Mechatronics and Intelligent Transportation Systems
https://www.acadlore.com/journals/MITS

# A Lightweight Real-Time Vision Framework for Road Infrastructure Monitoring in Intelligent Transportation Systems

Quanliang Chen[1], Lin Zhang[2*], Jialin Ma[1], Ashim Khadka[3]

[1] Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, 223003 Huaian, China
[2] Faculty of Management Engineering, Huaiyin Institute of Technology, 223003 Huaian, China
[3] Nepal College of Information Technology, Pokhara University, 44700 Lalitpur, Nepal

[*] Correspondence: Lin Zhang (zlmjl@hyit.edu.cn)

**Received:** 11-15-2025 **Revised:** 01-10-2026 **Accepted:** 01-22-2026

**Citation:** Q. L. Chen, L. Zhang, J. L. Ma, and A. Khadka, "A lightweight real-time vision framework for road infrastructure monitoring in intelligent transportation systems," *Mechatron. Intell Transp. Syst.*, vol. 5, no. 1, pp. 1–10, 2026. https://doi.org/10.56578/mits050101.

**Abstract:** Reliable and timely perception of road surface conditions is a fundamental requirement in intelligent transportation systems (ITS), as it directly affects traffic safety, infrastructure maintenance, and the operation of connected and autonomous vehicles. Vision-based pothole detection has emerged as a practical solution due to its low sensing cost and deployment flexibility; however, existing deep learning approaches often struggle to achieve a satisfactory balance between detection accuracy, robustness to scale variations, and real-time performance on resource-constrained platforms. This study presents Partial Group-You Only Look Once (PG-YOLO), a lightweight real-time vision framework designed for road infrastructure monitoring in ITS. Built upon a compact one-stage detector, the proposed framework introduces a Partial Multi-Scale Feature Aggregation (PMFA) module to enhance the representation of small and irregular potholes under complex road conditions, as well as a Grouped Semantic Enhancement Attention (GSEA) module to improve high-level semantic discrimination with limited computational overhead. The framework is specifically designed to meet the low-latency and low-complexity requirements of vehicle-mounted and roadside sensing devices. Experimental evaluations conducted on a mixed road damage dataset demonstrate that the proposed approach achieves consistent improvements in detection accuracy while reducing model parameters and maintaining real-time inference speed. Compared with the baseline model, PG-YOLO improves precision, recall, and detection stability under challenging illumination and scale variations, while remaining suitable for edge deployment. These results indicate that the proposed framework can serve as an effective perception component for ITS, supporting continuous road condition awareness and data-driven maintenance and safety management.

**Keywords:** Intelligent transportation systems; Road infrastructure monitoring; Pothole detection; Real-time vision perception; Lightweight deep learning

## 1 Introduction

Road potholes are a pervasive form of pavement distress that degrades ride comfort, accelerates vehicle wear, and may trigger loss-of-control events, particularly for two-wheelers and at high speed. For highway agencies and city operators, rapid pothole discovery is also a prerequisite for preventive maintenance, cost-effective rehabilitation, and data-driven prioritization of repair actions [1, 2]. In intelligent transportation systems (ITS), pothole perception is therefore not merely an inspection task; it is an enabling capability for safety warning, infrastructure health monitoring, and reliable autonomous driving.

A vision-based pothole detector can be naturally integrated into several ITS and mechatronic mobility architectures. First, it can run on vehicle-mounted cameras with an on-board computer to provide real-time driver warnings or to support motion planning modules in autonomous vehicles. Second, it can be deployed on roadside monitoring units or smart city sensing nodes to continuously assess pavement condition on key road segments. Third, it can be integrated into infrastructure inspection platforms, such as maintenance vehicles or inspection robots, to automate routine road surface surveys.

With the rapid advancement of deep learning, convolutional neural network (CNN)-based detectors have become a mainstream solution for road damage perception. Representative frameworks include one-stage models (e.g., the You Only Look Once (YOLO) family [3]) and two-stage models (e.g., Faster Regions with Convolutional Neural Networks (Faster R-CNN) [4]), as well as Single Shot MultiBox Detector (SSD) [5] and transformer-based detectors such as DEtection TRansformer (DETR) variants [6]. Early work mainly used CNNs for classification or coarse recognition. For example, Mittal et al. demonstrated the effectiveness of Visual Geometry Group 16 (VGG16) for pothole recognition [7], and Ye et al. [8] improved robustness to noise and scale variations by introducing a pre-pooling strategy. Subsequent studies investigated two-stage detection pipelines, for example, Yebes et al. [9] introduced a large-scale pothole dataset and performed comparisons using Faster R-CNN, while Lan et al. [10] combined detection with pixel-level statistics to quantify pothole areas. Although accurate, two-stage methods often incur high computational overhead, which makes them less suitable for real-time ITS deployments.

With the development of one-stage detection algorithms, research attention has gradually shifted toward lightweight models that balance detection accuracy and inference speed. Tithi et al. [11] implemented pothole detection and vehicle speed control based on SSD and MobileNet architectures. Ahmed [12] proposed VGG16 with dilated convolutions as Faster R-CNN backbone, reducing computation and improving pothole detection accuracy and speed. Building on this progress, Maeda et al. [13] released a public dataset covering multiple categories of road surface distress, which has facilitated standardized evaluation in related research. Some studies [14, 15] further improved the detection performance of small-scale potholes under complex illumination and occlusion conditions by incorporating multi-scale feature fusion and attention mechanisms. In recent years, models such as YOLOv4, YOLOv7, and YOLOv8 [16–18] have been successively introduced into pothole detection tasks, achieving notable improvements in both real-time performance and detection accuracy. Improving Detection Stability of Small-Scale Potholes in Complex Backgrounds through Multi-Scale Feature Modeling and Contextual Information Enhancement Strategies [19]. In addition, certain works enhance structural modeling of road defects using depth or geometric cues, whereas others improve deployment efficiency on embedded and edge devices through efficient feature representation and network architecture optimization [20, 21].

Despite these successes, some challenges remain unresolved. Existing methods struggle to effectively detect potholes with highly variable morphologies and scale differences, and detection models are often computationally expensive, limiting their practical applicability. Building upon previous research, this study proposes a novel pothole detection model, Partial Group-YOLO (PG-YOLO), by integrating the Partial Multi-Scale Feature Aggregation (PMFA) and Grouped Semantic Enhancement Attention (GSEA) modules. The contributions of this study are summarized as follows:

(1) A PMFA module is designed to enhance the extraction of fine-grained details and small-object features through multi-scale convolutions and partial channel fusion, thereby effectively improving small-object detection performance in complex environments.

(2) A GSEA module is proposed, which employs parallel modeling of grouped convolutions and global attention to enhance high-level semantic representation and salient-region awareness, thereby further improving detection robustness under complex environments and interference conditions.

## 2 Proposed Method

The development of the PG-YOLO model is aimed at addressing two key challenges in road pothole detection:

(a) Road potholes exhibit arbitrary shapes and complex geometric structures, making them considerably more difficult to detect than objects with relatively fixed shapes, such as vehicles, pedestrians, and cyclists. Moreover, potholes generally occupy a small proportion of image pixels, resulting in limited extractable appearance information and a high risk of feature loss during forward propagation in deep networks, which in turn constrains detection accuracy.

(b) ITS deployments typically require low latency and low power consumption on embedded or edge hardware, which constrains model size and computational complexity.

To address these challenges, we enhance the YOLOv11n model and propose the PG-YOLO framework. By replacing the C3k2 modules in the YOLOv11n backbone with the PMFA module, the model captures multi-scale receptive fields from local to global regions, improving edge detail representation and detection accuracy. Additionally, the GSEA module is incorporated to disperse fused high-level semantic features across different detection layers, shortening the feature propagation path, reducing information loss, and increasing detection speed.

The architecture of PG-YOLO is illustrated in Figure 1.

### 2.1 Partial Multi-Scale Feature Aggregation Module

Road potholes typically present large scale variations, irregular shapes, and intricate edge details, which can lead existing detection networks to lose fine-grained features or fail to adequately represent small and irregular targets. To enhance the network's perception and representation of such targets, this study proposes a PMFA module.

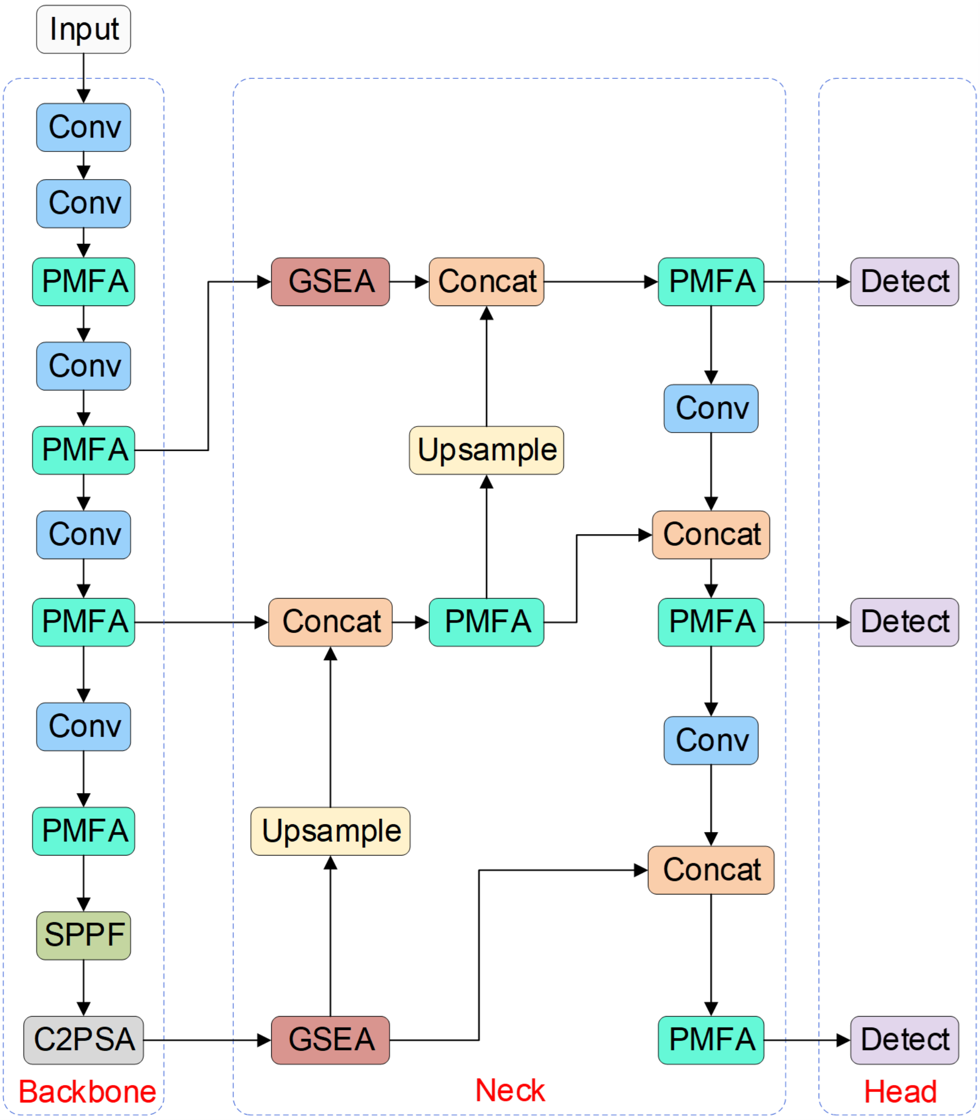

**Figure 1.** Partial Group-You Only Look Once (PG-YOLO) network architecture

The PMFA incorporates residual connections, as introduced in ResNet [22], effectively mitigating the vanishing gradient problem. It also adopts the 1 × 1 convolution strategy from FasterNet [23] to reduce channel dimensionality, thereby decreasing model parameters and computational cost. Consequently, the module integrates multi-scale feature extraction, edge information enhancement, and convolutional feature fusion to improve the model's representational capacity.

In the PMFA module, the input feature map is processed through stacked multi-scale convolutions (3 × 3, 5 × 5, 7 × 7), enabling progressive extraction and fusion of features across different receptive fields. This captures both local and global information, allowing comprehensive modeling from fine edge details to larger contextual regions. Potholes often vary in size and shape, and the PMFA module's multi-scale pathways enable the network to capture spatial features across scales, improving small-object detection. Moreover, its partial channel convolution reduces redundant computation, preserves feature density, and enhances both efficiency and feature attention.

The PMFA module also employs 1 × 1 convolution and residual connections to fuse multi-scale features while retaining the original input, strengthening feature correlations and preserving maximal information. This feature reconstruction mechanism enables the model to maintain global structural consistency while more effectively extracting key pothole edge information. As a result, the backbone network can fully capture features of targets at different scales, further enhancing the overall feature extraction capability. The structure is shown in Figure 2, and the implementation process of the PMFA module is as follows:

$$Y = X + \mathrm{Conv}_{1\times1}\left(\mathrm{Concat}\left(F_{3\times3}, F_{5\times5}, F_{7\times7}\right)\right) \tag{1}$$

where, $F_{(k\times k)}$ represents the intermediate feature maps produced by $k \times k$ convolutions on partial channels, $X$

represents the input features, $Y$ represents the module output, $Concat$ represents channel-wise concatenation.

In summary, PMFA expands the effective receptive field across multiple scales while preserving fine edge details through partial computation and residual fusion, thereby improving small and irregular pothole detection with limited overhead.

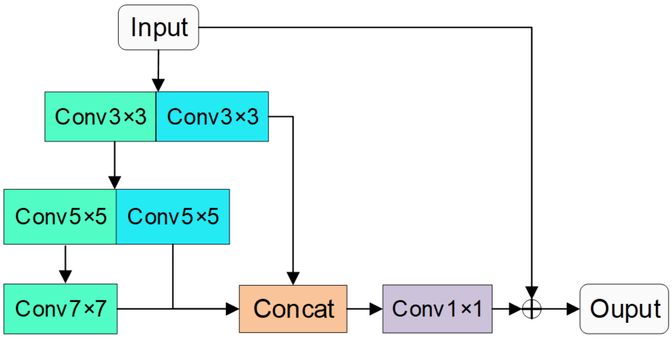

**Figure 2.** Partial Multi-Scale Feature Aggregation (PMFA) module

### 2.2 Grouped Semantic Enhancement Attention Module

To further enhance semantic representation and salient-region awareness, we propose the GSEA module, which is designed to be computationally efficient for edge deployment. As shown in Figure 3, GSEA adopts a dual-branch architecture that combines local grouped interactions and global channel attention, and fuses them via residual learning.

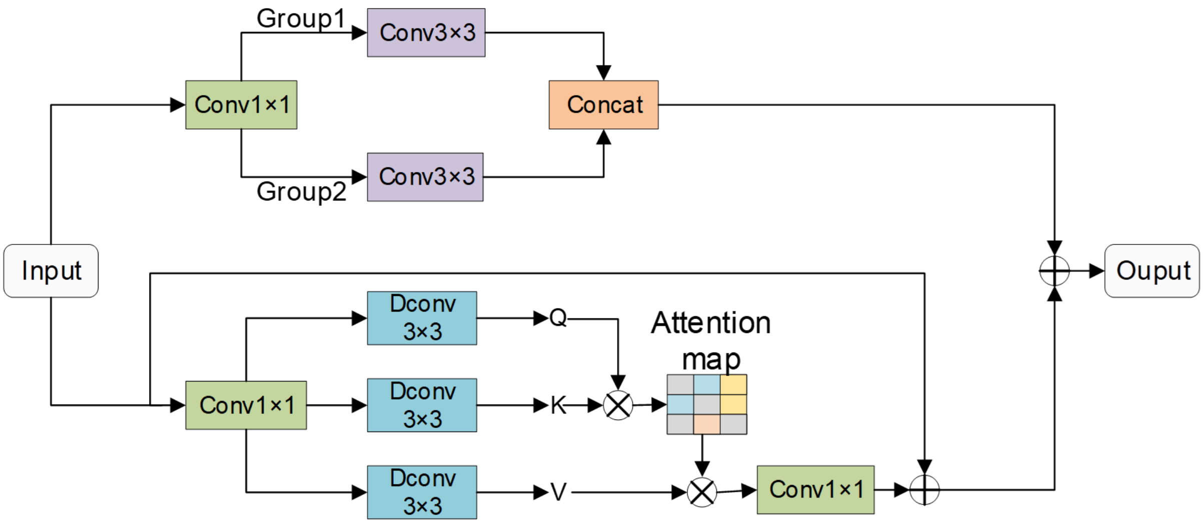

**Figure 3.** Grouped Semantic Enhancement Attention (GSEA) module

In the local branch, the input feature map is first processed by a $1 \times 1$ convolution to adjust channel dimensions and reduce redundancy, enhancing the efficiency of subsequent operations. It then passes through two stages of $3 \times 3$ grouped convolutions, where each group models local patterns, retaining fine details while lowering computational overhead.

$$F_{conv} = \text{Concat}\left[W^1_{3\times3}\left(Y_1^{\text{Group 1}}\right), W^2_{3\times3}\left(Y_2^{\text{Group 2}}\right)\right] \qquad (2)$$

where, Group 1 has $1 - \hat{C}/2$ channels, and Group 2 has $\hat{C}/2 + 1 - \hat{C}$ channels, $Y$ represents the module output. In the global branch, after compressing the channels using a $1 \times 1$ convolution, $3 \times 3$ depthwise separable convolutions are employed to generate the query $(Q)$, key $(K)$, and value $(V)$. For channel attention computation, $(Q)$ is reshaped into $\left(\hat{Q} \in \mathbb{R}^{HW\times\hat{C}}\right)$ and $(K)$ is transposed into $(\widehat{K} \in \mathbb{R}^{HW\times\hat{C}})$, followed by feature weighting to produce the output.

$$Q, K, V = \left(\text{Split}\left(W^{dw}_{3\times3}\left(W_{1\times1}(Y)\right)\right)\right) \qquad (3)$$

$$A = \mathrm{Softmax}(\hat{K}\hat{Q}/\alpha) \in \mathbb{R}^{\hat{C}\times\hat{C}} \tag{4}$$

$$F_{att} = W_{1\times1}(V \times A) + Y \tag{5}$$

where, $Split$ denotes the channel-wise split operation; $A$ represents the computation of the attention map; $\alpha$ represents a learnable scaling factor; Softmax corresponds to normalization, $Y$ represents the module output.

Finally, the local features from the local branch are fused with the attention features from the global branch via a residual connection. In summary, GSEA enhances discriminative semantics by jointly modeling efficient local interactions and global channel dependencies, improving robustness under complex backgrounds with modest computational overhead.

## 3 Experimental Design and Result Analysis

### 3.1 Dataset

In this study, model training is conducted on MyDataset, which is constructed by combining the open-source Road Damage Dataset 2022 (RDD2022) dataset [24] with a self-collected dataset. RDD2022 contains 47,420 images captured using smartphones and high-resolution cameras across six countries—Japan, India, the Czech Republic, Norway, the United States, and China—and includes over 55,000 annotated instances of four types of road damage: longitudinal cracks, transverse cracks, alligator cracks, and potholes. After data filtering, 3,195 valid images were retained. To further enhance the model's generalization ability in complex environments, additional images collected from online sources and field acquisition were incorporated, resulting in a final dataset of 4,920 images.

All images were randomly shuffled and annotated using the LabelImg tool in the PASCAL Visual Object Classes (VOC) format, with object locations and class labels stored in XML files. The dataset was then split into training, validation, and test sets with a ratio of 7:1:2.

The multi-country and multi-device nature of MyDataset introduces diversity in pavement texture, lane markings, camera viewpoints, and ambient illumination, which helps approximate real transportation environments. Nevertheless, some conditions remain under-represented (e.g., severe nighttime scenes, heavy rain, extreme motion blur, and rare pavement materials). Accordingly, the reported results should be interpreted as performance on the available visual conditions, and domain adaptation or additional data collection may be required for deployment in unseen environments.

### 3.2 Experimental Environment

The experimental environment configuration is shown in Table 1.

**Table 1.** Experimental environment

| Environment Settings | Configuration |
|---|---|
| System environment | Windows 11 |
| Programming language version | Python 3.10 |
| Deep learning framework | PyTorch 2.2.2 |
| GPU | RTX 4070 Ti Super (16 GB) |

All models were trained with identical hyperparameter settings, using Stochastic Gradient Descent (SGD) optimization and a cosine annealing learning rate strategy. The specific training hyperparameters are summarized in Table 2.

**Table 2.** Training hyperparameters

| Parameter | Value |
|---|---|
| Learning rate | 0.01 |
| Image size | 640 × 640 |
| Optimizer | Stochastic Gradient Descent (SGD) |
| Batch size | 32 |
| Epochs | 200 |
| Weight decay | 0.0005 |

### 3.3 Model Evaluation Indicators

In this study, the evaluation metrics include:

Giga Floating-Point Operations (GFLOPs): the number of floating-point operations required by the model, used to measure its computational complexity and estimate execution cost. Params (Parameters): the total number of model parameters, which reflects the model size and structural complexity. FPS (Frames Per Second): the number of images processed per second, serving as a key indicator of the model's real-time inference capability.

Precision (P): The proportion of true positive samples among all samples predicted as positive by the model.

$$P = \frac{TP}{TP + FP} \tag{6}$$

Recall (R): The proportion of true positive samples correctly detected by the model among all actual positive samples.

$$R = \frac{TP}{TP + FN} \tag{7}$$

Average Precision (AP): The average precision value for a single category.

$$AP = \int_0^1 P(R)dR \tag{8}$$

Mean Average Precision (mAP): The mean of Average Precision values over all object categories.
where, $N$ denotes the number of object classes, and $AP_i$ is the $AP$ of the $i$-th class.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{9}$$

### 3.4 Ablation Experiment

To evaluate the effectiveness of the proposed modules, YOLOv11n is used as the baseline for ablation experiments on the test set. Each module is incrementally removed to quantify its contribution to detection performance. The results are reported in Table 3, where bold values indicate the best results and "✓" denotes the presence of a module.

**Table 3.** Ablation study evaluating the impact of model components

| Model | PMFA | GSEA | mAP50 | P (%) | R (%) | Param (M) | GFLOPs | FPS (f/s) |
|---|---|---|---|---|---|---|---|---|
| Yolov11n | | | 67.5 | 75.4 | 58.5 | 2.58 | 6.5 | 143 |
| Model_1 | ✓ | | 68.3 | 76.1 | 61.5 | **2.13** | **5.9** | **172** |
| Model_2 | | ✓ | 69.4 | 77.6 | 62.1 | 2.37 | 7.1 | 122 |
| Ours | ✓ | ✓ | **70.7** | **78.8** | **64.4** | 2.31 | 6.3 | 158 |

Note: Partial Multi-Scale Feature Aggregation (PMFA), Grouped Semantic Enhancement Attention (GSEA) ,Giga Floating-Point Operations (GFLOPs), Parameter (Param)

As shown in Table 3, adding PMFA alone improves mAP@0.5 from 67.5% to 68.3%, precision from 75.4% to 76.1%, and recall from 58.5% to 61.5%, while reducing parameters from 2.58 M to 2.13 M . This indicates that multi-scale aggregation and partial computation help preserve small-pothole cues without increasing complexity.

Adding GSEA alone improves mAP@0.5 to 69.4% and increases precision and recall to 77.6% and 62.1%, respectively, demonstrating that efficient local-global attention strengthens semantic discrimination. When both modules are enabled, PG-YOLO achieves the best overall accuracy (70.7% mAP@0.5) with 2.31 M parameters and 6.3 GFLOPs , and it runs at 158 FPS . These results suggest that PMFA and GSEA are complementary and jointly improve detection robustness under complex roadscene conditions.

The proposed model is lightweight (2.31 M parameters and 6.3 GFLOPs at 640 × 1640 input), which makes it suitable for resource-constrained edge deployment. On a desktop GPU (RTX 4070 Ti Super), PG-YOLO achieves 158 FPS, corresponding to a per-frame latency of about 6.3 ms . Although embedded platforms typically provide lower peak throughput, the low parameter count and modest compute budget indicate that real-time operation at common camera rates (e.g., 30 FPS) is feasible with hardware acceleration. Practical deployment can further benefit from engineering optimizations such as TensorRT compilation, FP16 quantization, and efficient batching or pipelining, without changing the model structure. Therefore, PG-YOLO is designed as a perception module that can be embedded into ITS pipelines. In vehicle-mounted settings, the model can process camera streams in real time to support driver warnings, ADAS functions, or autonomous driving planning.

Overall, the ablation results in Table 3 demonstrate that both proposed modules are effective and complementary, with no performance conflict when combined, and confirm the effectiveness of the proposed method for multi-scale pothole detection.

As shown in Figure 4, the precision, recall, and mAP (mAP@50 and mAP@50–95) curves of PG-YOLO (Ours) gradually converge after approximately 150 training epochs. Compared with other models, PG-YOLO exhibits significantly smaller fluctuations, indicating a more stable training process, more balanced parameter optimization, and superior convergence and reliability.

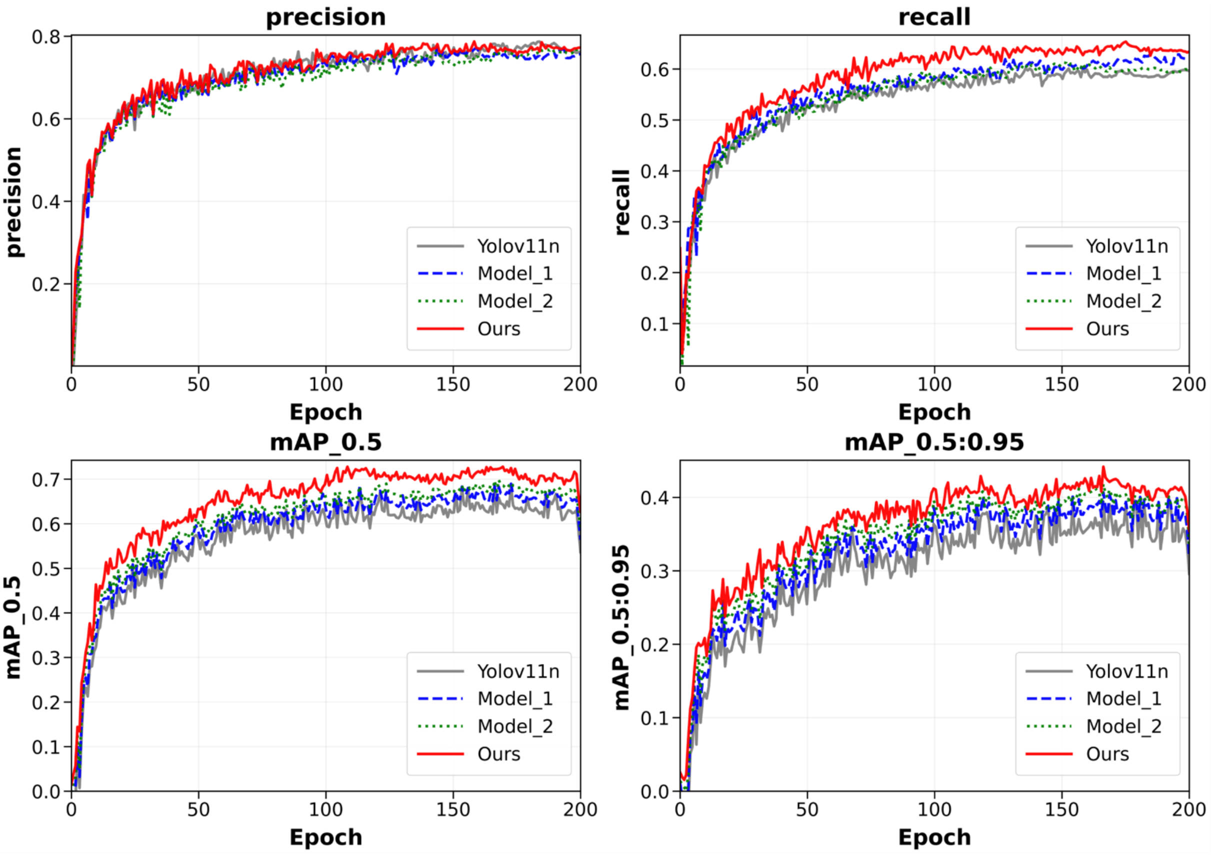

**Figure 4.** Comparison of different evaluation metrics

### 3.5 Comparative Experiment

To further validate the effectiveness and generalization capability of the proposed PG-YOLO model, comparative experiments were conducted against several state-of-the-art object detection models under identical dataset settings. The experimental results are summarized in Table 4, where bold values indicate the best performance.

As shown in Table 4, PG-YOLO achieves the highest precision (78.8%) and recall (64.4%) among the compared lightweight models, with 70.7% mAP@0.5. Compared with YOLOv11s, PG-YOLO reduces parameters by 7.10M and computation by 15.0 GFLOPs, while slightly improving precision, recall, and mAP@0.5, and increasing inference speed by 41 FPS. These results demonstrate that the proposed design improves detection accuracy without sacrificing deployment efficiency.

Overall, PG-YOLO shows strong adaptability to multi-scale pothole detection in complex road scenes, enabling more effective feature extraction when visual cues are weak and thus improving detection robustness.

### 3.6 Comparison of Detection Results

To further evaluate the effectiveness of the improved model, PG-YOLO and the original YOLOv11 were tested, and the results are illustrated in Figure 5. In Figure 5, subfigures (1) and (3) present the detection results and corresponding heatmaps generated by YOLOv11, while subfigures (2) and (4) show the detection results and heatmaps produced by PG-YOLO.

**Table 4.** Comparison results of different models on the MyDataset dataset

| Method | P (%) | R (%) | mAP50 (%) | Param (M) | GFLOPs | FPS (f/s) |
|---|---|---|---|---|---|---|
| Faster R-CNN | 60.2 | 42.6 | 48.3 | 36.3 | 190.4 | 35 |
| SSD | 53.2 | 37.1 | 41.1 | 39.5 | 160.8 | 43 |
| RT-DETR-r18 | 54.9 | 44.4 | 45.3 | 19.8 | 56.9 | 58 |
| YOLOv5n | 75.4 | 58.6 | 66.2 | 2.50 | 7.1 | **173** |
| YOLOv7tiny | 76.0 | 60.5 | 65.9 | 8.32 | 21.8 | 108 |
| YOLOv8n | 76.5 | 59.5 | 67.3 | 3.01 | 8.1 | 151 |
| YOLOv10n | 73.4 | 60.1 | 67.6 | 2.56 | 6.7 | 136 |
| YOLOv11n | 75.4 | 58.5 | 67.5 | 2.58 | 6.5 | 143 |
| YOLOv11s | 76.0 | 64.2 | 70.5 | 9.41 | 21.3 | 117 |
| YOLOv12n | 75.1 | 58.3 | 66.1 | 2.51 | **5.8** | 103 |
| YOLOv13n | 73.0 | 59.4 | 65.6 | 2.45 | 6.1 | 91 |
| **Ours** | **78.8** | **64.4** | **70.7** | **2.31** | 6.3 | 158 |

Note: Parameter (Param), Giga Floating-Point Operations (GFLOPs), Single Shot MultiBox Detector (SSD)

As can be observed from Figure 5, the original YOLOv11 model is prone to false positives and missed detections when dealing with small-sized potholes and low-light conditions. In contrast, PG-YOLO demonstrates noticeable improvements under these challenging scenarios. These results indicate that PG-YOLO consistently outperforms the original YOLOv11 in detection performance.

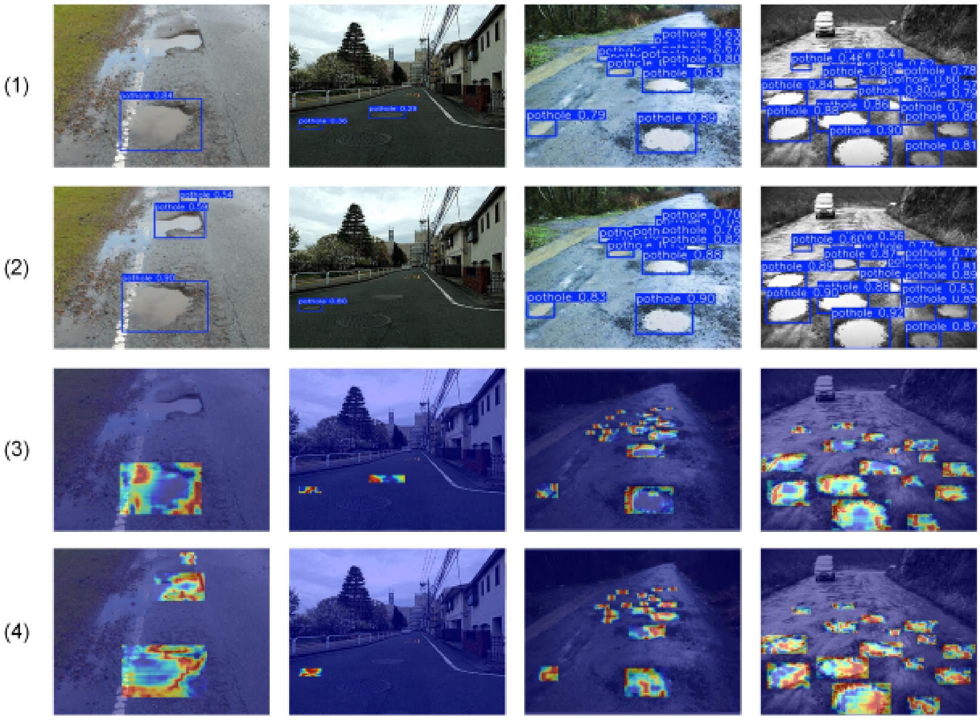

**Figure 5.** Comparison of different metrics

## 4 Conclusions

This paper presented PG-YOLO, a real-time and lightweight pothole detector tailored for intelligent transportation and mechatronic mobility applications. Building on YOLOv11n, we introduced the PMFA module to improve multi-scale feature representation and preserve fine edge cues via partial-channel aggregation, and the GSEA module to strengthen salient-region semantics through efficient local–global attention.

Experiments on a mixed pothole dataset (4,920 images) showed that PG-YOLO improves mAP@0.5, precision, and recall by 3.2, 3.4, and 5.9 percentage points over YOLOv11n while reducing parameters by 10%. PG-YOLO reaches 158 FPS on an RTX 4070 Ti Super, suggesting strong potential for real-time operation on vehicle-mounted cameras, roadside monitoring units, and inspection robots.

Future work will focus on expanding the dataset to better cover extreme transportation conditions, improving robustness via context-aware modeling and background suppression, and validating end-to-end deployment in real ITS pipelines with localization and maintenance decision modules.

## Author Contributions

Conceptualization, Q.L.C. and L.Z.; methodology, Q.L.C.; investigation, Q.L.C.; data curation, Q.L.C. and A.K.; writing—original draft preparation, Q.L.C.; writing—review and editing, L.Z., J.L.M., and A.K.; visualization, Q.L.C.; supervision, L.Z. All authors have read and agreed to the published version of the manuscript.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

[1] A. Bakirbayeva, M. Rakhimov, Z. Aubakirova, A. Aldungarova, I. Menendez, and K. Mukhambetkaliev, “Impact of pavement quality on improving logistics schemes in the transportation industry,” in *2024 IEEE 6th International Symposium on Logistics and Industrial Informatics*, Karaganda, Kazakhstan, 2024, pp. 157–162. https://doi.org/10.1109/LINDI63813.2024.10820422

[2] J. Braunfelds, U. Senkans, P. Skels, R. Janeliukstis, J. Porins, S. Spolitis, and V. Bobrovs, “Road pavement structural health monitoring by embedded fiber-bragg-grating-based optical sensors,” *Sensors*, vol. 22, no. 12, p. 4581, 2022. https://doi.org/10.3390/s22124581

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91

[4] 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, 2017. https://doi.org/10.1109/TPAMI.2016.2577031

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *European Conference on Computer Vision 2016*, Cham, Switzerland, 2016, pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

[6] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European Conference on Computer Vision 2020*, Cham, Switzerland, 2020, pp. 213–229. https://doi.org/10.1007/978-3-030-58452-8_13

[7] S. Mittal, A. Kumar, S. Negi, A. Rathi, S. Nautiyal, and M. Mittal, “Potholes detection system utilizing visual geometry group-16 (VGG16),” in *2024 8th International Conference on Parallel, Distributed and Grid Computing*, Waknaghat, Solan, India, 2024, pp. 94–97. https://doi.org/10.1109/PDGC64653.2024.10984411

[8] W. Ye, W. Jiang, Z. Tong, D. Yuan, and J. Xiao, “Convolutional neural network for pothole detection in asphalt pavement,” *Road Mater. Pavement Des.*, vol. 22, no. 1, pp. 42–58, 2019. https://doi.org/10.1080/14680629.2019.1615533

[9] J. J. Yebes, D. Montero, and I. Arriola, “Learning to automatically catch potholes in worldwide road scene images,” *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 3, pp. 192–205, 2021. https://doi.org/10.1109/MITS.2019.2926370

[10] J. Lan, H. Wang, Z. Zhu, and Q. Zhang, “Computer vision based pothole road detection and recognition,” in *2024 5th International Conference on Computer Vision, Image and Deep Learning*, Zhuhai, China, 2024, pp. 505–509. https://doi.org/10.1109/CVIDL62147.2024.10603934

[11] A. Tithi, F. Ali, and S. Azrof, “Speed bump & pothole detection with single shot multibox detector algorithm & speed control for autonomous vehicle,” in *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0*, Rajshahi, Bangladesh, 2021, pp. 1–5. https://doi.org/10.1109/ACMI53878.2021.9528185

[12] K. R. Ahmed, “Smart pothole detection using deep learning based on dilated convolution,” *Sensors*, vol. 21, no. 24, p. 8406, 2021. https://doi.org/10.3390/s21248406

[13] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, “Road damage detection and classification using deep neural networks with smartphone images,” *Comput. Aided. Civ. Infrastruct. Eng.*, vol. 33, no. 12, pp. 1127–1141, 2018. https://doi.org/10.1111/mice.12387

[14] S. S. Park, V. T. Tran, and D. E. Lee, "Application of various YOLO models for computer vision-based real-time pothole detection," *Appl. Sci.*, vol. 11, no. 23, p. 11229, 2021. https://doi.org/10.3390/app112311229

[15] K. Ko, I. Jang, J. H. Choi, J. H. Lim, and D. U. Lee, "Stochastic decision fusion of convolutional neural networks for tomato ripeness detection in agricultural sorting systems," *Sensors*, vol. 21, no. 3, p. 917, 2021. https://doi.org/10.3390/s21030917

[16] K. K. Sai, D. D. V. Kumar, A. Sahrudhay, and K. Dharavath, "Pothole detection using deep learning," in *2023 2nd International Conference on Futuristic Technologies*, Belagavi, Karnataka, India, 2023, pp. 1–6. https://doi.org/10.1109/INCOFT60753.2023.10425461

[17] M. V. Rao, S. K. Dubey, and A. Badholia, "Pothole identification and dimension approximation with YOLO Darknet CNN," in *2022 4th International Conference on Inventive Research in Computing Applications*, Coimbatore, India, 2022, pp. 1207–1213. https://doi.org/10.1109/ICIRCA54612.2022.9985669

[18] E. Saranya, R. Nivetha, S. Abirami, M. Mohaideen Arsath, and S. Dharaneesh, "Revolutionizing road maintenance: YOLO based pothole detection system," in *2024 10th International Conference on Advanced Computing and Communication Systems*, Coimbatore, India, 2024, pp. 1991–1997. https://doi.org/10.1109/ICACCS60874.2024.10716937

[19] S. H. Abdelwahed, B. K. Sharobim, B. Wasfey, and L. A. Said, "Advancements in real-time road damage detection: A comprehensive survey of methodologies and datasets," *J. Real-Time Image Process.*, vol. 22, p. 137, 2025. https://doi.org/10.1007/s11554-025-01683-1

[20] Z. Feng, Y. Guo, and Y. Sun, "Segmentation of road negative obstacles based on dual semantic-feature complementary fusion for autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 4, pp. 4687–4697, 2024. https://doi.org/10.1109/TIV.2024.3376534

[21] Z. Lin and W. Pan, "YOLO-ROC: A high-precision and ultra-lightweight model for real-time road damage detection," *Research Square*, 2025. https://doi.org/10.21203/rs.3.rs-7221917/v1

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

[23] J. Chen, S. H. Kao, H. He, W. Zhuo, S. Wen, C. H. Lee, and S. H. G. Chan, "Run, don't walk: Chasing higher FLOPS for faster neural networks," *arXiv:2303.03667*, 2023. https://doi.org/10.48550/arXiv.2303.03667

[24] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, "RDD2022: A multi-national image dataset for automatic road damage detection," *Geoscience Data J.*, vol. 11, no. 4, pp. 846–862, 2024. https://doi.org/10.1002/gdj3.260