SRS-YOLO: Improved YOLOv8-Based Smart Road Stud Detection

Guoqiang Mao, *Fellow, IEEE*, Keyin Wang, *Graduate Student Member, IEEE*, Haoyuan Du, Baoqi Huang, *Senior Member, IEEE*, Xiaojiang Ren, *Member, IEEE*, Tianxuan Fu, and Zhaozhong Zhang

***Abstract*— Smart road studs have been extensively deployed as road safety and data collection devices. Accurate and reliable detection of smart road studs and its further integration into the perception and control modules of connected and autonomous vehicles (CAVs) undoubtedly benefit road boundary detection, localization of CAVs and augument the safety of CAVs' driving. This work investigates real-time, accurate and reliable detection of smart road studs, which is a challenging task for CAVs because existing methods fail to achieve accurate and real-time smart road stud detection, especially in harsh road environment. To address these challenges, we first build a real-world smart road stud dataset, and then propose and validate a lightweight and efficient smart road stud detection model based on the you only look once 8th version (YOLOv8), called SRS-YOLO. First, a Squeeze-and-Excitation (SE) attention module is used to improve the coarse-to-fine (C2F) module to differentiate the channel importance of feature maps and improve the detection accuracy of smart road studs. Second, a novel downsampling module (DownS) that integrates the average pooling and the max pooling is designed to reduce the number of parameters and minimize information loss during the downsampling process. Third, the loss function is replaced with the Normalized Wasserstein Distance (NWD) loss to alleviate the sensitivity to location deviations when computing the loss for small targets. The experimental results demonstrate that the proposed SRS-YOLO outperforms other state-of-the-art methods, and achieves a 87.92% mean average precision at a real-time speed of 78 frames/s. Our dataset is available at: https://github.com/wky-xidian/smart-road-stud-dataset.**

***Index Terms*— SRS-YOLO, smart road stud detection, attention mechanism, real-time detection system, connected and autonomous vehicle.**

I. Introduction

ROAD studs have been extensively used for more than 80 years in a number of countries in a variety of applications [1], [2]. As early as 1930s, the UK began using road studs to mark road boundaries, lane directions, and intersections to enhance driving safety during nighttime and adverse weather conditions [3]. In the Netherlands, road studs are widely used as part of the road infrastructure. They serve various purposes, including: lane delineation, road edge marking, and intersection marking [4]. In the US and Canada, road studs are utilized to mark lane boundaries and lane directions on highways. They offer additional visual guidance during nighttime and low visibility conditions to help drivers stay in the correct lane [5]. In Malaysia, road studs are deployed at traffic light intersections and roundabouts to delineate traffic flow and guide drivers safely through these complex areas, reducing traffic accidents [6]. Fig. 1 shows some examples of road stud applications in different scenarios.

With the advancement of electronics, communication, sensing and solar technology, smart road studs integrating light emitting diode (LED) and various sensors, such as temperature, humidity, light, vibration and magnetic sensors, become feasible recently and are increasingly being applied in intelligent transportation systems [7]. LED lights embedded in road studs significantly improve visibility, especially in low-light conditions, such as fog, rain, or darkness - forward illumination can be increased from 100 meters to approx 900 meters [8]. As a reliable and ubiquitously deployed sensing device, smart road stud can achieve vehicle detection, wireless data transmission, and processing, which support digital twin systems [9]. Smart road studs can also be used to detect traffic accidents and interact with drivers by changing the light color to inform drivers of dangerous driving conditions ahead.

Accurate and reliable detection of smart road studs and its further integration into the perception and control modules of connected and autonomous vehicles (CAVs) is important. The advancement of CAV technologies has reached a stage where autonomous driving in benign environment becomes feasible but it remains a challenge in complex environment. The incorporation of smart road studs into the road-vehicle collaboration landscape may help to alleviate the challenge. First, accurate and reliable detection of smart road studs through CAV's onboard cameras can help CAVs to accurately detect the road/lane boundary lines [10]. This is especially important in harsh environment and poor weather conditions such as night, heavy rain, fog, etc. Second, by detecting traffic accidents and informing the CAVs through changing the color of smart road studs, the CAVs can be informed of hazardous road conditions beyond the line-of-sight detection range of CAV's onboard sensors [11]. Finally, the smart road

Received 19 June 2024; revised 23 November 2024 and 20 January 2025; accepted 21 February 2025. This work was supported by NSFC under Grant U21A20446. The Associate Editor for this article was H. Huang. *(Corresponding author: Keyin Wang.)*

Guoqiang Mao is with the School of Transportation, Southeast University, Nanjing 210096, China (e-mail: g.mao@ieee.org).

Keyin Wang, Haoyuan Du, Xiaojiang Ren, and Tianxuan Fu are with the Research Institute of Smart Transportation and the School of Telecommunications Engineering, Xidian University, Xi'an 710071, China (e-mail: keyinwang@stu.xidian.edu.cn; 23013221187@stu.xidian.edu.cn; xjren@xidian.edu.cn; futianxuan@stu.xidian.edu.cn).

Baoqi Huang is with the Engineering Research Center of Ecological Big Data, Ministry of Education, and Inner Mongolia Key Laboratory of Wireless Networking and Mobile Computing, College of Computer Science, Inner Mongolia University, Hohhot 010021, China (e-mail: cshbq@imu.edu.cn).

Zhaozhong Zhang is with the School of Information Engineering, Nanchang Hangkong University, Nanchang 330063, China (e-mail: 71033@nchu.edu.cn).

Digital Object Identifier 10.1109/TITS.2025.3545942

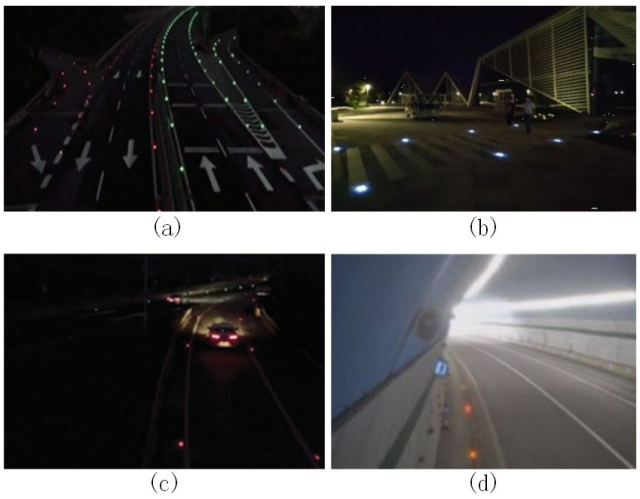


Fig. 1. Examples of road stud applications in different scenarios. (a) highway, (b) crosswalk, (c) ramp-merge, and (d) tunnel.

studs can also serve as landmarks and assist the lane-level localization of CAVs, especially in global navigation satellite system (GNSS)-denied environment or in environment with a lack of landmarks [7].

This paper investigates real-time, accurate and reliable detection of smart road studs via vehicle onboard cameras. Ideally, smart road stud detection needs to meet the following requirements:

1) Accuracy: it is important to correctly identify and locate smart road studs within images or video frames to enhance the ability of CAVs to perceive their surrounding environment.

2) Real-time performance: for CAV applications, the ability to identify smart road studs quickly is obviously an important requirement [12].

3) Robustness: for a fast moving CAV, the images of smart road studs captured by onboard cameras can be complex and time-varying. Robust detection algorithms are essential for the safe, reliable, and effective operation of CAVs in diverse and dynamic environments [13].

4) Lightweight: considering the limited computing resources of onboard devices and energy consumption constraints, a lightweight detection model is more desirable [14].

For fast moving CAVs, smart road studs are relatively small (occupying an area less than and equal to 1024 pixels [15], [16]), with inconsistent brightness and often blurry backgrounds. These factors make smart road stud detection a challenging task. Traditional object detection algorithms based on handcrafted features struggle to address the aforementioned challenges in achieving accurate and robust smart road stud detection [17], [18]. Since the emergence of AlexNet [19] in 2012, deep neural network (DNN)-based detection algorithms have been dominating the entire field of object detection [20], [21], [22], [23]. As DNNs have the remarkable capability to automatically learn and extract semantic, high-level, and deep features from images, this eliminates the need for complex handcrafted feature identification. They can also handle complex scenarios and diverse visual appearances. Based on these advantages, we are considering DNN-based object detection algorithm for smart road stud detection.

Today's popular DNN-based object detection algorithms can be broadly divided into two categories: two-stage and one-stage detectors. Two-stage detectors first identify regions of interest and then refine and classify them in separate steps, while one-stage detectors directly identify bounding boxes and class probabilities in a single step [24]. Typical two-stage detectors include region-based convolutional neural network (R-CNN) [20], Fast R-CNN [21], Faster R-CNN [25], and Mask R-CNN [26], etc., while one-stage detectors include single shot multibox detector (SSD) series [23] and you only look once (YOLO) series [22]. It is noteworthy that in the development of object detection algorithms, YOLO algorithms have become increasingly popular due to their accuracy and speed. YOLO algorithms are suitable for detecting general objects. There is still significant room for improvement in detection accuracy, real-time performance, robustness, and model complexity when it comes to detecting small objects like smart road studs.

In this paper, we propose a new smart road stud detection method based on YOLOv8 algorithm, referred to as SRS-YOLO. Specifically, a Squeeze-and-Excitation (SE) attention module is added as part of coarse-to-fine (C2F) module to enhance the detection accuracy of smart road studs by distinguishing the importance of different channels in the feature maps at the cost of a modest increase in model complexity. Moreover, a new downsampling structure (DownS) is designed, combining the average pooling and the max pooling, to reduce the loss of smart road stud-related features due to downsampling. Compared with the original convolutional downsampling method, DownS greatly reduces the number of parameters. To further improve small target recognition, the Normalized Wasserstein Distance (NWD) loss is applied during the model training process, which can alleviate the sensitivity to location deviations when computing the loss for small targets, thereby improving the model's adaptability to detecting smart road studs. Because there are no existing smart road stud datasets, we first build a dataset containing smart road stud images to train and test machine learning-based smart road stud detection algorithms. Finally, we deploy the trained smart road stud detection model on an experimental vehicle to validate the effectiveness of the proposed algorithm. The developed algorithm not only is useful for smart road stud detection but also can be applied to the more general problem of small target detection. The following is a list of the main contributions of this paper:

1) Considering the balance between accuracy and model complexity, a Squeeze-and-Excitation attention module is incorporated into the C2F module to distinguish the importance of feature maps on a channel-wise basis, ensuring real-time performance while enhancing smart road stud detection accuracy, especially in inconsistent brightness and blurry backgrounds. Additionally, the introduction of the Normalized Wasserstein Distance loss function, which measures the similarity between predicted boxes and ground truth boxes using the Wasserstein distance, improves the model's robustness to detecting small targets like the smart road studs.

2) A new downsampling module, i.e., DownS, is developed, which combines the average pooling and max pooling.

Compared to the original convolutional downsampling, DownS reduces information loss during downsampling process, improves smart road stud detection accuracy, and reduces the number of model parameters, thus leading to a lightweight model, which is beneficial for real-world deployment. Experiments conducted on the dataset validate the effectiveness of the proposed DownS module.

3) A real-time smart road stud detection system is developed and implemented on an experimental vehicle to validate the feasibility and effectiveness of the proposed algorithm, which demonstrates superior performance.

The paper is organized as follows. Section II introduces related work. Section III provides details on the network structure of the SRS-YOLO model and the related modules. Section IV introduces the experimental details, including dataset preparation, model evaluation, and comparative experiments. Finally, conclusions and future work are drawn in Section V.

II. Related Work

The inaugural work of YOLO series is YOLOv1 proposed by R. Joseph et al. in 2015, which is the first DNN-based one-stage object detection model [22]. The main advantage of this model over other models developed during the same time period is its extremely fast detection speed. Based on the YOLOv1 framework, a series of versions have been proposed [27], [28], [29]. The main differences between different versions of the YOLO series lie in improvements and optimizations in network architecture, balancing accuracy and speed, feature extraction methods, handling of object sizes, data augmentation and regularization, loss function design, etc. Even though YOLOv9 [30] and YOLO-World [31] have been released recently, considering the requirement for model stability in real-world applications, we chose the more mature YOLOv8. The YOLO series have achieved excellent performance in general object detection. However, they face significant challenges when it comes to detecting small objects in certain scenarios [32]. To address this issue, many improved algorithms based on YOLO have emerged [15], [33], [34]. In this section, we will focus on the most relevant researches aimed at enhancing the performance of the YOLO model through attention mechanisms, loss functions and lightweight strategies.

A. Improvements of YOLO Based on Attention Mechanisms

Attention mechanisms are commonly used in machine learning and have their conceptual basis drawn from research on human vision [35]. The main principle of attention mechanism can be summarized as assigning different weights to different parts of the input data so as to enhance the model's focus on key information. Typical attention mechanism modules include SE [36], Channel Attention (CA) [37], Convolutional Block Attention Module (CBAM) [38], Efficient Multi-Scale Attention (EMA) [39], and so on. An important approach to improving the small object detection performance of YOLO is to incorporate attention mechanism modules into the original model.

Sun et al. [40] proposed an improved YOLOv5 network to deal with inner wall defects, in which a SE module is added between the network's backbone and neck to improve the feature extraction efficiency of small objects. Peng et al. [41] introduced a multiscale feature fusion lightweight-YOLO for remote sensing image detection. In this model, they incorporated a CA module into the feature fusion network. This addition enables the network to capture direction- and location-aware information across channels simultaneously, thereby enhancing the detection accuracy. Peng et al. [42] proposed a tire detection approach based on YOLO, in which a CBAM module is added to improve the detection accuracy of small tire defects. Dai et al. [43] combined YOLO and vision transformer to realize automatic detection of foreign objects between platform screen doors and metro train doors. Although incorporating attention modules into YOLO can improve the model's accuracy to some extent, adding attention modules increases the model's parameters, leading to increased model complexity and reduced efficiency. This is not acceptable in CAV applications where real-time performance is crucial. Therefore, when using attention mechanisms to improve the YOLO models, it is important to strike an optimum balance between the detection accuracy and the model complexity.

B. Loss Functions in YOLO

The loss function is a key component in machine learning, directly impacting the training process [44]. The selection of an appropriate loss function is crucial for improving the model performance.

YOLO originally utilized the intersection over union (IoU) loss function [45], which is a commonly used loss function in object detection tasks. The IoU loss function measures the accuracy of object localization by comparing the overlap between the predicted bounding box and the ground truth bounding box. Rezatofighi et al. [46] introduced the Generalized IoU, which addresses IoU's plateau issue, particularly in scenarios involving non-overlapping bounding boxes. To expedite model convergence during training, the Distance-IoU algorithm was proposed by incorporating the normalized distance between the predicted bounding box and the ground truth bounding box. Furthermore, considering three geometric factors in the bounding box regression simultaneously, i.e., the overlap area, the central point distance, and the aspect ratio, the Complete IoU loss was proposed, which has faster convergence and better performance [47]. In addition to the aforementioned loss functions, loss functions such as Efficient IoU [48], alpha IoU [49], and Wise IoU [50] have also been proposed and applied in YOLO. However, IoU-related loss functions may lead to poor model performance in small object detection tasks because of the imbalance between positive and negative samples during training. Therefore, there is a need to improve model performance from the perspective of loss function.

C. Lightweight Strategies in the YOLO

In many practical applications, object detection models need to run in real-time on edge devices with limited resources.

This requires the model to reduce the number of parameters and computational complexity while maintaining accuracy, in order to improve inference speed and conserve hardware resources [51].

In recent years, many studies have proposed improved YOLO models based on various lightweight strategies. Peng et al. [41] proposed using deep separable convolution to replace the standard convolution layer in the backbone network of the YOLO model, in order to reduce the number of model parameters in remote sensing image detection model. Tang et al. [52] also proposed using deep separable convolution in the object detection model for antenna interference sources to reduce the number of parameters in the YOLO model. Ning et al. [53] reduced the number of model parameters of pavement distress detection model by replacing ordinary convolution with distribution shifting convolution, which decomposes the traditional convolution kernel into two parts: variable quantized kernel and distributed shift. Guan et al. [54] designed a modified one-shot aggregation block with an attention mechanism and integrated it into the railway obstacle detection network to reduce the number of model parameters. Although the aforementioned lightweight strategies reduce the number of model parameters by employing different convolution forms or module structures, they also diminish the feature representation capability, resulting in a decline in object detection accuracy.

III. The Proposed SRS-YOLO Algorithm

Although YOLOv8 has shown excellent performance in various object detection tasks, when it comes to the detection of smart road studs, issues such as blurry backgrounds, varying brightness, and small target areas can cause frequent false positives and missed detection. Therefore, it is important to improve the performance of YOLOv8 for smart road stud detection.

A. Overall Framework of SRS-YOLO

Fig. 2 illustrates the framework of the proposed SRS-YOLO. There are three basic modules, namely the backbone module, the neck module and the head module, where the backbone module is responsible for extracting features from the input, the neck module is used to integrate features from different scales, and the head module outputs detection results.

1) Backbone: The backbone of SRS-YOLO comprises Conv, C2F, DownS, and the spatial pyramid pooling-fast (SPPF) modules. For the Conv module, there are three submodules, which are the two-dimensional convolution (Conv2d), the batch normalization (BN), and the sigmoid linear unit (SiLU). The SPPF module consists of Conv, maxpooling (MaxPool2d) and Concat modules. The DownS module is a new downsampling module proposed in this paper. The following section will provide a detailed introduction to the C2F and DownS modules. Given the input smart road stud image $\boldsymbol{I} \in \mathbb{R}^{H\times W\times C}$, where H, W, and C are the heigh, width, and the number of channels of the input image, respectively. According to YOLOv8's parameter settings, H and W are both 640 pixels, C is 3. The image then passes through the backbone to complete smart road stud-related features extraction.

TABLE I
Feature Map Size After Processing by Different Modules

Location	Feature Map Size	Location	Feature Map Size
1	$640 \times 640 \times 3$	13	$40 \times 40 \times 384$
2	$320 \times 320 \times 16$	14	$40 \times 40 \times 128$
3	$160 \times 160 \times 32$	15	$80 \times 80 \times 128$
4	$160 \times 160 \times 32$	16	$80 \times 80 \times 192$
5	$80 \times 80 \times 64$	17	$80 \times 80 \times 64$
6	$80 \times 80 \times 64$	18	$40 \times 40 \times 64$
7	$40 \times 40 \times 128$	19	$40 \times 40 \times 192$
8	$40 \times 40 \times 128$	20	$40 \times 40 \times 128$
9	$20 \times 20 \times 256$	21	$20 \times 20 \times 128$
10	$20 \times 20 \times 256$	22	$20 \times 20 \times 384$
11	$20 \times 20 \times 256$	23	$20 \times 20 \times 256$
12	$40 \times 40 \times 256$		

2) Neck: The neck of SRS-YOLO consists of multiple Upsample, Concat, C2F-SE, and DownS modules. the C2F-SE module is obtained by adding the SE module to the C2F module. In the neck module, based on the different scale feature maps extracted from the backbone, a feature pyramid network (FPN) is used to construct a top-down feature pyramid to achieve an initial fusion of multi-scale features. On top of the FPN, the path aggregation network (PAN) bottom-up path is used to pass lower-level features to higher levels, further enriching the multi-scale features. Finally, through skip connections, features from different scales are fused to ensure that each layer contains rich contextual information, enhancing the model's ability to detect objects at different scales.

3) Head: The head of SRS-YOLO is the same as that of YOLOv8. It decouples the bounding box regression loss (Bbox Loss) and the classification loss (Cls Loss), and enhances the training stability and detection accuracy of the model. Specifically, the parameter c is the number of detection types, the number 5 represents the four coordinates (x, y, h, w) and confidence, where x and y denote the center point coordinates, and h and w denote height and width of the predicted bounding box, respectively.

The dimensions of the feature maps after processing through different modules are shown in Table I, and the locations in the table are provided in Fig. 2.

In the basis of YOLOv8, SRS-YOLO introduces SE attention module that allows the model to focus on smart road stud-related features under inconsistent brightness and blurry backgrounds, thereby improving the accuracy of smart road stud detection. The DownS module is designed to reduce information loss during downsampling process while also reducing the number of model parameters. Additionally, the NWD loss function is used during training model to enhance the model's robustness in detecting small targets like the smart road studs.

B. C2F-SE Module

The SE module enhances the network's representational capability by facilitating dynamic recalibration of features on a channel-wise basis [36]. Specifically, as shown in Fig. 3, the SE is divided into the following three steps.

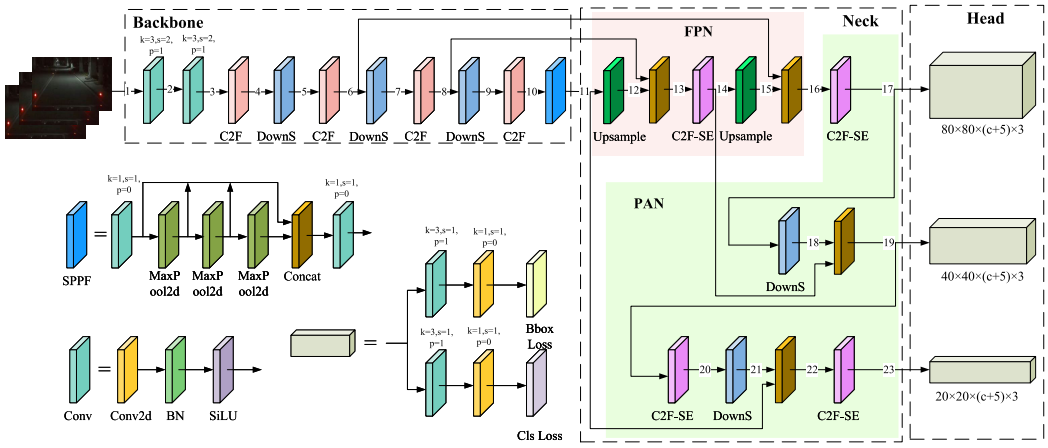


Fig. 2. An illustration of the framework of the proposed SRS-YOLO. The symbols k, s, and p respectively represent the kernel size, stride, and padding in a convolution operation.

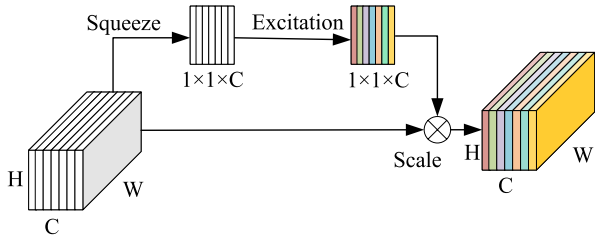


Fig. 3. Architecture of the SE module.

1) *Squeeze*: By performing the average pooling (AvgPool2d), the feature map $\boldsymbol{U}$ with size $H \times W \times C$ is transformed into a $1 \times 1 \times C$ feature vector $\boldsymbol{z}$, aiming to increase the model's global receptive field and extract richer features:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j) \tag{1}$$

where z_c and $\boldsymbol{u}_c$ are the c-th elements of $\boldsymbol{z}$ and $\boldsymbol{U}$, respectively.

2) *Excitation*: To utilize the information aggregated during the *Squeeze* operation, the *Excitation* operation is employed to comprehensively capture channel-wise dependencies, which employs two fully connected (FC) layers to reduce and then increase the dimensionality, aiming to effectively integrate channel information. The channel weights of the feature map can be obtained through the *Excitation* operation as follows:

$$\boldsymbol{s} = \sigma(\boldsymbol{W}_2 \delta(\boldsymbol{W}_1 \boldsymbol{z})) \tag{2}$$

where $\boldsymbol{s}$ is the weight vector of the feature map $\boldsymbol{U}$, σ refers to the Sigmoid function, δ refers to the Rectified Linear Unit (ReLU) function, $\boldsymbol{W}_1$ is the parameter vector of the dimentionality-reduction layer with reduction ratio R, and $\boldsymbol{W}_2$ is the parameter vector of the dimentionality-increasing layer with an increasing ratio R.

3) *Scale*: The *Scale* operation is responsible for multiplying channel weight and feature map as follows:

$$\boldsymbol{x}_c = s_c \boldsymbol{u}_c \tag{3}$$

where $\boldsymbol{x}_c$ refers to the c-th new feature map, s_c is the c-th channel weight.

The SE attention module does not change the size of the feature map. However, by introducing the SE attention mechanism, the model can dynamically adjust the importance of different channels in the feature map, allowing the network to focus more on important features, thereby improving the model's detection accuracy.

Considering the simplicity and efficiency of the SE module, it is better suited for real-time smart road stud detection. The SE module can enhance the detection capability of smart road studs by adaptively adjusting the channel weights of feature maps under different lighting conditions and complex backgrounds. This helps to better highlight the features of smart road studs in blurred images, thus improving the model's robustness and accuracy. Therefore, we incorporate the SE module into the C2F module, producing the new C2F-SE module. By replacing the C2F module in the backbone of YOLOv8 with the C2F-SE module, the network can more accurately locate and identify the target of interest while avoiding excessive computational overhead. The architecture of the C2F-SE is shown in Fig. 4. In the C2F-SE module, the input feature map is evenly divided into two parts along the channel dimension after passing through the Conv module. One part of the feature map after the Split is processed layer by layer through multiple Darknet Bottleneck (DB) modules to extract deeper features. The Concat operation concatenates the output of all DB modules and the other part of the previously split feature map along the channel dimension, enhancing feature diversity. The concatenated feature map is then input into the SE module to capture more smart road stud-related features. The C2F-SE module increases the depth of the network by stacking multiple DB blocks containing two Conv modules, without significantly increasing the computational overhead.

C. DownS Module

In both the backbone and the neck modules of YOLOv8, downsampling operations are directly implemented through Conv, which leads to a significant increase in the number of parameters. At the same time, downsampling through Conv results in information loss in the feature maps, reducing their resolution and making smart road stud-related features coarser, thus affecting the accuracy of smart road stud detection.

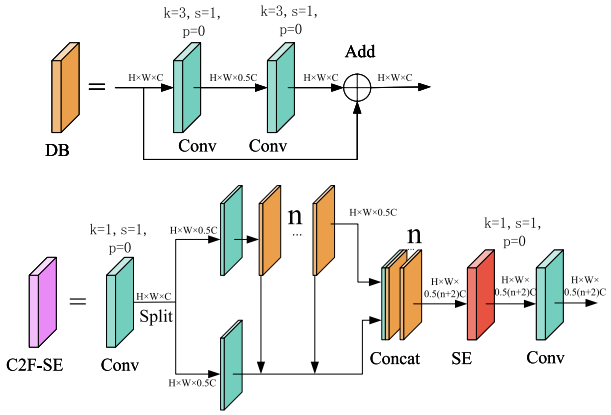


Fig. 4. Architecture of the C2F-SE.

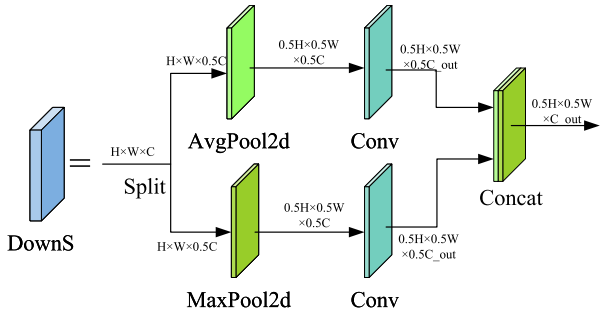


Fig. 5. Architecture of the DownS.

Inspired by the downsampling method in YOLOv9 [30], we design a new downsampling approach, i.e., DownS, to solve the challenge, as illustrated in Fig. 5.

The operation process of DownS can be described as follows: The input feature map with size $H \times W \times C$ is first split into two parts along the channel dimension. One part goes through AvgPool2d and Conv with a convolutional kernel of 1×1, while the other part goes through MaxPool2d and Conv with a convolutional kernel of 3×3, resulting in two identical feature maps of size $0.5H \times 0.5W \times 0.5C\_out$. Finally, the two parts are concatenated to obtain the downsampled feature map with size $0.5H \times 0.5W \times C\_out$**.** This module reduces the number of channels passed through the Conv module through Split operation, thereby reducing the number of parameters. Additionally, it integrates the average pooling and the max pooling to reduce information loss during downsampling process.

To quantitatively measure the performance of the DownS module in reducing model parameters, we take the fourth layer of the YOLOv8 model's backbone module as an example and calculate the number of parameters for both the convolutional downsampling and the DownS downsampling methods. The feature map with dimension $160 \times 160 \times 32$, after passing through the downsampling module of the fourth layer, results in a feature map of dimension $80 \times 80 \times 64$. For the convolutional downsampling method with a convolutional kernel of 3×3, the number of parameters is $(3 \times 3 \times 32 + 1) \times 64 + 64 = 18560$, and for the DownS method, the input feature map of size $160 \times 160 \times 32$ is split into two parts of size $160 \times 160 \times 16$ each, one part undergoes AvgPool2d with a window size of 2×2 and a stride of 2, followed by a 1×1 convolution with 32 output channels, resulting in a feature map of size $80 \times 80 \times 32$, the other part goes through MaxPool2d with a window size of 2×2 and a stride of 2, followed by a 3×3 convolution with 32 output channels, also resulting in a feature map of size $80 \times 80 \times 32$. The two feature maps are then concatenated to produce a final feature map of size $80 \times 80 \times 64$**,** the number of parameters is $(1 \times 1 \times 16 + 1) \times 32 + 32 + (3 \times 3 \times 16 + 1) \times 32 + 32 = 524$.

D. Loss Function

The IoU-based metrics are highly sensitive to variations in small objects, where even slight location deviations can lead to significant drop in the value of IoU, resulting in inaccurate label assignments. Consequently, the IoU-based loss functions are not ideal metrics for the detection of smart road studs. In order to improve the performance of smart road stud detection, We replace the default localization loss function, the Complete IoU loss function, in YOLOv8 with the NWD loss function [55], a metric specifically designed for small objects.

Because smart road studs are not standard rectangles, there are often some background pixels in their bounding boxes, with the foreground pixels concentrated on the center of the bounding boxes and the background pixels concentrated on the boundary of the bounding boxes. To better describe the weights of different pixels in the bounding box, the bounding box is modeled as a two-dimensional Gaussian distribution. The center coordinate of the bounding box serves as the center point of the Gaussian distribution, and the width and height of the bounding box are used as the length and width of the Gaussian distribution. Specifically, for a horizontal bounding box, the equation of its inscribed ellipse can be represented as follows:

$$\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} = 1 \tag{4}$$

where (μ_x, μ_y) is the center of the inscribed ellipse, σ_x and σ_y are the lengths of semi-axises along x and y axises, respectively, $\mu_x = c_x$, $\mu_y = c_y$, $\sigma_x = w/2$, $\sigma_y = h/2$, (c_x, c_y) is the center of the bounding box, w and h are the width and height of the bounding box, respectively.

The probability density function of a two-dimensional Gaussian distribution can be described as follows:

$$f\left(\mathrm{x}|\boldsymbol{\mu}, \sum\right) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \sum^{-1} (\mathbf{x}-\boldsymbol{\mu})\right)}{2\pi|\sum|^{\frac{1}{2}}} \tag{5}$$

where $\mathbf{x}$, $\boldsymbol{\mu}$, and $\sum$ are the coordinate (x, y), the mean vector, and the co-variance matrix of the Gaussian distribution, respectively. When $(\mathbf{x}-\boldsymbol{\mu})^T \sum^{-1} (\mathbf{x}-\boldsymbol{\mu}) = 1$, the ellipse represented by (4) will be a density contour of the two-dimensional Gaussian distribution, the bounding box (c_x, c_y, w, h) can be modeled as a two-dimensional Gaussian distribution $N(\boldsymbol{\mu}, \sum)$ with

$$\mu = \begin{bmatrix} c_x \\ c_y \end{bmatrix}, \sum = \begin{bmatrix} \frac{w^2}{4} & 0 \\ 0 & \frac{h^2}{4} \end{bmatrix} \tag{6}$$

The similarity between the ground truth bounding box $(c_{x_g}, c_{y_g}, w_g, h_g)$ and the predicted bounding box

$(c_{x_p}, c_{y_p}, w_p, h_p)$ can quantified by the distance between two Gaussian distributions, which is calculated using the 2^{nd} Wasserstein distance [56] as follows:

$$D_2^2(N_p, N_g) = \| \left([c_{x_p}, c_{y_p}, w_p, h_p]^T, [c_{x_g}, c_{y_g}, w_g, h_g]^T \right) \|_2^2 \quad (7)$$

where N_p and N_g are the Gaussian distributions of the predicted bounding box and ground truth bounding box, respectively. Using its exponential form normalization, a new metric dubbed NWD is obtained as follows:

$$NWD(N_p, N_g) = \exp\left(-\frac{\sqrt{D_2^2(N_p, N_g)}}{C} \right) \quad (8)$$

where C is a constant, selected based on empirical experience. In our study, $C = 12$.

The NWD metric is chosen as the loss function:

$$L_{NWD} = 1 - NWD(N_p, N_g) \quad (9)$$

Due to the varying distances between smart road studs and the vehicle, their sizes and shapes appear inconsistent in the image. Most smart road studs, being farther from the vehicle, also appear smaller in the image. These factors make it difficult for the model to converge when trained using the IoU-based loss. NWD loss models the bounding boxes as two-dimensional Gaussian distributions and uses the Wasserstein distance to measure the similarity between bounding boxes. The Wasserstein distance can assess the similarity between distributions even when there is little or no overlap, and NWD is insensitive to the scale of objects. Therefore, NWD loss is more suitable than IoU-based loss for training the smart road stud detection model.

IV. Experiments and Analysis

A. Dataset Establishment

For smart road stud detection, there are no datasets available for training and testing deep learning models at present. To overcome this limitation, a smart road stud dataset is developed. The smart road stud images are captured using a visual camera (Stereolabs, ZED 2i). The camera operates at a frame rate of 30 FPS and has an image resolution of 1280 × 720 with a field of view of 90° × 60°. An example of the data collection scenario is shown in Fig. 6. Smart road studs are deployed along both sides of road with the distance between smart road studs set to 15 meters. A total of 2360 images of smart road studs are captured by the camera installed on the vehicle. Some samples of smart road stud dataset are shown in Fig. 7. The software LabelImg is used to mark the labels and coordinates of the smart road studs in images to obtain ground truth. Finally, the dataset is randomly divided into three sets: the training set, the validation set, and the test set with a ratio of 6: 2: 2.

B. Workstation Configuration and Hyperparameters Setting for Training Models

The workstation configuration and model hyperparameters are shown in Table II. For training models, the Stochastic Gradient Descent optimizer is used and the learning rate is updated by cosine annealing.

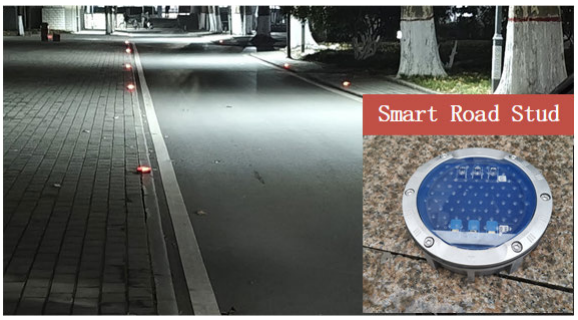


Fig. 6. Smart road studs and their deployment.

Fig. 7. Samples of the smart road stud dataset.

TABLE II
Workstation Configuration and Model Hyperparameters

Workstation Configuration	
CPU	Intel(R) Core(TM) i7-10700 @ 2.90GHz
GPU	NVIDIA GeForce GTX 1660 SUPER
Memory	16GB
Deep Learning Framework	PyTorch
Model Hyperparameters	
Epochs	200
Image Size	640×640
Training Batch Size	16
Initial Learning Rate	0.01
Final Learning Rate	0.0001
Momentum	0.937
Weight Decay	0.0005

C. Evaluation Metrics

To evaluate the performance of SRS-YOLO, the metrics selected include mean average precision (mAP), frames/s (FPS), the number of parameters, and giga floating-point operations (GFLOPs) which is used to measure the complexity of the model [57]. The calculation formulas of precision (P), recall (R), and mAP are as follows:

$$\begin{cases} P = \dfrac{TP}{TP + FP} \\ R = \dfrac{TP}{TP + FN} \\ mAP = \dfrac{\sum_{n=1}^{N} AP(n)}{N} \end{cases} \quad (10)$$

TABLE III
COMPARISON OF DIFFERENT ATTENTION MODULES

Modules	mAP	GFLOPs	Parameters (Million)	FPS	Score
C2F-EMA	0.8698	9.6	3.08	67	0.3337
C2F-CBAM	0.8693	8.6	3.48	77	0.5270
C2F-CA	0.8583	8.3	3.06	69	0.5301
C2F-SE	0.8675	8.2	3.07	82	0.9769

TABLE IV
COMPARISON OF DIFFERENT POSITIONS OF C2F-SE

Position	mAP	GFLOPs	Parameters (Million)	FPS
Backbone	0.8541	8.2	3.04	66
Neck	0.8700	8.2	3.04	83
All	0.8675	8.2	3.07	82

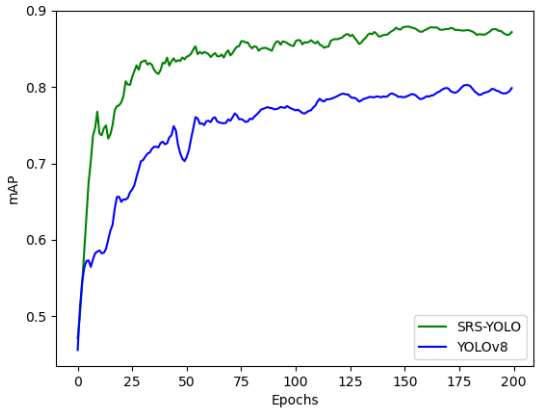


Fig. 8. Variation of the mAP with increasing epoch index.

where TP, FP, and FN are the number of true-positive cases, false-positive cases, and false-negative cases, respectively, $Ap = \int_0^1 PdR$, N is the number of detection types. In this study, $N = 1$.

In addition, technique for order preference by similarity to an ideal solution (TOPSIS) is used to score the various algorithms, which is a within-group comprehensive evaluation method [43]. The specific calculation steps of TOPSIS are as follows: First, construct a decision matrix based on the selected indicators. Then, normalize the decision matrix considering the benefit indicators and cost indicators. Next, perform a weighted treatment on the normalized matrix according to the weights of each attribute. After that, calculate the Euclidean distance between each alternative in the weighted matrix and the ideal solution (the maximum value of each attribute) as well as the negative ideal solution (the minimum value of each attribute). Finally, by calculating the ratio of the distance from each alternative to the negative ideal solution and the distance to the ideal solution, a performance score is obtained. We use mAP, FPS, the number of parameters, and GFLOPs to calculate the TOPSIS score, with weights set at 40%, 20%, 20% and 20%, respectively. Among them, mAP and FPS are benefit indicators, while the number of parameters and GFLOPs are cost indicators.

D. Comparison of Different Attention Modules

To compare the effects of different attention modules on the performance of smart road stud detection, we incorporate EMA, CBAM, CA, and SE modules into C2F separately, resulting in C2F-EMA, C2F-CBAM, C2F-CA, and C2F-SE. We respectively adopt these modules to replace the C2F module in YOLOv8, resulting in different models. Then, we use TOPSIS scores to compare these models. The experimental results are shown in Table III. EMA and CBAM consider both spatial and channel dimensions of feature maps, which enables more accurate recognition of smart road studs. In contrast, CA only considers spatial dimension, while SE only considers channel dimension. Therefore, the mAP of C2F-EMA and C2F-CBAM is higher than that of C2F-CA and C2F-SE. However, the C2F-EMA and C2F-CBAM models are complex, leading to higher computational complexity and more parameters. Compared to C2F-CA, C2F-SE has slightly more parameters, but its advantages lie in higher mAP, lower model complexity, and faster detection speed. These advantages make C2F-SE more suitable for smart road stud detection in autonomous driving scenarios. According to the TOPSIS scores, C2F-SE also has the highest score.

E. Application of C2F-SE

The SE module enhances the model's representational capacity, but this may lead the model to focus too much on certain local features while overlooking the overall shape or contour information, which could affect the overall detection performance of smart road studs. The backbone and neck of YOLOv8 both contain C2F. To determine the optimal usage scheme for the SE module, we replace the C2F in the backbone or neck with C2F-SE, and the resulting model outcomes are shown in Table IV. It is evident that when only the C2F module in the neck is replaced with C2F-SE, the model's performance becomes optimal. This is because applying the SE module within the backbone, which primarily extracts basic features, could potentially interfere with capturing essential low-level spatial information. In contrast, the neck is primarily responsible for fusing multi-scale features extracted by the backbone. Placing the SE module in the neck enables adaptive re-weighting of features after integration, allowing the model to emphasize features associated with specific targets (e.g., smart road studs) without disrupting low-level spatial accuracy.

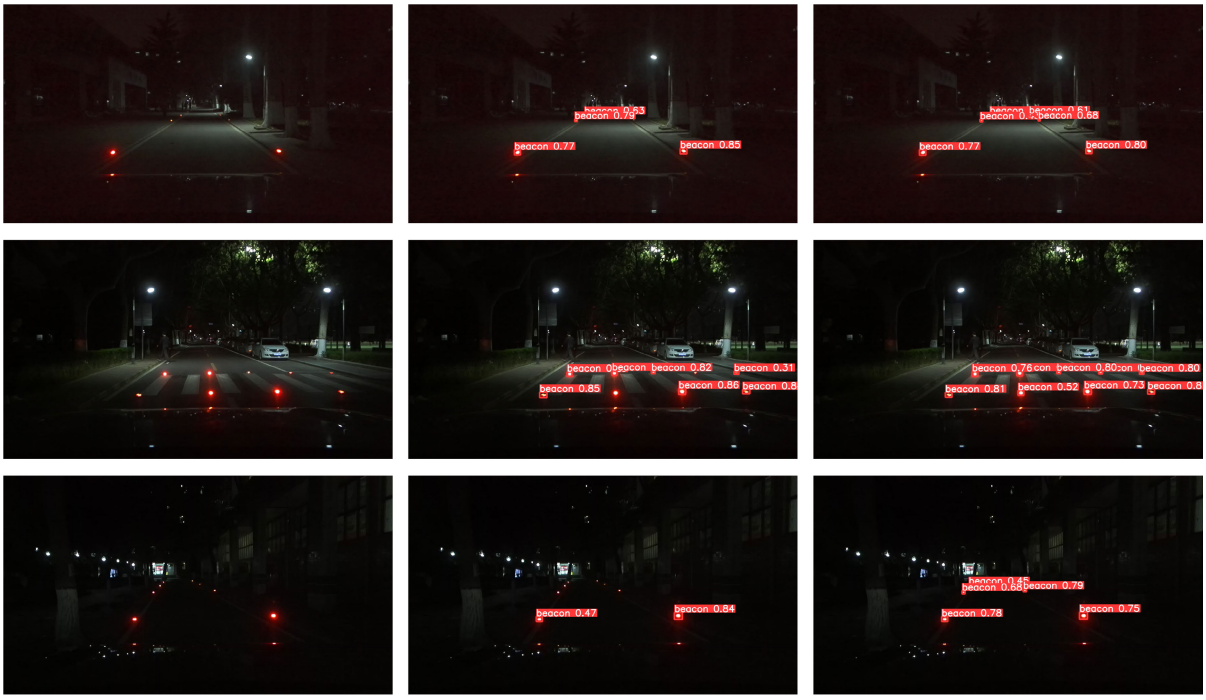
F. Analysis of Ablation Experiments

In order to demonstrate the effectiveness of the proposed strategies on smart road stud detection, ablation experiments are conducted on the smart road stud dateset. The evaluation metrics include mAP, GFLOPs, the number of parameters and FPS. Additionally, we have also calculated the TOPSIS scores for the different improvement methods.

From Table V, it can be seen that compared to the original model, the three proposed improvement methods in this paper all result in an increase in mAP, and the degree of improvement is similar. C2F-SE incorporates an SE module into C2F, leading to an increase in the number of parameters. However, the SE module enhances the network's representational capability by dynamically recalibrating features on

TABLE V
PERFORMANCE COMPARISON OF THE MODELS WITH DIFFERENT IMPROVEMENT STRATEGIES

C2F-SE	DownS	NWD	mAP	GFLOPs	Parameters (Million)	FPS	Score
			0.7971	8.2	3.01	80	0.3478
√			0.8700	8.2	3.04	83	0.5828
	√		0.8627	7.5	2.60	69	0.5653
		√	0.8686	8.2	3.01	80	0.5577
√	√	√	0.8792	7.5	2.63	78	0.8137



(a) (b) (c)

Fig. 9. Comparison of detection results in different scenarios. (a) Original images. (b) Detection results of YOLOv8. (c) Detection results of SRS-YOLO. (During dataset annotation, we labeled "smart road stud" as "beacon".)

Fig. 10. The experimental vehicle with the visual camera and industrial control computer.

a channel-wise basis. This enables the network to more accurately locate and identify smart road studs. As a result, C2F-SE significantly improves mAP, but there is no increase in GFLOPs. DownS reduces parameters during downsampling by dividing the feature map into two parts. It also combines average pooling and max pooling to minimize feature loss of smart road studs. Therefore, DownS increases mAP, reduces model complexity, and decreases the number of parameters. The NWD loss function measures the similarity between two bounding boxes based on Wasserstein distance and is insensitive to the scale of the targets, enhancing the detection accuracy for small targets such as smart road studs. However, NWD does not affect the model complexity or the number of parameters. By simultaneously employing C2F-SE, DownS, and NWD, the model achieves superior detection performance, mAP is increased by 10.30%, GFLOPs is reduced by 8.54%, and the number of parameters is reduced by 12.62%. The TOPSIS scores indicate that among the three improvements, DownS achieved the highest score, demonstrating that DownS provides the best overall performance in terms of accuracy, model complexity, and the number of parameters.

Fig. 8 illustrates the change in mAP metric over increasing epochs for both SRS-YOLO and YOLOv8. A notable enhancement in mAP is observed for the upgraded model in comparison to the original model, validating the effectiveness

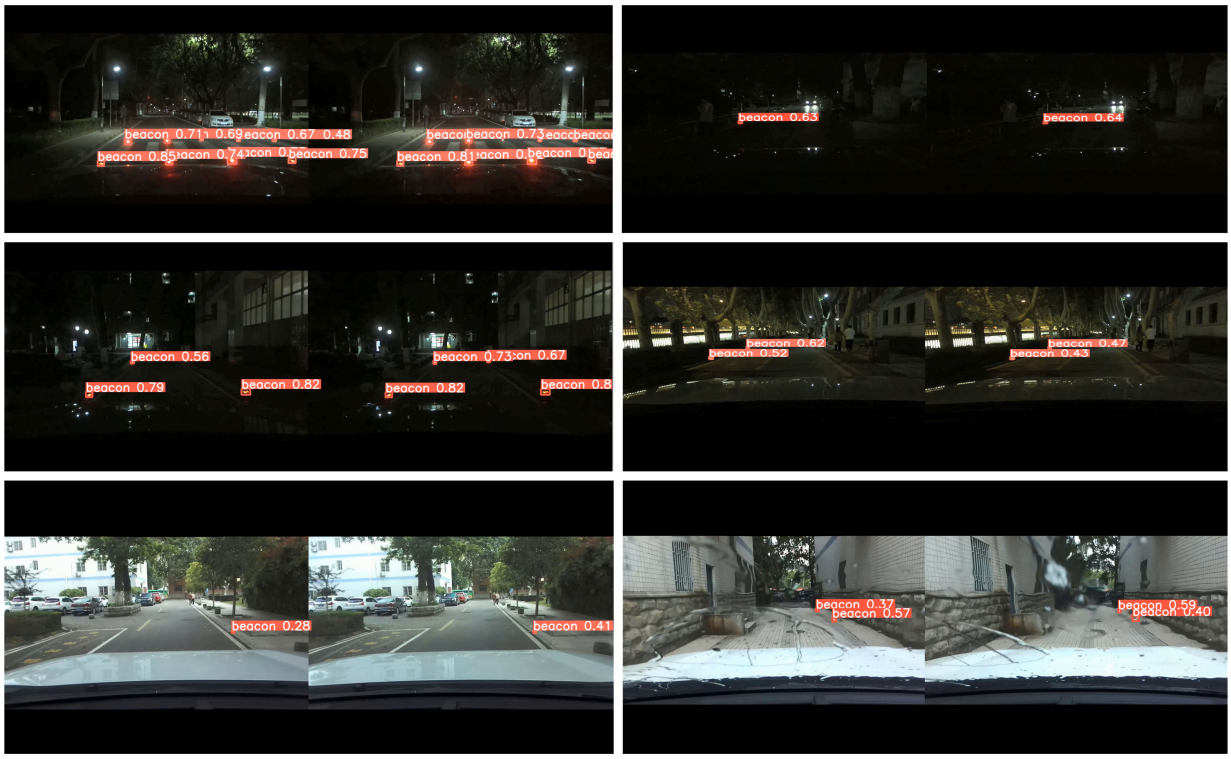


Fig. 11. The real-time detection results of SRS-YOLO in different scenarios.

of the proposed strategy. Additionally, reducing GFLOPs and the number of parameters is more beneficial for deploying the model in real vehicles.

G. Comparison With State-of-the-Art Methods

In this subsection, to validate the superiority of SRS-YOLO in smart road stud detection, a comprehensive comparison is made with the representative advanced object detection models: Faster R-CNN, SSD, EfficientDet-d1 [58], YOLOv9, and other latest lightweight small object detection models improved based on YOLOv8: DS-YOLOv8 [59] and CACS-YOLO [60]. All models are trained for 200 epochs on the smart road stud dataset, and mAP, GFLOPs, the number of parameters, and FPS are used to evaluate the performance of models in smart road stud detection.

As shown in Table VI, SRS-YOLO achieves the highest mAP, the fewest number of parameters, and the highest FPS. Although EfficientDet-d1 has a fewer GFLOPs, its mAP and FPS are significantly lower than those of SRS-YOLO, and its number of parameters is much larger than that of SRS-YOLO. This suggests that SRS-YOLO is more suitable for real-time smart road stud detection in CAVs.

TABLE VI
COMPARISON WITH STATE-OF-THE-ART METHODS

Model	mAP	GFLOPs	Parameters (Million)	FPS
Faster R-CNN	0.4651	208	41.35	8
SSD	0.6018	174.8	23.61	21
EfficientDet-d1	0.2401	**3.68**	6.55	23
YOLOv9	0.8450	266.1	60.80	19
DS-YOLOv8	0.8593	8.2	3.07	68
CACS-YOLO	0.8540	7.1	2.65	61
SRS-YOLO	**0.8792**	7.5	**2.63**	**78**

H. Robustness Analysis

To validate the robustness of the smart road stud detection model proposed in this paper, we collect images of smart road studs in different scenarios, including the placement of smart road studs on both sides of the road and on both sides of the pavement, and test SRS-YOLO and YOLOv8 on these images separately. Fig. 9 shows some randomly selected smart road stud images from different scenarios along with the detection results of different models. It is evident that in some extreme situations, such as smart road studs being too blurry due to distance or too bright due to close proximity, YOLOv8 tends to have missed detection, while SRS-YOLO effectively addresses these issues. Additionally, we create a new test dataset using data from different scenarios and various lighting conditions. We test both YOLOv8 and SRS-YOLO on this new dataset, where YOLOv8 achieved an mAP of 0.483, while SRS-YOLO achieved an mAP of 0.781. From both qualitative and quantitative perspectives, it is evident that SRS-YOLO is more robust than YOLOv8 in smart road stud detection.

I. Real-World Application

To validate the effectiveness of the SRS-YOLO in real-world scenarios, we deploy SRS-YOLO on an experimental vehicle for real-time smart road stud detection. The experimental vehicle, as shown in Fig. 10, includes a vision camera and an industrial control computer. The camera is mounted

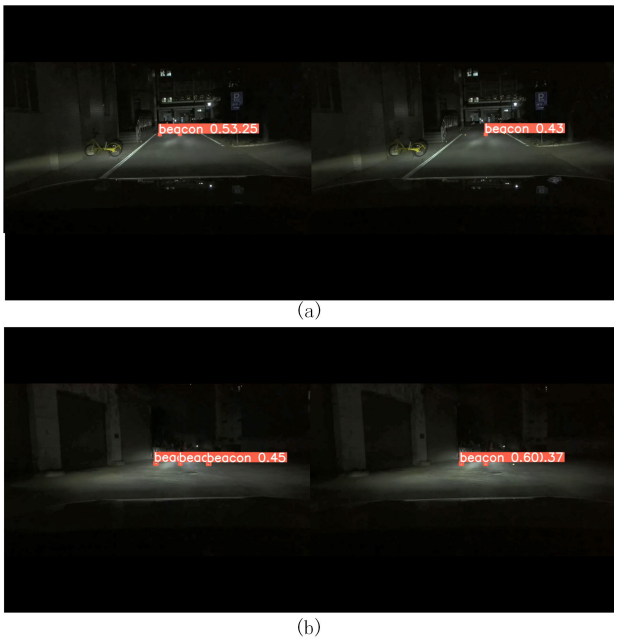


Fig. 12. The real-time detection results of SRS-YOLO for detecting passive reflective road studs. (a) Scene with interference from other lights. (b) Scene without interference from other lights.

at the front of the vehicle to capture images of smart road studs. The industrial control computer is placed in the vehicle's trunk. The image captured by the camera will be resized to 640 × 640 pixels using YOLOv8's default image resizing method before being inputted into SRS-YOLO. The industrial control computer is equipped with 64 GB RAM, an Inter(R) Core(TM) i7-13700KF @ 3.4 GHz CPU, and an NVIDIA GeForce RTX 4070 Ti GPU.

The SRS-YOLO deployed on the vehicle is used for smart road stud detection. As shown in Fig. 11, our experimental scenarios considered various light interferences in urban environments, including different lighting conditions, vehicle tail lights, headlights from oncoming lanes, as well as weather factors such as cloudy and rainy conditions. Through the driving experiment, we confirm that SRS-YOLO is capable of real-time smart road stud detection for every frame of the binocular images captured by onboard camera. This demonstrates the effectiveness of applying SRS-YOLO in real-world scenarios.

To test the detection performance of SRS-YOLO for passive reflective road studs, we conduct additional experiments. The experimental results are shown in Fig. 12. In scene (a), three passive reflective road studs appear in both the left and right camera images. SRS-YOLO successfully detects two studs in the left camera image, while only one is detected in the right camera image. This is due to the presence of other light interference, which makes the features of these road studs less distinguishable for SRS-YOLO, leading to poorer detection performance. In scene (b), without light interference, SRS-YOLO successfully detected all three road studs in the left camera image and two road studs in the right camera image, demonstrating better detection performance. The experimental results confirm the effectiveness of SRS-YOLO in detecting passive reflective road studs. However, its performance may suffer from interference from other light sources.

To explore the energy consumption of running SRS-YOLO on autonomous vehicles, we use HWiNFO [61] software to measure the power consumption of SRS-YOLO running on the experimental vehicle, which is recorded at 2.986 W and is negligible compared to the energy required to maintain the vehicle's motion [62]. This low power consumption demonstrates the suitability of SRS-YOLO for energy-constrained applications in CAVs.

V. Conclusion and Future Work

In this work, a novel smart road stud detction method was proposed based on YOLOv8, and a real-time vehicle onboard smart road stud detection system was established. First, a smart road stud dataset with 2360 images was built to train and test deep learning models. Second, a lightweight and efficient smart road stud detection model SRS-YOLO was designed. In SRS-YOLO, we introduced SE attention mechanism to enhance smart road stud detection accuracy by distinguishing the importance of different channels in feature maps. In addition, we proposed a new downsampling method, DownS, to reduce the number of parameters. DownS combines the average pooling and the max pooling to reduce information loss during the downsampling process, which is advantageous for improving detection performance. Furthermore, we trained the model using the NWD loss function, which can reduce the sensitivity to location deviation, thereby improving the detection performance for small targets. The experimental results confirmed the superior performance of SRS-YOLO. Compared with the baseline model, the mAP is increased by 10.30%, the number of parameters is reduced by 12.62%, and the GFLOPs is reduced by 8.54%. Finally, we deployed a real-time smart road stud detection system on an experimental vehicle to validate the practical application of SRS-YOLO.

Due to the limitation of the experimental condition and that the experimental vehicle is not allowed to drive legally in open roads, the proposed algorithm is tested only on limited road conditions. In the future, we expect to extensively validate the performance of the algorithm, especially on a vehicle driving at a high speed. Additionally, improving the detection performance of SRS-YOLO for passive reflective road studs will be a focus of our future research.

References

[1] A. Portera and M. Bassani, "Examining the impact of different LED road stud layouts on driving performance and gaze behaviour at nighttime," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 103, pp. 430–441, May 2024.

[2] F. Angioi, A. Portera, M. Bassani, J. de Oña, and L. L. D. Stasi, "Smart on-road technologies and road safety: A short overview," *Transp. Res. Proc.*, vol. 71, pp. 395–402, May 2023.

[3] R. Llewellyn, "The influence of active road studs on safe driving behaviour," M.S. thesis, Edinburgh Napier Univ., 2024. [Online]. Available: https://napier-repository.worktribe.com/output/4045237

[4] A. Shahar, R. Brémond, and C. Villa, "Can light emitting diode-based road studs improve vehicle control in curves at night? A driving simulator study," *Lighting Res. Technol.*, vol. 50, no. 2, pp. 266–281, Apr. 2018.

[5] N. Reed, "Driver behaviour in response to actively illuminated road studs: A simulator study," TRL, Wokingham, U.K., Tech. Rep. PPR143, 2006.

[6] A. Mole, "Leading lights," *Traffic Technol. Int. Annu. Rev.*, vol. 5, no. 12, 2002.

[7] G. Mao, Y. Hui, X. Ren, C. Li, and Y. Shao, "The Internet of Things for smart roads: A road map from present to future road infrastructure," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 6, pp. 66–76, Nov. 2022.

[8] Y. Sun et al., "Smart road stud-empowered vehicle magnetic field distribution and vehicle detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7357–7362, Mar. 2023.

[9] R. He, G. Mao, Y. Hui, and Q. Cheng, "Geomagnetic sensor based abnormal parking detection in smart roads," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2023, pp. 1060–1065.

[10] Z. Tao, W. Quan, and H. Wang, "Innovative smart road stud sensor network development for real-time traffic monitoring," *J. Adv. Transp.*, vol. 2022, pp. 1–9, May 2022.

[11] N. Vikram and S. Ashok, "Vehicle detection methods for intelligent road stud application: A review," *Int. J. Electr. Electron. Eng.*, vol. 7, no. 1, pp. 41–50, Jan. 2015.

[12] X. Wang, K. Li, and A. Chehri, "Multi-sensor fusion technology for 3D object detection in autonomous driving: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 2, pp. 1148–1165, Feb. 2024.

[13] J. Van Brummelen, M. O'Brien, D. Gruyer, and H. Najjaran, "Autonomous vehicle perception: The technology of today and tomorrow," *Transp. Res. C, Emerg. Technol.*, vol. 89, pp. 384–406, Apr. 2018.

[14] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digit. Signal Process.*, vol. 126, Jun. 2022, Art. no. 103514.

[15] Y. Zhang, M. Ye, G. Zhu, Y. Liu, P. Guo, and J. Yan, "FFCA-YOLO for small object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5611215.

[16] G. Cheng et al., "Towards large-scale small object detection: Survey and benchmarks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13467–13488, Nov. 2023.

[17] T. Zhang et al., "HOG-ShipCLSNet: A novel deep learning network with HOG feature fusion for SAR ship classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5210322.

[18] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023.

[19] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, vol. 25, no. 2, pp. 1–11.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 580–587.

[21] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[23] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf.*, Oct. 2016, pp. 21–37.

[24] S. Y. Alaba and J. E. Ball, "Deep learning-based image 3-D object detection for autonomous driving: Review," *IEEE Sensors J.*, vol. 23, no. 4, pp. 3378–3394, Feb. 2023.

[25] 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.

[26] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[27] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[28] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2018, pp. 1–6.

[29] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.

[30] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao, "YOLOv9: Learning what you want to learn using programmable gradient information," 2024, *arXiv:2402.13616*.

[31] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "YOLO-world: Real-time open-vocabulary object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 16901–16911.

[32] H. Wang, C. Liu, Y. Cai, L. Chen, and Y. Li, "YOLOv8-QSD: An improved small object detection algorithm for autonomous vehicles based on YOLOv8," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–16, 2024.

[33] M. He, L. Qin, X. Deng, and K. Liu, "MFI-YOLO: Multi-fault insulator detection based on an improved YOLOv8," *IEEE Trans. Power Del.*, vol. 39, no. 1, pp. 168–179, Oct. 2024.

[34] S. Xie, M. Zhou, C. Wang, and S. Huang, "CSPPartial-YOLO: A lightweight YOLO-based method for typical objects detection in remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 388–399, 2024.

[35] G. Brauwers and F. Frasincar, "A general survey on attention mechanisms in deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3279–3298, Apr. 2023.

[36] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.

[37] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13708–13717.

[38] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," 2018, *arXiv:1807.06521*.

[39] D. Ouyang et al., "Efficient multi-scale attention module with cross-spatial learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[40] X. Sun, X. Jia, Y. Liang, M. Wang, and X. Chi, "A defect detection method for a boiler inner wall based on an improved YOLO-v5 network and data augmentation technologies," *IEEE Access*, vol. 10, pp. 93845–93853, 2022.

[41] G. Peng, Z. Yang, S. Wang, and Y. Zhou, "AMFLW-YOLO: A lightweight network for remote sensing image detection based on attention mechanism and multiscale feature fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4600916.

[42] C. Peng, X. Li, and Y. Wang, "TD-YOLOA: An efficient YOLO network with attention mechanism for tire defect detection," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, 2023.

[43] Y. Dai, W. Liu, H. Wang, W. Xie, and K. Long, "YOLO-Former: Marrying YOLO and transformer for foreign object detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022.

[44] X. Yang, Z. Song, I. King, and Z. Xu, "A survey on deep semi-supervised learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 8934–8954, Sep. 2023.

[45] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network," in *Proc. ACM Multimedia Conf.*, Oct. 2016, pp. 512–520.

[46] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.

[47] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12993–13000.

[48] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient IOU loss for accurate bounding box regression," *Neurocomputing*, vol. 506, pp. 146–157, Sep. 2022.

[49] J. He, S. Erfani, X. Ma, J. Bailey, Y. Chi, and X. Hua, "Alpha-IoU: A family of power intersection over union losses for bounding box regression," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, Dec. 2021, pp. 1–12.

[50] Z. Tong, Y. Chen, Z. Xu, and R. Yu, "Wise-IoU: Bounding box regression loss with dynamic focusing mechanism," 2023, *arXiv:2301.1005*.

[51] X. Yue, H. Li, and L. Meng, "An ultralightweight object detection network for empty-dish recycling robots," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2023.

[52] X. Tang et al., "YOLO-ant: A lightweight detector via depthwise separable convolutional and large kernel design for antenna interference source detection," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–18, 2024.

[53] Z. Ning, H. Wang, S. Li, and Z. Xu, "YOLOv7-RDD: A lightweight efficient pavement distress detection model," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, pp. 6994–7003, Jul. 2024.

[54] L. Guan, L. Jia, Z. Xie, and C. Yin, "A lightweight framework for obstacle detection in the railway image based on fast region proposal and improved YOLO-tiny network," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–16, 2022.

[55] J. Wang, C. Xu, W. Yang, and L. Yu, "A normalized Gaussian Wasserstein distance for tiny object detection," 2021, *arXiv:2110.13389*.

[56] S. Wang, "Gaussian Wasserstein distance based ship target detection algorithm," in *Proc. IEEE 2nd Int. Conf. Electr. Eng., Big Data Algorithms (EEBDA)*, Feb. 2023, pp. 286–291.

[57] Y. Xiao et al., "A review of object detection based on deep learning," *Multimedia Tools Appl.*, vol. 79, nos. 33–34, pp. 23729–23791, Sep. 2020.

[58] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," 2019, *arXiv:1911.09070*.

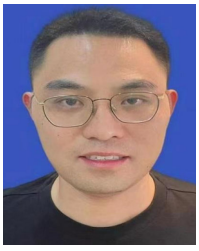
[59] L. Shen, B. Lang, and Z. Song, "DS-YOLOv8-based object detection method for remote sensing images," *IEEE Access*, vol. 11, pp. 125122–125137, 2023.

[60] Z. Cao, K. Chen, J. Chen, Z. Chen, and M. Zhang, "CACS-YOLO: A lightweight model for insulator defect detection based on improved YOLOv8m," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–10, 2024.

[61] P. Maxwell, D. Niblick, and D. C. Ruiz, "Using side channel information and artificial intelligence for malware detection," in *Proc. IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA)*, Jun. 2021, pp. 408–413.

[62] J. Huang, G. Song, F. He, and Z. Tan, "Energetic impacts of autonomous vehicles in real-world traffic conditions from nine open-source datasets," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 9, pp. 9901–9914, Sep. 2023.

Guoqiang Mao (Fellow, IEEE) is a Chair Professor and the Director of the Center for Smart Driving and Intelligent Transportation Systems, Southeast University. From 2014 to 2019, he was a Leading Professor, the Founding Director of the Research Institute of Smart Transportation, and the Vice-Director of the ISN State Key Laboratory, Xidian University. Before that, he was with the University of Technology Sydney and The University of Sydney. He has published 300 papers in international conferences and journals that have been cited more than 15,000 times. His H-index is 57 and was in the list of Top 2% most-cited scientists worldwide by Stanford University in 2022, 2023, and 2024 both by Single Year and by Career Impact. His research interests include intelligent transport systems, the Internet of Things, wireless localization techniques, mobile communication systems, and applied graph theory and its applications in telecommunications. He is a fellow of AAIA and IET. He received the Top Editor Award for outstanding contributions to IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY in 2011, 2014, and 2015. He has been serving as the Vice-Director of Smart Transportation Information Engineering Society, Chinese Institute of Electronics, since 2022. He was the Co-Chair of the IEEE ITS Technical Committee on Communication Networks from 2014 to 2017. He has served as the chair, the co-chair, and a TPC member for several international conferences. He is an Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS since 2018, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2014 to 2019, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2010 to 2020.



Keyin Wang (Graduate Student Member, IEEE) received the master's degree in control engineering from Hubei University of Automotive Technology, Shiyan, China, in 2022. He is currently pursuing the Ph.D. degree with the College of Communication Engineering, Xidian University, Xi'an, China. His research interests include vehicle localization based on multisource information fusion.

Haoyuan Du received the bachelor's degree in communication engineering from Dongguan University of Technology, Dongguan, China, in 2021. He is currently pursuing the master's degree with Guangzhou Institute of Technology, Xidian University, Guangzhou, China. His research interests include new-generation electronic information technology.

Baoqi Huang (Senior Member, IEEE) received the B.E. degree in computer science from Inner Mongolia University (IMU), Hohhot, China, in 2002, the M.S. degree in computer science from Peking University, Beijing, China, in 2005, and the Ph.D. degree in information engineering from The Australian National University, Canberra, ACT, Australia, in 2012. He is with the College of Computer Science, IMU, where he is currently a Professor. His research interests include indoor localization and navigation, wireless sensor networks, and mobile computing. He was a recipient of the Chinese Government Award for Outstanding Chinese Students Abroad in 2011.

Xiaojiang Ren (Member, IEEE) received the Ph.D. degree in computer science from The Australian National University in 2016. He is an Associate Professor with Xidian University, Xi'an, China. His research interests include intelligent transport systems, the Internet of Things, wireless sensor networks, routing protocol design for wireless networks, and optimization problems.

Tianxuan Fu received the master's degree in transport information engineering and control from Lanzhou Jiaotong University, Lanzhou, China, in 2019. He is currently pursuing the Ph.D. degree with the College of Communication Engineering, Xidian University, Xi'an, China. His research interests include target tracking, information fusion, and intelligent transportation systems.

Zhaozhong Zhang received the B.Eng. degree from the University of Nottingham, U.K., in 2013, and the M.Sc. and Ph.D. degrees in automotive engineering and transport systems from Cranfield University, in 2015 and 2020, respectively. He is currently a Lecturer with the School of Information Engineering, Nanchang Hangkong University. His research interests include vehicle localization and target tracking.