An Improved YOLOv8 Based Smart Road Stud Detection Method

Guoqiang Mao[†], Keyin Wang[‡*], Haoyuan Du[‡], Xiaojiang Ren[‡],

[†]School of Transportation, Southeast University, Nanjing 210096, China
[‡]School of Telecommunications Engineering, Xidian University, Xi'an 710071, China
[*]Email: keyinwang@stu.xidian.edu.cn

2025 IEEE Wireless Communications and Networking Conference (WCNC) | 979-8-3503-6836-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/WCNC61545.2025.10978697

***Abstract*—Smart road studs are widely used for road safety and traffic data collection. Their accurate and reliable detection, and integration into the perception and control modules of connected and autonomous vehicles (CAVs), enhances road boundary detection, vehicle localization, and driving safety. However, real-time, accurate and reliable detection of the small-sized smart road studs is challenging for fast moving CAVs, especially in harsh environments. To address the 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). We then introduce a novel downsampling module (DownS) combining the average pooling and the max pooling to reduce the number of parameters and minimize information loss during downsampling. Furthermore, we replace the loss function with Normalized Wasserstein Distance (NWD) loss to reduce sensitivity to location deviations in small target detection. Finally, we deploy a real-time smart road stud detection system on an experimental vehicle to validate the feasibility and effectiveness of the proposed algorithm. The experimental results demonstrate that the proposed algorithm significantly enhances the accuracy and efficiency of smart road stud detection, increasing the mean average precision by 9.58% and reducing the number of parameters by 13.71%. Our dataset is available at: https://github.com/wky-xidian/smart-road-stud-dataset.**

***Index Terms*—YOLOv8, smart road stud, real-time detection system, connected and autonomous vehicle.**

I. Introduction

ROAD studs have been extensively used for over 80 years in various countries for multiple applications [1], [2]. As early as 1930s, the UK implemented road studs to mark boundaries, lane directions, and intersections, improving driving safety during nighttime and adverse weather conditions [3]. In the Netherlands, road studs are a crucial part of road infrastructure, used for lane delineation, edge marking, and intersection marking [4]. In North America, specifically the US and Canada, road studs mark lane boundaries and directions on highways, to help drivers stay in the correct lane during nighttime and low-visibility conditions [5].

With advancements in electronics, communication, sensing, and solar technology, smart road studs integrating light emitting diodes (LEDs) and various sensors (e.g., temperature, humidity, light, vibration, and magnetic sensors) have become feasible and are increasingly applied in intelligent transportation systems [6]. LEDs embedded in road studs significantly enhance visibility in low-light conditions like fog, rain, or darkness [7]. As reliable, widely deployed sensing devices, smart road studs facilitate vehicle detection, wireless data transmission, and processing, which support digital twin systems [8]. They can also detect traffic accidents and interact with drivers by changing light colors to indicate 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. First, accurate and reliable detection of smart road studs through CAV's onboard cameras can help CAVs identify road and lane boundaries, which is especially critical in harsh environments and adverse weather conditions [9]. Second, the smart road 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 [6].

Detecting smart road studs for CAVs is challenging due to their small size, inconsistent brightness, and often blurry backgrounds. These factors make traditional object detection methods, which rely on handcrafted features, less effective in providing accurate and reliable detection. Since AlexNet's introduction in 2012, deep neural network (DNN)-based detection algorithms have taken the lead in object detection [10], [11], [12]. DNNs excel in automatically learning and extracting high-level features from images, eliminating the need for complex manual feature extraction, and are adept at handling diverse and complex visual scenarios [13]. Given these advantages, we are exploring the use of a DNN-based object detection algorithm for smart road stud detection.

Modern DNN-based object detection algorithms are typically categorized into two types: 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 achieve bounding box detection and class probability estimation simultaneously in a single step [14]. Typical two-stage detectors include region-based convolutional neural network (R-CNN) [15], Fast R-CNN [16], and Faster R-CNN [17], etc., while one-stage detectors include single shot multibox detector (SSD) series [12] and you only look once (YOLO) series [11]. It is noteworthy that YOLO algorithms have gained significant popularity in the development of object detection methods because of their impressive accuracy and

speed. 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 [11]. Based on the YOLOv1 framework, a series of versions have been proposed [18], [19], [20]. YOLO algorithms are suitable for detecting general objects. There is still significant room for improvement in detection accuracy, lightweighting, robustness, and model complexity when it comes to detecting small objects like smart road studs.

In this paper, we introduce a novel smart road stud detection method based on YOLOv8 algorithm. We propose a new downsampling structure, DownS, which integrates average pooling and max pooling to mitigate the loss of smart road stud-related features due to downsampling. More importantly, it significantly reduces the number of model parameters compared to the original downsampling method. 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. Due to the absence of 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 models. Finally, we deploy the trained smart road stud detection model on an experimental vehicle to validate the effectiveness of the proposed algorithm. The following is a list of the main contributions of this paper:

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

2) The NWD loss function is introduced, which measures the similarity between predicted boxes and ground truth boxes using the Wasserstein distance, improves the model's robustness to detecting smart road studs.

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 smart road stud detection method.

The remaining sections of this paper are organized as follows. Section II provides details on the network structure of the proposed model. Section III introduces the experimental details, including dataset preparation, model evaluation, and comparative experiments. Finally, conclusions and future work are drawn in Section IV.

II. The Proposed Model

A. Overall Framework

Even though YOLOv9 [21] and YOLO-World [22] have been released recently, considering the requirement for model stability in real-world applications, we chose the more mature YOLOv8 as our benchmark framework. Although YOLOv8 has demonstrated outstanding performance in a variety of object detection tasks, it faces challenges when detecting smart road studs due to blurry backgrounds, varying brightness, and small target areas. Therefore, it is necessary to improve YOLOv8's capabilities for smart road stud detection. Fig. 1 illustrates the framework of the proposed model. 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 the proposed model comprises Conv, coarse-to-fine-1 (C2F-1), 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 C2F-1 module consists of Conv, Split, Concat and Darknet Bottleneck-1 (DB-1) modules. For the DB-1 module, there are two Conv modules. Moreover, the DownS module is a new downsampling module proposed in this paper. The following section will provide a detailed introduction to the DownS module. Given the input smart road stud image $\boldsymbol{I} \in \mathbb{R}^{H \times W \times C}$, where H, W, and C are the height, 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.

2) Neck: The neck of the proposed model consists of multiple Upsample, Concat, DownS, and C2F-2 modules. The C2F-2 module consists of Conv, Split, Concat and DB-2 modules. The difference between DB-2 and DB-1 lies in the connection method. In the neck module, the learned smart road stud-related features are enhanced through cross-stage feature fusion.

3) Head: The head of the proposed model 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.

In the basis of YOLOv8, the proposed model introduces DownS modules to reduce information loss during downsampling process while also reducing the number of model parameters. Additionally, the NWD loss function is used during training process to enhance the model's robustness in detecting smart road studs.

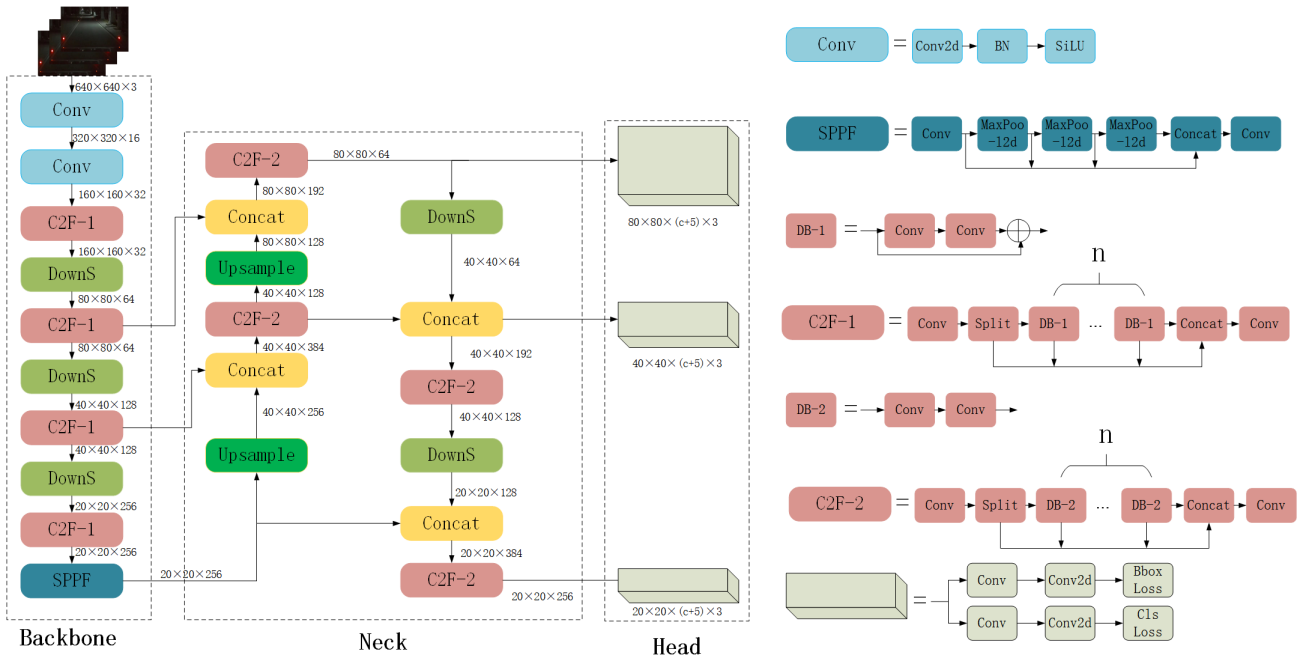

Fig. 1. An illustration of the framework of the proposed model.

B. DownS Module

In YOLOv8, both the backbone and neck modules use convolutional layers for downsampling, which significantly increases the number of parameters. Additionally, this approach can cause information loss in the feature maps by reducing their resolution and making features related to smart road studs coarser, thereby impacting the accuracy of smart road stud detection. Inspired by the downsampling method in YOLOv9 [21], we design a new downsampling approach, i.e., DownS, to solve these challenges, as illustrated in Fig. 2.

In DownS, the input feature map 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. Finally, the two parts are concatenated to obtain the downsampled feature map. DownS reduces the number of channels processed by the Conv module using a Split operation, which helps decrease the number of parameters. Additionally, it combines average pooling and max pooling to minimize information loss during the downsampling process.

To quantitatively assess the performance of the DownS module in reducing model parameters, we use the fourth layer of the YOLOv8 model's backbone as an example. We calculate and compare the number of parameters for both the convolutional downsampling method and the DownS downsampling method. 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 number of parameters is $(1 \times 1 \times 16 + 1) \times 32 + 32 + (3 \times 3 \times 16 + 1) \times 32 + 32 = 524$.

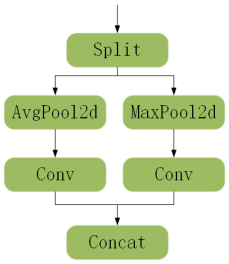

Fig. 2. Architecture of the DownS.

C. Loss Function

The intersection over union (IoU)-based metrics are highly sensitive to variations in small objects, making them unsuitable for the detection of smart road studs. In order to improve the performance of smart road stud detection, we replace the default Complete IoU loss function in YOLOv8 with the NWD loss function [23], a metric specifically designed for small objects.

Smart road studs do not fit standard rectangular shapes and often include background pixels within their bounding boxes, with foreground pixels concentrated in the center and background pixels along the edges. To more accurately represent 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:

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

where (μ_x, μ_y) is the center of the inscribed ellipse, σ_x and σ_y are the lengths of semi-axes along x and y axes, 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:

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

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 $\left(\mathbf{x}-\boldsymbol{\mu}\right)^T\sum^{-1}\left(\mathbf{x}-\boldsymbol{\mu}\right)=1$, the ellipse represented by (1) 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 $N\left(\boldsymbol{\mu},\sum\right)$ with

$$\mu=\begin{bmatrix} c_x \\ c_y \end{bmatrix}, \sum=\begin{bmatrix} \frac{w^2}{4} & 0 \\ 0 & \frac{h^2}{4} \end{bmatrix} \tag{3}$$

The similarity between the ground truth bounding box $\left(c_{x_g}, c_{y_g}, w_g, h_g\right)$ and the predicted bounding box $\left(c_{x_p}, c_{y_p}, w_p, h_p\right)$ can quantified by the distance between two Gaussian distributions, which is calculated using the 2^{nd} Wasserstein distance [24] as follows:

$$D_2^2\left(N_p, N_g\right)=\| \left(\left[c_{x_p}, c_{y_p}, w_p, h_p\right]^T, \left[c_{x_g}, c_{y_g}, w_g, h_g\right]^T\right)\|_2^2 \tag{4}$$

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:

$$NWD\left(N_p, N_g\right)=\exp\left(-\frac{\sqrt{D_2^2\left(N_p, N_g\right)}}{C}\right) \tag{5}$$

where C is a constant, selected based on empirical experience.

The NWD metric is chosen as the loss function:

$$L_{NWD}=1-NWD\left(N_p, N_g\right) \tag{6}$$

III. Experiments and Analysis

A. Dataset Establishment

Currently, there are no available datasets for training and testing deep learning models for smart road stud detection. To address this gap, a dedicated smart road stud dataset is developed. 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. 3. 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.

Fig. 3. Samples of the smart road stud dataset.

B. Workstation Configuration and Hyperparameters Setting

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

TABLE I
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 the proposed model, the metrics selected include precision (P), recall (R), mean average precision (mAP), frames/s (FPS), the number of parameters, and giga floating-point operations per second (GFLOPs) which is used to measure the complexity of the model [25]:

$$\begin{cases} P=\frac{TP}{TP+FP} \\ R=\frac{TP}{TP+FN} \\ mAP=\frac{\sum_{n=1}^{N}AP(n)}{N} \end{cases} \tag{7}$$

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, $N = 1$.

D. Analysis of Ablation Experiments

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, and the number of parameters.

From Table II, it can be seen that both of the proposed improvement methods in this paper enhance the mAP compared to the original model. 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, which measures the similarity between bounding boxes using Wasserstein distance, is less sensitive to target scale, thus improving detection accuracy for smart road studs. By simultaneously employing DownS and NWD, the model achieves superior detection performance, mAP is increased by 9.58%, GFLOPs is reduced by 8.54%, and the number of parameters is reduced by 13.71%.

TABLE II
PERFORMANCE COMPARISON OF THE MODELS WITH DIFFERENT IMPROVEMENT STRATEGIES

DownS	NWD	mAP	GFLOPs	Parameters
		0.7971	8.2	3011034
√		0.8627	7.5	2598371
	√	0.8686	8.2	3011034
√	√	0.8735	7.5	2598371

E. Comparison with State-of-the-Art Methods

In this subsection, to validate the superiority of the proposed model in smart road stud detection, a comprehensive comparison is made with the representative advanced two-stage model: Faster R-CNN, one-stage model: SSD, and the latest version of YOLO: YOLOv9. Table III presents the comparison results. It is easy to note that the detection precision of the Faster R-CNN model is higher than one-stage models, this is because Faster R-CNN consists of two stages: region proposal and bounding boxes generation. In the first stage, the model extracts candidate regions that may contain objects. In the second stage, these candidate regions are classified and regressed to precisely locate and identify the targets. This staged approach helps improve the detection precision of two-stage models. Compared to the proposed model, Faster R-CNN, SSD, and YOLOv9 have larger GFLOPs, higher number of parameters, and lower FPS, which leads to models being bulky and slow running. These characteristics make it challenging to apply Faster R-CNN, SSD, and YOLOv9 in CAVs that demand real-time and lightweight models. In contrast, the proposed model performs significantly better in these aspects.

Fig. 4. The experimental vehicle with the visual camera and industrial control computer.

F. Real-world application

To validate the effectiveness of the proposed model in real-world scenarios, we deploy the proposed model on an experimental vehicle for real-time smart road stud detection. The experimental vehicle, as shown in Fig. 4, includes a vision camera (Stereolabs, ZED 2i) and an industrial control computer. The camera is mounted 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 frame rate of the camera is 30 FPS, and it has 1280 × 720 image resolution with a field of view as 90° × 60°. The image captured by the camera will be resized to 640 × 640 pixels before being inputted into the proposed model. 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 proposed model deployed on the vehicle is used for smart road stud detection. Through the driving experiment, we confirm that the proposed model is capable of real-time smart road stud detection for every frame of the binocular images captured by onboard camera. The detection results are shown in Fig. 5. This demonstrates the effectiveness of applying the proposed model in real-world scenarios.

IV. CONCLUSION AND FUTURE WORK

In this work, a novel smart road stud detection 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 was designed. We proposed a new downsampling module, 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

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

Model	P	R	mAP	GFLOPs	Parameters	FPS
Faster R-CNN	0.8971	0.4142	0.4651	208	41348000	8
SSD	0.7585	0.6950	0.6018	174.8	23612000	21
YOLOv9	0.8673	0.7614	0.8450	266.1	60797222	19
Proposed Method	0.8543	0.7914	0.8735	7.5	2598371	79

Fig. 5. The real-time detection results of the proposed model in different scenarios.

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 the proposed model. Compared with the baseline model, the mAP is increased by 9.58%, the number of parameters is reduced by 13.71%, 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 the proposed model.

Due to experimental constraints and the legal restrictions preventing the experimental vehicle from driving on open roads, the proposed model has only been tested under limited road conditions. In the future, we plan to extensively validate the algorithm's performance, particularly on a vehicle traveling at high speeds.

ACKNOWLEDGMENT

This research is supported by NSFC grant, Grant number: U21A20446.

REFERENCES

[1] A. Portera and M. Bassani. Examining the impact of different led road stud layouts on driving performance and gaze behaviour at night-time. *Transportation Research Part F: Psychology and Behaviour*, 103:430–441, 2024.

[2] F. Angioi, A. Portera, M. Bassani, J. Ona, and L. Di Stasi. Smart on-road technologies and road safety: A short overview. In *Proceedings of the Conference on Transport Engineering, CIT 2023*, volume 71, pages 395–402, 2023.

[3] C. Crawford and V. Doran. Be my light be my guide: Intelligent road studs - the way forward. *Traffic technology international*, (1), 2005.

[4] A. Shahar, R. Bremond, and C. Villa. Can light emitting diode-based road studs improve vehicle control in curves at night? a driving simulator study. *Lighting Research and Technology*, 50(2):266–281, APR 2018.

[5] N. Reed. Driver behaviour in response to actively illuminated road studs: A simulator study. *Published Project Report Ppr*, 2006.

[6] 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 Intelligent Transportation Systems Magazine*, 14(6):66–76, 2022.

[7] Y. Sun, H. Wang, W. Quan, X. Ma, Z. Tao, M. Elhajj, and W. Ochieng. Smart road stud-empowered vehicle magnetic field distribution and vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, 24(7):7357–7362, 2023.

[8] R. He, G. Mao, Y. Hui, and Q. Cheng. Geomagnetic sensor based abnormal parking detection in smart roads. In *Proceedings of the IEEE Global Communications Conference. (GLOBECOM)*, pages 1060–1065, 2023.

[9] Z. Tao, W. Quan, and H. Wang. Innovative smart road stud sensor network development for real-time traffic monitoring. *Journal of Advanced Transportation*, 2022, 2022.

[10] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25(2), 2012.

[11] J. Redmon, S. Divvala, R. Girshick, and A Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg. Ssd: Single shot multibox detector. *Springer, Cham*, 2016.

[13] Z. Zhao, P. Zheng, S. Xu, and X. Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 2019.

[14] S. Alaba and J. Ball. Deep learning-based image 3-d object detection for autonomous driving: Review. *IEEE Sensors Journal*, 23(4):3378–3394, 2023.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (CVPR)*, pages 580–587, 2014.

[16] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[18] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (CVPR)*, pages 6517–6525, 2017.

[19] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv e-prints*, 2018.

[20] C. Wang, A. Bochkovskiy, and H. Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7464–7475, 2023.

[21] C. Wang, I. Yeh, and H. Liao. Yolov9: Learning what you want to learn using programmable gradient information, 2024.

[22] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan. Yolo-world: Real-time open-vocabulary object detection, 2024.

[23] J. Wang, C. Xu, W. Yang, and L. Yu. A normalized gaussian wasserstein distance for tiny object detection, 2022.

[24] S. Wang. Gaussian wasserstein distance based ship target detection algorithm. In *Proceedings of the IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, pages 286–291, 2023.

[25] Y. Xiao, Z. Tian, J. Yu, Y. Zhang, S. Liu, S. Du, and X. Lan. A review of object detection based on deep learning. *Multimedia Tools and Applications*, 79(33-34):23729–23791, SEP 2020.