foreground attention branch for learning and encoding foreground features. GAP refers to global average pooling. $\widetilde{\boldsymbol{m}}_i$, $\boldsymbol{v}_i$, and $\boldsymbol{z}_i$ share the same value as in Fig. 2.

TABLE I
ALGORITHM OF FFE

Algorithm: Foreground Feature Enhancement (FFE)

Input: the background point cloud $\boldsymbol{X}_b$, the current point cloud $\boldsymbol{X}_c$.
Output: the cascaded feature map $\boldsymbol{y}$, the foreground embedding $\widetilde{\boldsymbol{m}}_i$.
1: $\boldsymbol{X}_f = Octree(\boldsymbol{X}_b, \boldsymbol{X}_c)$; // $\boldsymbol{X}_f$ represents the foreground point cloud, $Octree()$ means the Octree algorithm.
2: do in parallel
$\boldsymbol{S}_f = pillar\_to\_scatter(\boldsymbol{X}_f)$; // $\boldsymbol{S}_f$ represents the foreground feature map, $pillar\_to\_scatter()$ is to transform points to pillars and feature maps.
$\boldsymbol{S}_c = pillar\_to\_scatter(\boldsymbol{X}_c)$; // $\boldsymbol{S}_c$ represents the feature map of current frame.
$\boldsymbol{m}_i = points\_to\_binary(\boldsymbol{X}_f)$; // $\boldsymbol{m}_i$ represents the binary map of the foreground, $points\_to\_binary()$ is to transform points to the binary map.
3: **do in parallel**
$\boldsymbol{y} = concatenate(\boldsymbol{S}_f, \boldsymbol{S}_c)$; // $concatenate()$ is to cascade foreground and current feature maps.
$\widetilde{\boldsymbol{m}}_i = mask\_to\_embedding(\boldsymbol{m}_i)$; // $\widetilde{\boldsymbol{m}}_i$ represents the foreground embedding. $mask\_to\_embedding()$ is to encode the foreground region to the embedding for the fusion in the deep-layer features.
4: return $\boldsymbol{y}, \widetilde{\boldsymbol{m}}_i$

features is challenging. To distinguish the foreground features, most methods add spatial attention modules into the network [36], [37] to enhance foreground features or model foreground–scene relationships [38]. However, these methods are all proposed for 2-D object detection and ignore the encoding of foreground information. To address this problem, FFE is proposed to enhance the discrimination of foreground features of the multiple stages, thus reducing false alarms. As shown in Fig. 3, two perspectives of explicit foreground feature learning are explored in FFE: the foreground feature cascade and the foreground attention branch. The two operations are processed in parallel after the extraction of the foreground point cloud to obtain the cascaded feature map and the foreground embeddings. The whole process of FFE is further explained in Table I.



(a) (b)

Fig. 4. (a) Octree structure and the result of foreground extraction. The red points shown in (b) belong to the foreground point cloud of the current frame.

1) Foreground Feature Cascade: FFE has two inputs: the background and the current point cloud. The background point cloud is saved while capturing the road segment without objects. In the foreground feature cascade, the module cascades the feature map encoded from the current point cloud with the foreground feature map to improve foreground feature discrimination. In the process, the foreground point cloud is encoded and fused with the current frame's feature map as the backbone's input. The cascaded feature map with the prior foreground information enables FecNet to learn foreground features while extracting features.

As shown in Fig. 3(a), this part exploits the nature of static backgrounds by utilizing Octree to distinguish the foreground points. The structure of Octree is shown in Fig. 4(a). Octree inputs both the background and foreground point cloud. By dividing each internal node into eight subsets, Octree can quickly compare the leaf nodes of the current and background frame to differentiate those point clouds not belonging to the background. In the process, Octree establishes leaf nodes in the background and current frames and then detects the foreground point cloud by comparing their leaf nodes. In most cases, the foreground point cloud can easily be extracted from the current frame [39]. Fig. 4(b) shows the result of the foreground extraction. Afterward, the foreground point

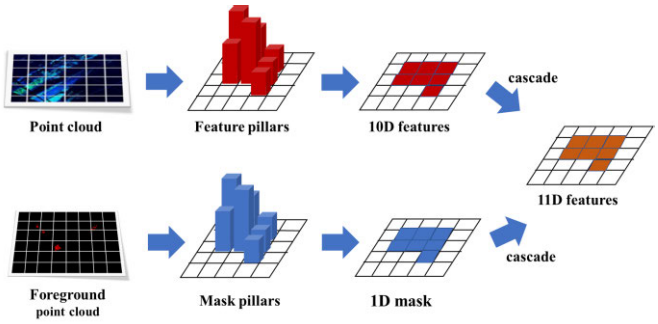


Fig. 5. Process of the foreground feature cascade. The two branches are processed in parallel, as shown in Fig. 3(a). The red pillars represent the pillars encoded by the original point cloud; the blue pillars represent the pillars encoded by the one-hot mask; the orange pillars are cascaded pillars with foreground information.

cloud is encoded by both the foreground feature cascade and foreground attention branch.

The extracted foreground point cloud is given a one-hot mask with 0/1 to mark the foreground region. Then, both the original point cloud of the current frame and the one-hot mask are encoded into pillars in which the original point cloud is represented by $|\boldsymbol{D_0}| = 10$ dimensions, as shown in red in Fig. 5

$$\boldsymbol{D_0} = \{x, y, z, i, x_c, y_c, z_c, x_p, y_p, z_p\} \tag{1}$$

where x, y, and z denote the point cloud coordinates in the LiDAR coordinate system, i is the intensity of each point, x_c, y_c, and z_c represent the coordinate of the point cloud geometric center in the pillar, and x_p, y_p, and z_p represent the Euclidean distance offset between the coordinates of the pillar's geometric center and each point in the pillar.

The one-hot mask of the foreground point cloud is represented by the $|\boldsymbol{D_m}| = 1$ dimension, as shown by the blue one in Fig. 5

$$\boldsymbol{D_m} = \{m\} \tag{2}$$

where m represents the one-hot mask, 0 denotes the background point cloud, and 1 marks the foreground point cloud.

The mask pillars (blue pillars) with foreground region information are fused with feature pillars (red pillars) extracted from the original point cloud to get the cascaded pillars (orange pillars). Through cascading the foreground mask with point cloud features, the cascaded pillars have $|\boldsymbol{D}| = 11$ dimensions, including $\boldsymbol{D_m}$ and $\boldsymbol{D_0}$, as orange pillars shown in Fig. 5

$$\boldsymbol{D} = \{x, y, z, i, x_c, y_c, z_c, x_p, y_p, z_p, m\}. \tag{3}$$

2) Foreground Attention Branch: In general, limited features of objects with similar characteristics to the background are gradually lost in deep layers due to multiple convolutions and downsampling. This situation leads to the problem of inappropriate feature distribution between objects and the background. To address the problem, this part encodes the foreground features and then fuses them with the deep-layer features to increase the disparity between foreground and background features.

As shown in Figs. 3(b) and 2(b), the foreground attention branch encodes the foreground point cloud to the embedding vector $\widetilde{\boldsymbol{m}}_i$, which will be then used to reweight the deep-layer features $\boldsymbol{v}_i$ in multilevel branches to produce new feature maps $\boldsymbol{z}_i$. To obtain the foreground embedding vector $\widetilde{\boldsymbol{m}}_i$, this part first transforms the foreground point cloud to a binary map as the foreground mask $\boldsymbol{m}_i$. The binary map $\boldsymbol{m}_i$ has zeros everywhere except in locations occupied by objects and is then analyzed by the projection function $\psi(\cdot)$: $R^{\mathbf{1}\times H\times W} \rightarrow R^{\mathbf{64}\times H/\mathbf{4}\times W/\mathbf{4}}$, in which $\boldsymbol{H} \times \boldsymbol{W}$ means the binary map size. As shown in the following equation, $\psi(\cdot)$ is followed by a global average pooling (GAP) and sigmoid gate function to sum out the spatial information for obtaining the foreground embedding $\widetilde{\boldsymbol{m}}_i$:

$$\widetilde{\boldsymbol{m}}_i = \frac{1}{1 + \exp(-\mathrm{GAP}(\psi(\boldsymbol{m}_i)))} \tag{4}$$

where $\widetilde{\boldsymbol{m}}_i$ is a foreground embedding and $\psi(\cdot)$ is defined as a simple form that is implemented by two $\mathbf{3} \times \mathbf{3}$ convolutional layers (stride $= 2$). Using convolutions and the GAP operation allows the model to extract the object locations from the foreground mask $\boldsymbol{m}_i$. Afterward, FecNet fused the encoded foreground embedding $\widetilde{\boldsymbol{m}}_i$ with the deep-layer features so as to leverage the knowledge from the embeddings to learn a proper feature distribution.

To reweight the deep-layer features $\boldsymbol{v}_i$ to the new feature map $\boldsymbol{z}_i$ in multilevel branches, the foreground embedding $\widetilde{\boldsymbol{m}}_i$ is first transformed to the same channels as $\boldsymbol{v}_i$ in multilevel branches by the scale-ware projection function $K(\cdot)$: $R^{d\times H\times W} \rightarrow R^{d_u\times H\times W}$, as shown in the following equation:

$$\boldsymbol{r}_i = K(\widetilde{\boldsymbol{m}}_i) \tag{5}$$

where $\boldsymbol{r}_i$ denotes foreground embedding and $K(\cdot)$ denotes a projection function for channel transformation. A learnable $\mathbf{1} \times \mathbf{1}$ convolutional layer implements $K(\cdot)$ with output channels of $\boldsymbol{d_u}$. $\boldsymbol{d_u}$ represents the channel number of feature maps in different branches. Hence, the output feature map $\boldsymbol{z}_i$ can be naturally obtained by the following equation:

$$\boldsymbol{z}_i = \boldsymbol{v}_i \odot \boldsymbol{r}_i \tag{6}$$

where $\boldsymbol{z}_i$ denotes the final feature maps with enhanced foreground features and $\boldsymbol{z}_i$ is obtained by reweighting the multilevel feature map $\boldsymbol{v}_i$ with foreground embedding $\boldsymbol{r}_i$ through the pointwise inner product $\odot$.

B. Feature Cascade Backbone

In order to improve the performance of object detection, two key issues need to be considered: 1) the model should enhance the feature discrimination for objects with limited characteristics and 2) the model should focus on the features that contribute to the final results when predicting multiscale objects.

For objects with limited characteristics, the fusion of object features and their surroundings could naturally preserve the localization information. A suitable receptive field is required to learn rich contextual relationships and random

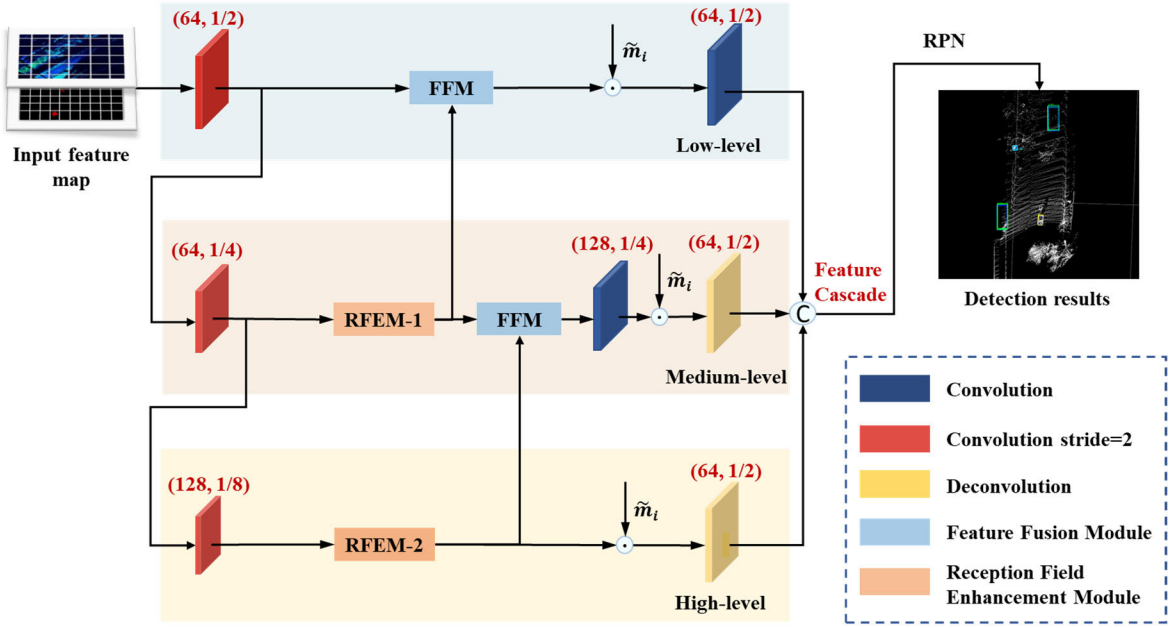


Fig. 6. Details of the feature cascade backbone. The input feature map comes from the output of the foreground feature cascade in FEE. $\tilde{m}_i$ denotes the foreground embeddings and shares the same value as in Figs. 2 and 3.

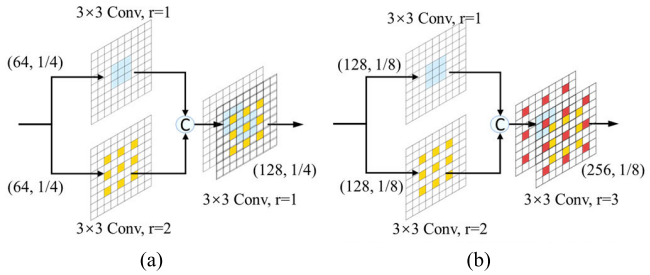


Fig. 7. Structure illustrating the RFEM. Conv. means the convolutional layer, and r denotes the expansion rate. (a) Illustration of RFEM-1 in medium level. (b) Illustration of RFEM-2 in high level.

relationships among these features [40], [41], [42] more effectively. Therefore, we proposed an RFEM.

Aiming to achieve multiscale object detection, most networks are designed with FPN [43], such as the pyramid-pooling module [44], [45] and pyramid-shaped hollow pooling [46], [47]. However, these modules are mostly used for image features that have balanced feature distributions. Different from images, point clouds have uneven feature distribution due to their sparsity. Only object-related features are effective for the prediction of multiscale objects. To alleviate these problems, we devised an FFM. The structure of the feature cascade backbone is shown in Fig. 6, and the computing process is shown in Table II. The input cascaded feature map is encoded by RFEM and FFM in the three branches (low level, medium level, and high level) in parallel after downsampling.

1) Receptive Field Enhancement Module: A suitable receptive field determines the scope of feature learning in convolutional neural networks, which is crucial to the performance of the network [48]. Through receptive field cascade, RFEM enables FecNet to associate object-correlated contexts, thereby learning the symbiotic relation between the scene and objects.

TABLE II
PROCESS OF THE FEATURE CASCADE BACKBONE

Algorithm: Feature cascade backbone

Input: the cascaded feature map $\boldsymbol{y}$, the foreground embedding $\tilde{\boldsymbol{m}}_i$.
Output: the detection outputs $\boldsymbol{O}_i$.
1: $\boldsymbol{y}_1 = downsampling(\boldsymbol{y})$; // The size of $\boldsymbol{y}_1$ is one-half of $\boldsymbol{y}$.
2: $\boldsymbol{y}_2 = downsampling(\boldsymbol{y}_1)$; // The size of $\boldsymbol{y}_2$ is one-fourth of $\boldsymbol{y}$.
3: $\boldsymbol{y}_3 = downsampling(\boldsymbol{y}_2)$; // The size of $\boldsymbol{y}_3$ is one-eighth of $\boldsymbol{y}$.
4: do in parallel
$\boldsymbol{e}_2 = RFEM(\boldsymbol{y}_2)$; // $\boldsymbol{e}_2$ represents the output of RFEM-1. $RFEM()$ is the RFEM module.
$\boldsymbol{v}_3 = RFEM(\boldsymbol{y}_3)$; // $\boldsymbol{v}_3$ represents the output of RFEM-2.
5: do in parallel
$\boldsymbol{v}_1 = FFM(\boldsymbol{y}_1, \boldsymbol{e}_2)$; // $\boldsymbol{v}_1$ represents the output of FFM. $FFM()$ is the FFM module.
$\boldsymbol{v}_2 = Conv(FFM(\boldsymbol{e}_2, \boldsymbol{v}_3))$; // $\boldsymbol{v}_2$ represents the output of FFM after a convolution. $Conv()$ is a convolution layer.
6: do in parallel
$\boldsymbol{z}_1 = transform\_channel(\tilde{\boldsymbol{m}}_i) \odot \boldsymbol{v}_1$; // $\boldsymbol{z}_1$ represents the reweighted feature map. $transform\_channel()$ is to transform $\tilde{\boldsymbol{m}}_i$ to the same channel as $\boldsymbol{v}_1$. $\odot$ means the pointwise inner product.
$\boldsymbol{z}_2 = transform\_channel(\tilde{\boldsymbol{m}}_i) \odot \boldsymbol{v}_2$; // $\boldsymbol{z}_2$ represents the reweighted feature map.
$\boldsymbol{z}_3 = transform\_channel(\tilde{\boldsymbol{m}}_i) \odot \boldsymbol{v}_3$; // $\boldsymbol{z}_3$ represents the reweighted feature map.
7: do in parallel
$\boldsymbol{z}'_1 = Conv(\boldsymbol{z}_1)$;
$\boldsymbol{z}'_2 = Deconv(\boldsymbol{z}_2)$; // $Deconv()$ is a deconvolution layer.
$\boldsymbol{z}'_3 = Deconv(\boldsymbol{z}_3)$;
8: $\boldsymbol{O}_i = RPN(concatenate(\boldsymbol{z}'_1, \boldsymbol{z}'_2, \boldsymbol{z}'_3))$; // $RPN()$ is a region proposal network. $concatenate()$ is to cascade the three reweighted feature maps.
9: return $\boldsymbol{O}_i$

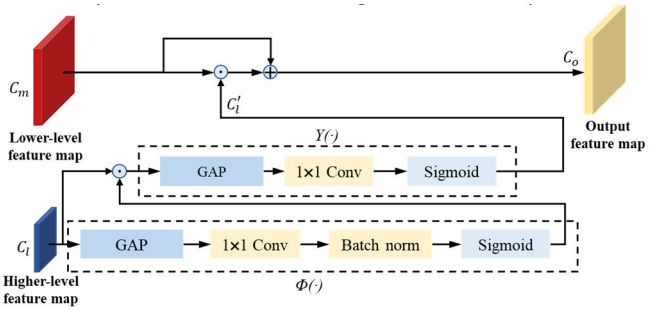


Fig. 8. Structure of the FFM in the feature cascade backbone. Sigmoid is an activation function.

RFEM is used in medium- and high-level branches after feature downsampling (the red convolutions in Fig. 6). Aiming to enhance object feature discrimination, RFEM cascades the multiscale receptive field and learns from the joint features of objects and their surroundings. As shown in Fig. 7(a) and (b), RFEM uses two $\mathbf{3} \times \mathbf{3}$ dilated convolutions with expansion rates of 1 and 2 to extract objects and their surrounding features. Afterward, a concatenation layer obtains the two features to associate objects with related contextual information. In the medium-level branch, RFEM uses a 128-channel $\mathbf{3} \times \mathbf{3}$ convolution to integrate the 128-channel cascading features. In the high-level branch, RFEM utilizes a 256-channel $\mathbf{3} \times \mathbf{3}$ dilated convolution (expansion rate $= 3$) to furtherly explore global context from the 256-channel cascading features for object detection. The above steps are represented as follows:

$$f_{\text{rfem}}(x) = f_{\text{conv}}(f_{\text{cat}}(f_{\text{conv}}(x), f_{\text{Dconv}}(x))) \tag{7}$$

where f_{rfem} denotes a projection function for RFEM, x is the input of the module, and f_{cat} is to concatenate a convolutional layer f_{conv} and a dilated convolutional layer f_{Dconv}.

RFEM has two structures in different branches. In the medium-level branch, the module can cascade features in the blue and yellow regions, which means that the object features (blue region) and their surrounding context (yellow region) are associated. In the high-level branch, the cascading features (yellow and blue regions) are further processed to capture a larger receptive field to expand the spatial context (green region). The proposed module improves the representation of objects by learning their features with spatial correlation, which enables the framework to perform better for objects with limited characteristics.

2) Feature Fusion Module: Although FPN [43] can integrate features with rich semantic information into features with more localization information to detect multiscale objects, such a naive connection is not enough to refine the effective object-correlated features and the complementariness between high- and low-level features. Thus, as shown in Figs. 6 and 8, an FFM is proposed to preserve effective features and deliver contextual information from deep layers to shallow layers. This module can extract suitable features from a higher level branch to guide the fusion between adjacent layers. Given the pyramid feature maps $\boldsymbol{C_l}$ and $\boldsymbol{C_m}$ from RFEM in higher and lower level branches, the upsampled feature map $\boldsymbol{C_l'}$ with object-related semantic features is computed via two projection functions $\Phi(\cdot)$ and $\Upsilon(\cdot)$. Hence, $\boldsymbol{C_l'}$ can be naturally obtained by the following equation:

$$\boldsymbol{C_l'} = \Upsilon(\boldsymbol{C_l} \odot \Phi(\boldsymbol{C_l})) \tag{8}$$

where $\Phi(\cdot)$ is implemented by a $\mathbf{1} \times \mathbf{1}$ convolutional layer followed by batch normalization and a sigmoid gate function, which is learnable to reweight the features contributing to multiscale object prediction through the pointwise inner product $\odot$. $\Upsilon(\cdot)$ is a channel transformation projection function similar to $\Phi(\cdot)$ without batch normalization. The refined feature map $\boldsymbol{C_l'}$ will be fused into the feature map $\boldsymbol{C_m}$ in shallow layers within the scale range of the next layer.

To aggregate upsampled feature maps from the higher level layer, the point-wised product and add operation are employed to improve computation and parameter efficiency. $\boldsymbol{C_m}$ is then reweighted and transformed by the refined feature map $\boldsymbol{C_l'}$, as shown in the following equation:

$$\boldsymbol{C_o} = \boldsymbol{C_m} \odot \boldsymbol{C_l'} + \boldsymbol{C_m} \tag{9}$$

where $\boldsymbol{C_o}$ denotes the aggregated feature maps with spatial information and contextual information. By implementing the proposed FecNet, the false alarms in object detection can be significantly decreased, and the efficiency of FecNet can be improved with the lightweight feature cascade backbone.

IV. Experiments

A. Dataset and Evaluation Criteria

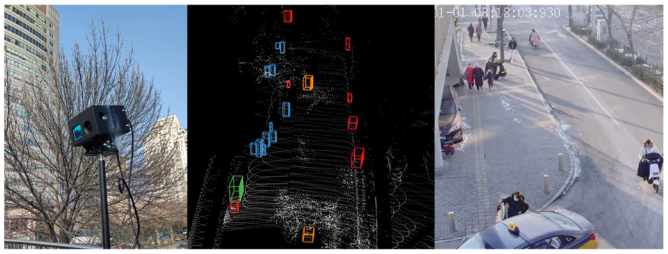
In applications of ITS, multiscale objects with limited features in complex backgrounds have always been the difficulty that restricts the accuracy of 3-D object detection using roadside LiDAR. From this perspective, this study created the roadside dataset to test the performance of the proposed network for further application in ITS, especially vulnerable road user detection. Compared with KITTI [34], Waymo [35], and DAIR-V2X-I [49], the dataset included complex backgrounds and multiscale objects with limited features.

The traffic data were recorded from the road segment in Zhichun Road, Beijing. Livox Horizon LiDAR was mounted at a fixed height of 6 m to collect the data. The capture frequency is 10 Hz; 3000 frames of LiDAR data with 19 882 objects annotated. There are totally three object classes: pedestrian, cyclist, and vehicle. The corresponding categories of objects are labeled according to KITTI [34], such as their label, seven-DOF 3-D bounding box (location: x, y, and z, and size: width, length, height), and occlusion. There were 2100 frames for training, 300 frames for evaluation, and 600 frames for testing. Table III shows the statistics of the number of vehicles, pedestrians, and cyclists for training, evaluation, and testing.

As shown in Fig. 9(b) and (c), the dataset includes a wealth of samples that vary in scale. Between categories, the scale of vehicles (e.g., vans) is significantly larger than that of pedestrians (e.g., children). Within the same categories, the scales of pedestrians vary with different states (sitting, occluded, and so on). Besides, as shown by the red boxes in Fig. 9(b), there are a large number of background obstacles

TABLE III
STATISTICS OF SAMPLES FOR TRAINING AND TESTING

Class Name	No. for Training	No. for evaluation	No. for Testing
Vehicle	3417	475	792
Pedestrian	9283	1474	1727
Cyclist	3948	581	715
Total	16648	2530	3234



(a) (b) (c)

Fig. 9. (a) Roadside LiDAR, (b) point cloud of the scenario, and (c) real-world road segment. Red boxes refer to background obstacles, blue boxes refer to pedestrians, green boxes refer to vehicles, and orange boxes refer to cyclists.

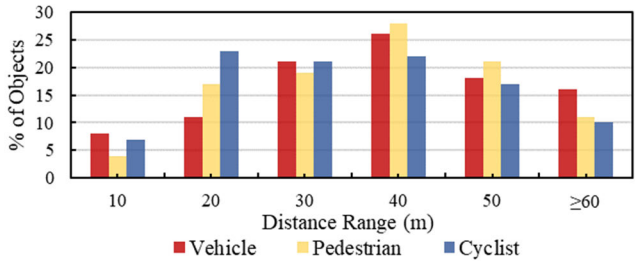
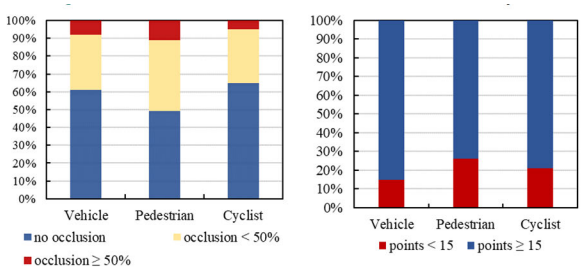


Fig. 10. Distance distribution of annotated objects.



(a) (b)

Fig. 11. (a) Occlusion and (b) point cloud number of objects.

(e.g., trees, bushes, and litter bins) similar in appearance to the object.

Considering the application in intelligent transportation scenarios, this study focuses on the objects within the road segment. The dataset collects objects at different distances to ensure a diversity of samples. As shown in Fig. 10, the depth of the captured objects can range from within 10 to over 60 m, and the max distance of objects is about 80 m. In addition, the dataset annotates three levels of occlusions: no occlusion, less than 50% occlusion, and more than 50% occlusion. Nearly, half of the objects are partially or severely occluded, as shown in Fig. 11(a). The point cloud number of objects is shown in Fig. 11(b). Both the degree of occlusion and the point cloud number suggest that objects with limited features are common in the dataset.

AP and mAP are used to evaluate object detection accuracy. Also, framework efficiency was tested using runtime ms and FPS. The definition of AP is given as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{11}$$

$$\text{AP} = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1.0\}} \max_{\tilde{r} \geq r} (\tilde{r}) \tag{12}$$

where Precision denotes the object detection accuracy and Recall represents the object detection completion rate. Correct detections are denoted as TP, and false detections are represented as FP. FN and denote missed detections and recall rate, respectively. AP is the area under the curve when Recall is used as the abscissa, Precision is used as the ordinate, and mAP is the average AP of all categories.

B. Data Training

The proposed framework and compared methods implemented the same data augmentation, including translation, rotation, and zoom. The translation can be plotted along the x- and y-axes. The probability of random rotation was set to 0.5, and the radian range of rotation was set to $[-0.78, 0.78]$. In order to increase the diversity and richness of the dataset, the zoom scale was set to [0.95, 1.05].

The experiments were trained on the NVIDIA Geforce GTX Titan X server using Pytorch. We use Adam's one-cycle optimizer over a GPU with 20 frames/batch, a weight decay of 0.01, and a momentum of 0.9. For all the experiments, the initial learning rate was set at 0.003.

C. Ablation Study

In this section, comprehensive experiments were designed to analyze the proposed modules, including FFE, RFEM, and FFM. The baseline consists of FPN [43] and a lightweight decoder, the same as the proposed FecNet. When FFE is not added, the input of the baseline is the current point cloud. By adding the modules to the baseline, the performance of the proposed modules could be evaluated. The results of the ablation experiments are shown in Table IV.

1) Foreground Feature Enhancement: Table IV [see (b), (f), (g), and (h)] shows the ablation results of FFE based on the baseline [Table IV, see (a)]. The FFE brings 0.44%, 1.07%, and 1.45% performance gains in AP of the three categories. The result suggests that FFE has a better performance compared to the baseline, especially on small-scale objects. Moreover, the performance gains cannot be influenced when FFE is used with RFEM and FFM, as shown in Table IV [see (f)–(h)]. This result indicates that FFE helps improve the accuracy of small-scale object detection.

The effects of FFE can be explained from two perspectives. First, the gains come from the foreground feature attention

TABLE IV
ABLATION EXPERIMENTS ON THE ROADSIDE DATASET. THE BOLD VALUES IN EACH COLUMN MEAN THE BEST ENTRIES

Method	Vehicle AP (%)	Pedestrian AP (%)	Cyclist AP (%)	mAP (%)
(a) Baseline	85.01	80.19	87.41	84.20
(b) +FFE	+0.44	**+1.07**	+1.45	+0.99
(c) +RFEM	**+1.58**	+0.38	+1.22	+1.06
(d) +FFM	+0.75	+0.84	+1.35	+0.98
(e) +RFEM+FFM	+1.20	+0.72	+1.57	+1.16
(f) +FFE+RFEM	+1.39	+1.01	+1.69	+1.36
(g) +FFE+ FFM	+1.06	+1.05	+1.76	+1.29
(h) +FFE+RFEM+FFM	**+2.94**	**+1.07**	**+2.34**	**+2.12**

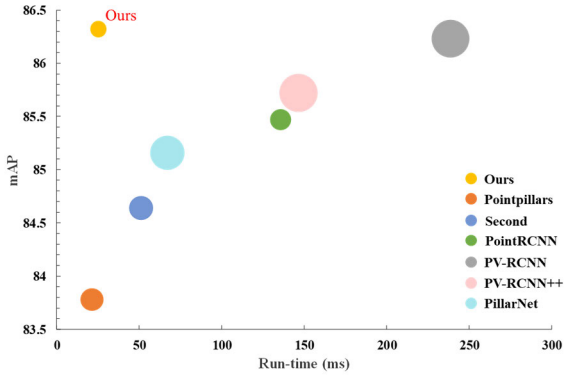


Fig. 12. Efficiency and accuracy of the methods. The size of the point radius indicates the number of parameters.

branch, which encodes the foreground point cloud into embeddings. Afterward, the possible foreground spatial information can be integrated into deep-layer features, which can help localize small-scale objects. Thus, due to the decrease in false alarms caused by complex backgrounds, the accuracy of small-scale objects obtains significant gains. Second, the gains come from fusing foreground features into the feature maps as the backbone input. The proposed foreground feature cascade enables FecNet to focus on foreground regions by learning from the prior foreground information while extracting features. By fusing features from multiple stages with the foreground features, FFE significantly improved the performance of object detection.

2) Receptive Field Enhancement Module: RFEM contains dilated convolutional layers with different receptive fields to expand and associate the relative contextual features. Table IV [see (c)] shows that the RFEM performs better with large-scale objects such as SUVs, vans, and cars, with 1.58% gains. Furthermore, RFEM has significant gains in mAP used with any module, as shown in Table IV [see (e), (f), and (h)].

The performance gain is significant in vehicle and cyclist detection, which mainly comes from two factors: 1) the feature extraction of vehicles and cyclists relies more on their surroundings, which may be related to the higher predictability of vehicles and bicycles and 2) RFEM associates object features with their surroundings to enhance feature discrimination of objects, especially localization information. RFEM combines local, contextual, and global features, which could effectively expand the receptive field and learn the joint features of scenes and objects. This helps to exploit the symbiotic relationship between objects and background and associate foreground-related contexts.

3) Feature Fusion Module: Compared to the baseline, FFM enables FecNet to preserve effective features of high-level feature maps. Table IV [see (d)] shows the ablation results of FFM. FFM boosts the performance of FPN (in the baseline) with a 0.98% gain in mAP, and it performs better in Table IV [see (e), (g), and (h)], with 1.16%, 1.29%, and 2.12% gains, respectively. The result demonstrates that the correlations of feature channels between adjacent layers modeled by the FFM significantly preserve more important semantic context information for multiscale object prediction. Thus, FFM contributes to detection accuracy.

Table IV [see (h)] shows that the combination of the three modules achieves the best performance improvement, with 2.94% in vehicles, 1.07% in pedestrians, 2.34% in cyclists, and 2.12% in mAP. The results indicate that these three proposed modules improved the performance of object detection using roadside LiDAR. The performance gain comes from obtaining complementary features at multilevel layers and enhancing the significant features such as foreground features, global features, and the surrounding context.

D. Comparison Study

To further evaluate FecNet, we also compared FecNet with representative 3-D object detectors on the roadside dataset. The detectors include voxel-based networks [30], [31] and point-based networks [50], such as the latest PillarNet [52] and PV-RCNN++ [51]. Accuracy and efficiency are evaluated by mAP, runtime (per frame), and the number of parameters. As shown in Fig. 12, FecNet achieved the best tradeoff between accuracy and efficiency with 86.32% mAP, 25.2 ms, and the smallest model size on TITAN X GPU.

As shown in Table V, FecNet outperforms other state-of-the-art 3-D object detection methods. In terms of accuracy, FecNet achieved the best result with 86.32%, in which the performance on vehicles and cyclists is significantly superior to other methods. The essential reason can be attributed to the fact that FecNet fully considers the roadside LiDAR characteristics, foreground information, object-related context, and the symbiotic relationship between objects and backgrounds. This enables FecNet to overcome the abovementioned challenges: false alarms caused by complex background interference and multiscale objects with limited characteristics.

Fig. 13 shows the experimental results in two typical scenes, in which FecNet is compared to the two typical networks: the highly efficient method (Pointpillars [31]) and the highly accurate method (PV-RCNN [32]). As shown in Fig. 13(b) and (c) in row 1, Pointpillars and PV-RCNN cannot correctly detect the pedestrian and the cyclist at a distance of about 58 m. However, despite the limited features of these objects, FecNet achieves better localization and categorizing

TABLE V
PERFORMANCE OF 3-D DETECTORS ON ROADSIDE DATASET. THE BOLD VALUES IN EACH COLUMN MEAN THE BEST ENTRIES

Methods	Vehicle AP (%)	Pedestrian AP (%)	Cyclist AP (%)	mAP (%)	TITAN X GPU	
					Runtime (ms)	Params
Pointpillars [31]	84.44	80.17	86.73	83.78	**21.4**	4.83M
SECOND [30]	85.55	80.65	87.73	84.64	51.2	5.33M
PointRCNN [50]	81.61	**88.47**	86.32	85.47	135.8	4.04M
PV-RCNN [32]	85.24	85.93	87.51	86.23	238.8	13.12M
PV-RCNN++ [51]	86.51	81.91	88.74	85.72	146.7	13.66M
PillarNet [52]	87.24	80.71	87.53	85.16	67.1	10.99M
Ours	**87.95**	81.26	**89.75**	**86.32**	**25.2**	**2.53M**

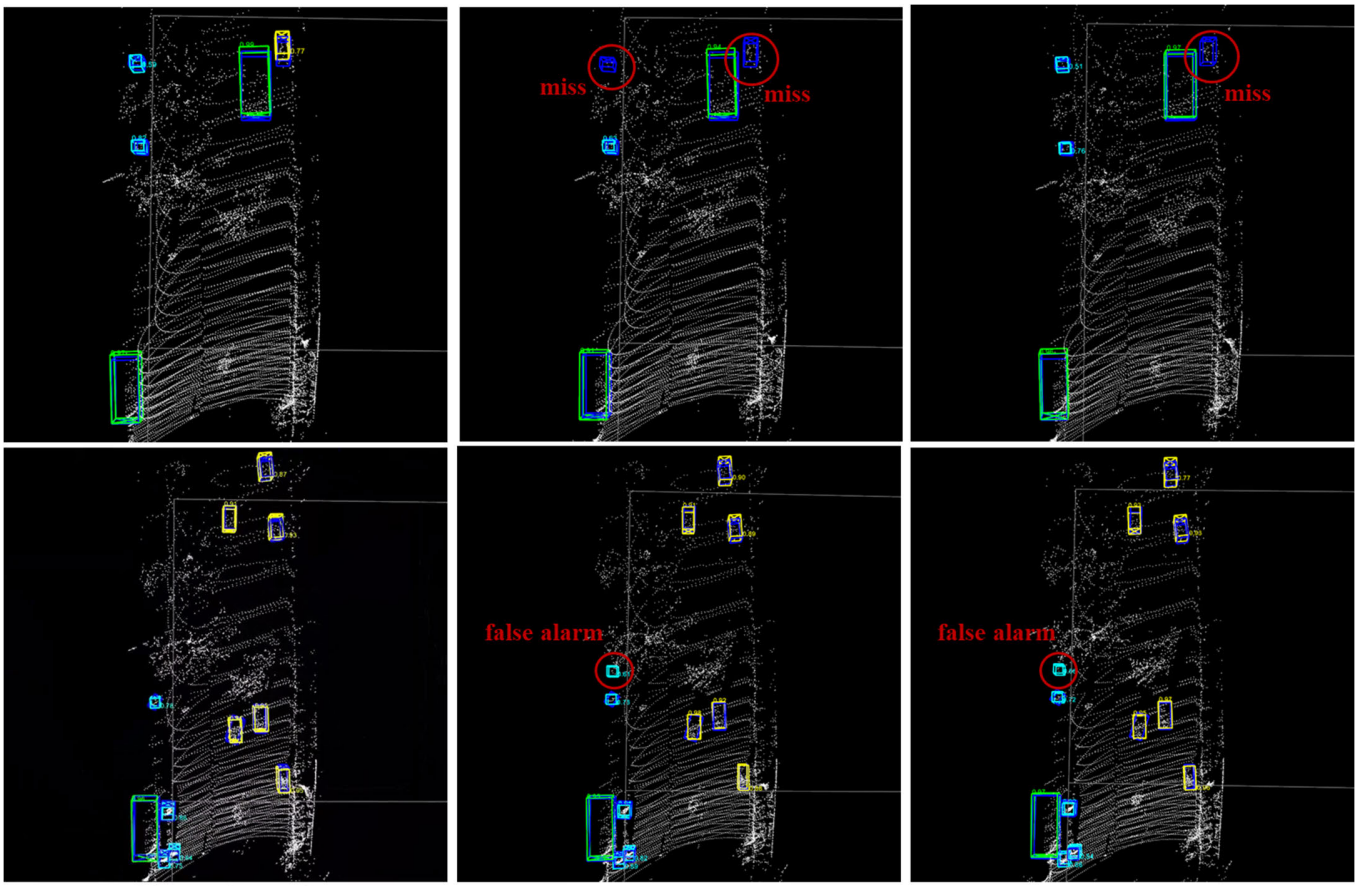


Fig. 13. Results on the roadside dataset. (a) Ours (FecNet). (b) Pointpillars. (c) PV-RCNN. Rows 1 and 2 show the results of these methods in two typical scenes (ground truth, vehicles, pedestrians, and cyclists).

for them. This is because the proposed RFEM and FFM allow FecNet to detect objects with limited characteristics by leveraging the symbiotic relationship between objects and background and associating object-related contexts. For pedestrian detection in row 2, both Pointpillars and PV-RCNN have false alarms. The similarity between the appearance characteristics of sparse pedestrian points and the background causes these two methods unable to effectively distinguish the objects from complex backgrounds. The proposed FFE enables FecNet to extract the foreground information and fuse them with features of multistage to improve object feature discrimination. Therefore, FecNet performs better in distinguishing valid objects, as shown in Fig. 13(a) in raw 2.

Moreover, FecNet achieves a better accuracy/speed tradeoff with the smallest model size on the roadside dataset. The inference speed of FecNet is about 40 FPS (runtime with 25.2 ms), comparable to the most efficient one, Pointpillars. The efficiency of FecNet benefits from a lightweight backbone and effective module design. The parameter quantity of FecNet is the smallest among the networks mentioned above, which is 2.53 M. This enables FecNet to achieve real-time detection on edge-computing devices with limited computing power.

V. Conclusion

In this study, aiming at improving the object detection performance of roadside LiDAR, a feature-enhancement and cascade network (FecNet) is proposed from the perspective of feature enhancement and feature cascade. The FFE module integrates foreground information into feature maps in different stages to differentiate objects from the complex background in traffic scenes by enhancing foreground feature discrimination. The feature cascade backbone associates object-correlated surroundings and preserves effective features in high-level feature maps to predict multiscale objects with limited characteristics. The results on the roadside dataset demonstrate that FecNet improves the performance of object detection in complex traffic scenarios and achieves the best tradeoff between accuracy and efficiency.

However, the scenarios in the established dataset are limited for further studies. More traffic scenes will be collected in future research to expand the dataset. It is expected to record more traffic data of multiscale objects with limited features and complex environments in different scenarios. On the expanded dataset, we will further improve the generalization and robustness of the proposed network to make 3-D object detection in ITS applications more reliable.

References

[1] S. Yang, M. Du, and Q. Chen, "Impact of connected and autonomous vehicles on traffic efficiency and safety of an on-ramp," *Simul. Model. Pract. Theory*, vol. 113, Dec. 2021, Art. no. 102374.

[2] A. Papadoulis, M. Quddus, and M. Imprialou, "Evaluating the safety impact of connected and autonomous vehicles on motorways," *Accident Anal. Prevention*, vol. 124, pp. 12–22, Mar. 2019.

[3] F. Rosique et al., "A systematic review of perception system and simulators for autonomous vehicles research," *Sensors*, vol. 21, no. 9, pp. 5668–5677, Feb. 2019.

[4] Z. Zhu, Z. Hu, W. Dai, H. Chen, and Z. Lv, "Deep learning for autonomous vehicle and pedestrian interaction safety," *Saf. Sci.*, vol. 145, Jan. 2022, Art. no. 105479.

[5] M. T. Rahman, K. Dey, S. Das, and M. Sherfinski, "Sharing the road with autonomous vehicles: A qualitative analysis of the perceptions of pedestrians and bicyclists," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 78, pp. 433–445, Apr. 2021.

[6] M. Hussain, N. Ali, and J.-E. Hong, "Vision beyond the field-of-view: A collaborative perception system to improve safety of intelligent cyber-physical systems," *Sensors*, vol. 22, no. 17, p. 6610, Sep. 2022.

[7] H. Xu, A. Berres, S. A. Tennille, S. K. Ravulaparthy, C. Wang, and J. Sanyal, "Continuous emulation and multiscale visualization of traffic flow using stationary roadside sensor data," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10530–10541, Aug. 2022.

[8] L. Wang, Z. Zhang, X. Di, and J. Tian, "A roadside camera-radar sensing fusion system for intelligent transportation," in *Proc. 17th Eur. Radar Conf. (EuRAD)*, Utrecht, The Netherlands, 2021, pp. 282–285.

[9] V. Vaquero, I. D. Pino, F. Moreno-Noguer, J. Solà, A. Sanfeliu, and J. Andrade-Cetto, "Dual-branch CNNs for vehicle detection and tracking on LiDAR data," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6942–6953, Nov. 2021.

[10] Z. Zhang, J. Zheng, H. Xu, X. Wang, X. Fan, and R. Chen, "Automatic background construction and object detection based on roadside LiDAR," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4086–4097, Oct. 2020.

[11] B. Li et al., "Enhancing 3-D LiDAR point clouds with event-based camera," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.

[12] P. Sun, C. Sun, R. Wang, and X. Zhao, "Object detection based on roadside LiDAR for cooperative driving automation: A review," *Sensors*, vol. 22, no. 23, p. 9316, Nov. 2022.

[13] X. Song et al., "Augmented multiple vehicles' trajectories extraction under occlusions with roadside LiDAR data," *IEEE Sensors J.*, vol. 21, no. 19, pp. 21921–21930, Oct. 2021.

[14] J. Zheng, S. Yang, X. Wang, Y. Xiao, and T. Li, "Background noise filtering and clustering with 3D LiDAR deployed in roadside of urban environments," *IEEE Sensors J.*, vol. 21, no. 18, pp. 20629–20639, Sep. 2021.

[15] L. Zhang, J. Zheng, R. Sun, and Y. Tao, "GC-Net: Gridding and clustering for traffic object detection with roadside LiDAR," *IEEE Intell. Syst.*, vol. 36, no. 4, pp. 104–113, Jul. 2021.

[16] H. Liu, C. Lin, B. Gong, and D. Wu, "Extending the detection range for low-channel roadside LiDAR by static background construction," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5702412.

[17] P. Sun, C. Sun, L. Wan, R. Wang, and X. Zhao, "Objects detection with 3-D roadside LiDAR under snowy weather," *IEEE Sensors J.*, vol. 22, no. 23, pp. 23051–23063, Dec. 2022.

[18] J. Chen, H. Xu, J. Wu, R. Yue, C. Yuan, and L. Wang, "Deer crossing road detection with roadside LiDAR sensor," *IEEE Access*, vol. 7, pp. 65944–65954, 2019.

[19] J. Wu, H. Xu, Y. Tian, R. Pi, and R. Yue, "Vehicle detection under adverse weather from roadside LiDAR data," *Sensors*, vol. 20, no. 12, p. 3433, Jun. 2020.

[20] J. Zhang, W. Xiao, B. Coifman, and J. P. Mills, "Vehicle tracking and speed estimation from roadside lidar," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5597–5608, Sep. 2020.

[21] J. Zhao, H. Xu, H. Liu, J. Wu, Y. Zheng, and D. Wu, "Detection and tracking of pedestrians and vehicles using roadside LiDAR sensors," *Transp. Res. C, Emerg. Technol.*, vol. 100, pp. 68–87, Mar. 2019.

[22] J. Wu, H. Xu, J. Zheng, and J. Zhao, "Automatic vehicle detection with roadside LiDAR data under rainy and snowy conditions," *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 1, pp. 197–209, Spring 2021.

[23] G. Wang, J. Wu, T. Xu, and B. Tian, "3D vehicle detection with RSU LiDAR for autonomous mine," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 344–355, Jan. 2021.

[24] A. Barrera, C. Guindel, J. Beltrán, and F. García, "BirdNet+: End-to-end 3D object detection in LiDAR bird's eye view," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.

[25] Z. Bai, G. Wu, M. J. Barth, Y. Liu, E. A. Sisbot, and K. Oguchi, "PillarGrid: Deep learning-based cooperative perception for 3D object detection from onboard-roadside LiDAR," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Macau, China, Oct. 2022, pp. 1743–1749.

[26] H. Shi, D. Hou, and X. Li, "Center-aware 3D object detection with attention mechanism based on roadside LiDAR," *Sustainability*, vol. 15, no. 3, p. 2628, Feb. 2023.

[27] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 77–85.

[28] R. Q. Charles, Y. Li, S. Hao, and G. J. Leonidas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 5105–6114.

[29] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4490–4499.

[30] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, Oct. 2018.

[31] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 12689–12697.

[32] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 10526–10535.

[33] S. Zhou, H. Xu, G. Zhang, T. Ma, and Y. Yang, "Leveraging deep convolutional neural networks pre-trained on autonomous driving data for vehicle detection from roadside LiDAR data," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 22367–22377, Nov. 2022.

[34] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[35] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 2443–2451.

[36] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[37] Z. Qin et al., "ThunderNet: Towards real-time generic object detection on mobile devices," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 6717–6726.
[38] Z. Zheng, Y. Zhong, J. Wang, and A. Ma, "Foreground-aware relation network for geospatial object segmentation in high spatial resolution remote sensing imagery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 4095–4104.
[39] Z. Gong, Z. Wang, B. Zhou, W. Liu, and P. Liu, "Pedestrian detection method based on roadside light detection and ranging," *SAE Int. J. Connected Automated Vehicles*, vol. 4, no. 4, pp. 413–422, Nov. 2021.
[40] T. Wu, S. Tang, R. Zhang, J. Cao, and Y. Zhang, "CGNet: A light-weight context guided network for semantic segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 1169–1179, 2021.
[41] C. Shen et al., "Multi-receptive field graph convolutional neural networks for pedestrian detection," *IET Intell. Transp. Syst.*, vol. 13, no. 9, pp. 1319–1328, May 2019.
[42] L. Jiao, S. Zhang, S. Dong, and H. Wang, "RFP-Net: Receptive field-based proposal generation network for object detection," *Neurocomputing*, vol. 405, pp. 138–148, Sep. 2020.
[43] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 936–944.
[44] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6230–6239.
[45] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[46] J. Chen, C. Wang, and Y. Tong, "AtICNet: Semantic segmentation with atrous spatial pyramid pooling in image cascade network," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–7, Jun. 2019.
[47] N. A. Mohamed, M. A. Zulkifley, and S. R. Abdani, "Spatial pyramid pooling with atrous convolutional for MobileNet," in *Proc. IEEE Student Conf. Res. Develop. (SCOReD)*, Batu Pahat, Malaysia, Sep. 2020, pp. 333–336.
[48] W. Luo, C. Wang, and Y. Tong, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Barcelona, Spain, 2016, pp. 4905–4913.
[49] H. Yu et al., "DAIR-V2X: A large-scale dataset for vehicle-infrastructure cooperative 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 21329–21338.
[50] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 770–779.
[51] S. Shi et al., "PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3D object detection," *Int. J. Comput. Vis.*, vol. 131, no. 2, pp. 531–551, Feb. 2023.
[52] G. Shi, R. Li, and C. Ma, "PillarNet: Real-time and high-performance pillar-based 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*, Tel Aviv, Israel, 2022, pp. 35–52.

Ziren Gong received the B.E. degree in traffic engineering from the Wuhan University of Technology, Wuhan, China, in 2020, and the M.E. degree with the School of Transportation Science and Engineering, Beihang University, Beijing, China, in 2023.

He is also with the Hefei Innovation Research Institute, Beihang University, Hefei, China. His research interests include computer vision and intelligent infrastructure.

Zhangyu Wang received the Ph.D. degree from Beihang University, Beijing, China, in 2021.

He is an Assistant Professor with the Research Institute for Frontier Science, and the State Key Laboratory of the Intelligent Transportation System, Beihang University, Beijing, China. His research interests include intelligent vehicles and computer vision.

Guizhen Yu received the Ph.D. degree from Jilin University, Changchun, China, in 2003.

He is currently a Professor with the School of Transportation Science and Engineering, Beihang University, Beijing, China. He is also with the Hefei Innovation Research Institute, Beihang University, Hefei, China. His research interests include intelligent vehicles, urban traffic operation, and intelligent transportation systems.

Wentao Liu received the degree in transportation science and engineering from Beihang University, Beijing, China, in 2020, where he is currently pursuing the Ph.D. degree with the School of Transportation Science and Engineering.

He is also with the Hefei Innovation Research Institute, Beihang University, Hefei, China. His research interests include multimodel perception and computer vision.

Songyue Yang received the M.S. degree in flight vehicle design from Beihang University, Beijing, China, in 2020, where he is currently pursuing the Ph.D. degree with the School of Transportation Science and Engineering.

He is also with the Hefei Innovation Research Institute, Beihang University, Hefei, China. His research interests include deep reinforcement learning and computer vision.

Bin Zhou received the Ph.D. degree from Beihang University, Beijing, China, in 2018.

He is an Assistant Professor with the Research Institute for Frontier Science and the State Key Laboratory of the Intelligent Transportation System, Beihang University, Beijing, China. His research interests include deep-learning algorithms, intelligent connected vehicles, big data analysis, and traffic network modeling.