Journal of Future Artificial Intelligence and Technologies

E-ISSN: 3048-3719

Research Article

Deploying YOLOv8 for Real-Time Road Crack Detection on Smart Road Length Measurement Devices

Trinh Luong Mien [1,*], Nguyen Dinh Tu [2], and Nguyen Van Lam [1]

[1] Department of Cybernetics, University of Transport and Communications, No.3 Cau Giay Street, 11512 Hanoi, Vietnam; e-mail : mientl@utc.edu.vn; labmit2021@gmail.com
[2] Department of Construction Machinery, University of Transport and Communications, No.3 Cau Giay Street, 11512 Hanoi, Vietnam; e-mail : nguyendinhtu@utc.edu.vn
* Corresponding Author : Trinh Luong Mien

Abstract: Nowadays, the construction, monitoring, quality control, and maintenance of roads always require high-precision, easy-to-use measuring devices in the field. This study develops an embedded computer program using the YOLOv8 model integrated into a smart road length measuring device to detect road surface cracks. First, the study analyzes and clarifies the outstanding points of the YOLOv8 model, including anchor-free: Eliminates the use of traditional anchor boxes, helping to simplify the training process, increase accuracy and reduce computational costs; C2f (Cross-Stage Partial with feature fusion); Replacing CSSPLayer in YOLOv5, this module improves feature extraction and maintains good computational performance; SPFF (Spatial Pyramid Pooling - Fast): Helps expand the receptive field without increasing computational costs, supporting object recognition at multiple scales; PAN++: Enhances the transfer of features from lower layers to the output, helping to detect small objects such as cracks well. Then, this work evaluated the performance of the proposed model by applying YOLOv8 on the surface crack dataset taken from the open-source Roboflow Universe. The results showed that the developed YOLOv8 model achieved good results with the recall indexes of 57.3%, mAP50 59.3%, precision of 64.4% mAP50-95 48.9% for YOLOv8n model, and can improve the accuracy further, reaching 79.8% using the YOLOv8m model, while still meeting the real-time processing speed of the device. These experimental results using the YOLOv8 model for detecting cracks on DongAnh road, Hanoi, Vietnam, confirm the possibility of applying the integrated road crack detection model on smart road length measuring devices in practice, achieved a precision of 43.3%, supporting the survey, inspection, supervision, construction and management of traffic infrastructure more scientifically and effectively.

Keywords: Artificial Intelligence; Crack Detection; Embedded Systems; Real-Time Processing; Smart Measuring Device; Surface Defect Recognition; Wheel-type length measuring; YOLOv8.

Received: May, 6th 2025
Revised: May, 24th 2025
Accepted: May, 26th 2025
Published: May, 29th 2025
Curr. Ver.: May, 30th 2025

Copyright: © 2025 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY SA) license (https://creativecommons.org/licenses/by-sa/4.0/)

1. Introduction

In modern society, transport infrastructure plays an extremely important role in socio-economic development, especially the extensive road network in Vietnam. Construction, supervision, inspection, assessment of quality, and maintenance of roads always require measuring equipment with high accuracy, speed, and ease of deployment in the field. In particular, road length and surface quality (cracks, ruts, subsidence, etc.) are important road indicators, directly affecting investment, construction, inspection, assessment and acceptance of road works.

Manual road surface quality inspection methods and traditional road length measurement methods such as tape measures, mechanical measuring wheels or manual measurement still have many disadvantages: low accuracy, easily affected by environmental factors, subjective dependence on the measurer, lack of automatic data storage and processing capabilities. Therefore, designing and manufacturing a smart road length measuring device and integrating

artificial intelligence technology to detect road surface cracks is a potential and necessary direction in the context of the digital economy and society.

The application of modern technologies, such as artificial intelligence and, internet of thing is becoming popular in social life [1], transportation [2]–[4], especially deep learning to detect surface cracks in construction and traffic works, is attracting the attention of many scientists around the world [5]–[18]. These studies have achieved certain results in proposing concrete crack detection models [6,7], bridge defect detection models [12], [16], road surface defect crack detection models [9-11], identification of stone cracks [18], crack detection in tiled sidewalks [7], pavement crack detection by YOLOv3 [8], bridge-pavement crack detection using deep learning [9], detect road cracks using two-stage CNN [10].

The advantages of deep learning methods include their ability to automatically extract relevant features, maintain high accuracy under complex lighting and environmental conditions, and adapt to various data types[5], [6], [14]. However, these models typically demand large training datasets, require high computational power, and are challenging to deploy directly on embedded systems without extensive optimization [11], [16].

A key limitation in prior studies is that most crack detection models are tested under laboratory or idealized conditions, where noise and environmental variability are minimized. Only a few works have evaluated the performance of these models when deployed directly on real-world embedded devices under outdoor conditions, where lighting, surface debris, and weather introduce substantial challenges [13], [15]. Moreover, limited research exists on combining crack detection with road length measuring systems into a unified platform, despite this integration being crucial for comprehensive, automated road infrastructure monitoring.

This study proposes a smart road length measuring device embedded with real-time crack detection capabilities using the YOLOv8 model to address these gaps. YOLOv8 is selected due to its anchor-free architecture and optimized backbone (C2f), which improve both computational efficiency and detection accuracy [17], [19], [20]. Its lightweight versions (YOLOv8n/s/m) offer a balanced trade-off between performance and speed, enabling deployment on edge devices such as Jetson Nano for field operations. In particular, YOLOv8's compatibility with formats like ONNX and TensorRT facilitates fast, low-latency inference [20], [21]. These features make YOLOv8 practical and scalable for real-time, on-device road surface inspection in dynamic outdoor environments. Accordingly, the objectives of this study are threefold:

1. To design and implement a smart road length measuring device that integrates road crack detection capabilities using a deep learning model.
2. To evaluate the performance of various YOLOv8 model versions (YOLOv8n, YOLOv8s, YOLOv8m) in detecting road surface cracks under controlled and real-world field conditions.
3. To demonstrate the feasibility of deploying a lightweight, real-time object detection model on embedded hardware (Jetson Nano), ensuring operational efficiency in outdoor environments.

The remainder of this paper is structured as follows: Section 2 describes the architecture and components of the smart road length measuring device. Section 3 explains the application and integration of the YOLOv8 model into the system. Section 4 presents the experimental setup, dataset, and results, followed by a discussion. Finally, Section 5 concludes the study and outlines potential directions for future work.

2. Development of the smart road length measuring device

The smart road length measuring device proposed in this study uses a measuring wheel with an integrated encoder sensor, which directly rolls on the road surface to determine the distance the wheel has travelled, from which the central microprocessor calculates and displays the distance and total length of the road section on the device screen, and simultaneously transmits data and images of the road to the data storage centre. This smart road length measuring device is also integrated with a camera and an image processing software program running on an embedded computer mounted on the device, allowing the detection of cracks, ruts, and depressions on the road surface. The basic specifications of the smart road length measuring device are as follows:

- Measuring device size (adjustable): (450-950) x 160 x 140 mm

- Measuring wheel diameter: 159 mm
- Measuring length., continuous display: 0-9999.9m
- Measurement accuracy: ±0.5%

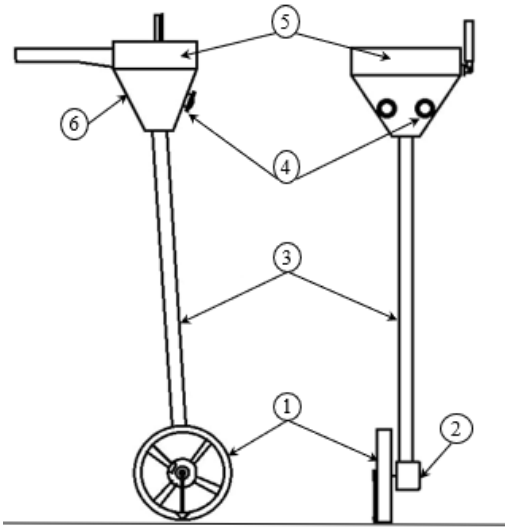

Figure 1. Main components of smart road length measuring device

The main components of the smart road length measuring device include (1)- measuring wheel, (2)- length measuring encoder sensor, (3)- an adjustable length measuring device body, (4)- dual camera to take photos and videos of the road surface; (5)- LCD screen displays the measured road length and warns of road surface defects; (6)- mini control cabinet including control circuit, communication & measurement signal processing, embedded computer & image processing program to detect road surface defects. The block diagram of the control system structure signal processing - data of the smart road measuring device is presented in Figure 2.

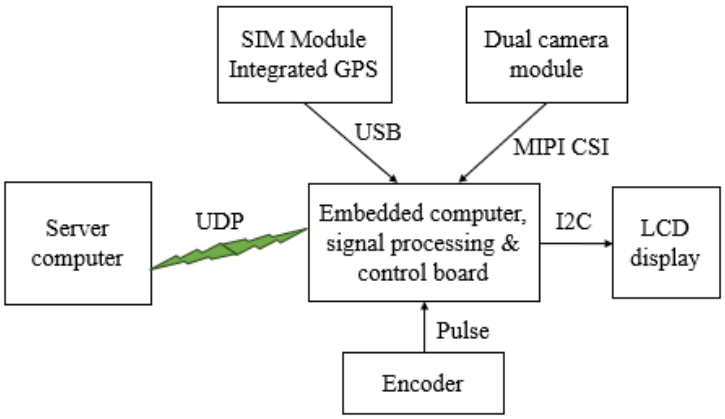

Figure 2. Structure of the smart road length measuring device control system

In this study, the main devices selected to create the smart odometer are as follows: 7600GH GPS Integrated Sim module; Jetson Nano 4GB embedded computer; IMX219-83 stereo camera module; Encoder sensor measures length five pulses per rotation; LCD 1602, 2004 (driver HD44780) I2C communication. Jetson Nano was chosen for its balance of AI processing performance (integrated CUDA GPU) and low power consumption, making it well-suited for real-time applications in the field.

3. Application of YOLOv8 model integrated on smart road length measuring device to detect road surface cracks

The YOLO model, developed by Joseph Redmon in 2016 [21], is one of computer vision's fastest and most accurate object detection models. The most notable difference from previous methods is that the YOLO model treats object detection as a single regression problem, using a convolutional neural network (CNN) to predict both the location and class of an object at the same time [21], allowing the YOLO model to process images at a significantly faster rate than two-stage models such as R-CNN. The YOLO model divides the image into grid cells, each responsible for predicting objects whose centres lie within that cell. Each cell predicts parameters that define the bounding box, including the centre coordinates, width, height, and the probability of detecting objects belonging to certain classes. This approach allows the YOLO model to perform object detection in a single pass through the network, providing real-time performance that previous methods have found difficult to achieve. The architecture of the YOLO model is built from three main components: backbone, neck & head. The backbone performs feature extraction, using a convolutional neural network to convert image data into multi-scale feature maps. The neck plays the role of an intermediate processing stage, using special layers to synthesize and enhance feature representations at different scales. The head plays the role of a prediction mechanism, producing the final result for object localization and classification based on the refined feature map [20], [21].

Ultralytics developed the YOLOv8 model in 2023 [20], [21]. It is a powerful and flexible model for many computer vision problems, optimized for high performance and easy deployment, including many versions, such as YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x, suitable for many different deployment environments from embedded devices to powerful GPU systems. The YOLOv8 model has an architecture consisting of three main components: backbone, neck and head. The model uses an improved backbone with the C2f module replacing CSPLayer in CSPDarknet, which enhances feature extraction and maintains computational efficiency, see Figure 3. The neck integrates Spatial Pyramid Pooling Fast (SPPF) and PAN++ to process information at multiple feature levels, supporting detecting objects with diverse sizes and locations. In particular, the YOLOv8 model adopts an anchor-free architecture and separates the head design between prediction branches (objectness, classification and regression), which helps the model learn more efficiently and improves overall accuracy. During training, the YOLOv8 model utilizes many new techniques such as Mosaic and MixUp image synthesis and enrichment, mixed-precision training, and loss functions such as CIoU and Distribution Focal Loss. These improvements help the model learn faster and more stably, especially effective in detecting small objects or complex environments.

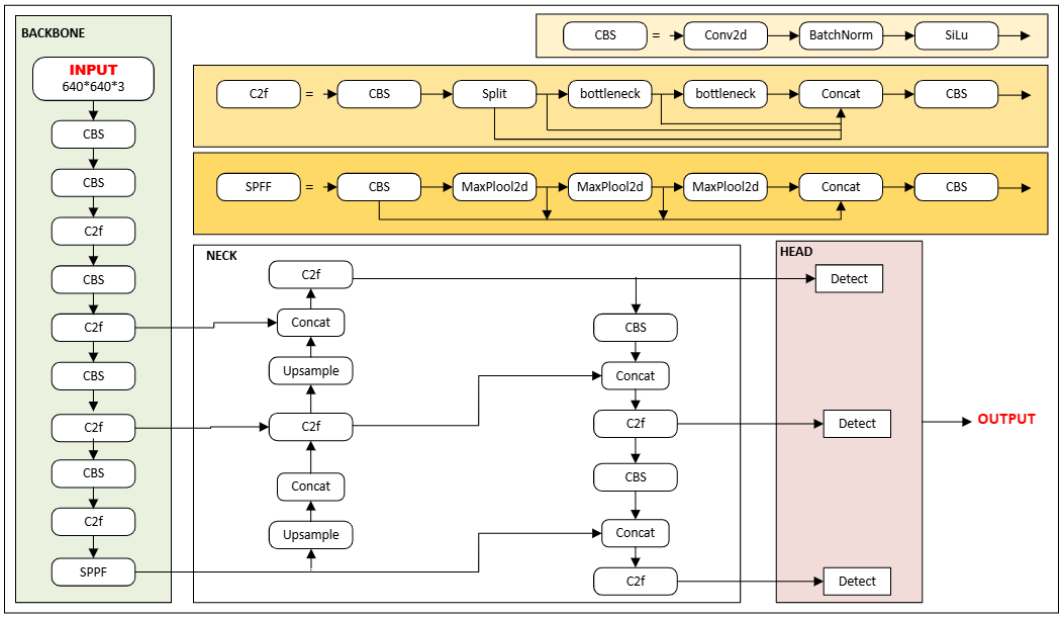

Figure 3. YOLOv8 model architecture diagram

Outstanding features of YOLOv8: Anchor-free: YOLOv8 eliminates the use of traditional anchor boxes, helping to simplify the training process, increase accuracy and reduce computational costs; C2f (Cross-Stage Partial with feature fusion): Replacing CSSPLayer in

YOLOv5, this module improves feature extraction and maintains good computational performance; SPPF (Spatial Pyramid Pooling - Fast): Helps expand the receptive field without increasing computational costs, supporting object recognition at multiple scales; PAN++: Enhances the transmission of features from lower layers to the output, helping to detect small objects such as cracks well.

In this work, there are two main reasons for choosing YOLOv8: firstly, the training and deployment process of YOLOv8 is simpler due to the support library providing Ultralytics, and secondly, the anchor-free mechanism allows for faster model inference. The YOLOv8 model achieves an optimal balance between accuracy and inference speed, which is suitable for real-time systems. The YOLOv8n version is particularly lightweight and easily deployed on Jetson Nano while still achieving superior accuracy compared to lighter models such as Tiny-YOLO or MobileNet-SSD. In addition, good support for formats such as ONNX and TensorRT helps optimize model performance when deployed on embedded hardware, meeting the requirements of continuous operation in the field. This work uses the following metrics to evaluate the YOLOv8 model.

- Recall measures the ratio of correct detections to the total actual objects.

$$R = \frac{TP}{TP + FN} \tag{1}$$

- Precision measures the ratio of correct predictions to total positive predictions.

$$P = \frac{TP}{TP + FP} \tag{2}$$

Where FN (False Negatives) is the number of missed objects, TP (True Positives) is the number of correct detections, FP (False Positives) is the number of false detections.

- Average Precision (AP) represents the average precision corresponding to the recall variation from 0 to 1.

$$AP = \int_0^1 P(R)dR \tag{3}$$

Where $P(R)$ precision as a function of recall R, dR is the differential element of recall. The integral is taken over the entire recall range from 0 to 1

- The mAP50 index is the mAP (average of AP values) calculated at a fixed Intersection over the Union (IoU) threshold of 0.5. It evaluates the model's accuracy when requiring a minimum 50% overlap between the prediction and the actual object.

$$mAP50 = \frac{1}{m}\sum_{i=1}^{m} AP_i \text{ (with } IoU = 0.5) \tag{4}$$

Where m number of object classes, AP_i is the average precision of class i.

- The mAP50-95 metric, similar to mAP50, mAP50-95 averages the AP for each class at different IoU thresholds from 0.5 to 0.95 (increasing in steps of 0.05), and then averages those AP values across all classes. This comprehensive aggregate metric reflects the model's performance under various levels of object localization rigor.

$$mAP50-95 = \frac{1}{N}\sum_{t\in\{0.5,0.55,\ldots,0.95\}} \frac{1}{m}\sum_{i=1}^{m} AP_i \text{ (with } IoU = t) \tag{5}$$

Where t is IoU threshold values, N is the number of thresholds, AP_i is the average precision of class i at threshold t.

4. Experimental Results

4.1. Dataset and Training Configuration

This study's training and evaluation process utilized a publicly available dataset from Roboflow Universe, comprising 3,913 annotated color images depicting diverse instances of road surface cracks. These images span a variety of settings, including urban and rural environments, and incorporate noise conditions such as shadows, faded paint, leaves, and debris.

Each image is annotated in YOLO format, where each line in the label file denotes a bounding box: [class_id, x_center, y_center, width, height], all values normalized to image dimensions.

This study focused on a single class "crack" with each image containing between one to five cracks of varying shapes (e.g., horizontal, vertical, diagonal) and degrees of severity. The original resolutions of the images vary, but all were resized to 640×640 pixels to ensure compatibility with the YOLOv8 model architecture. The dataset was divided into 92% for training, 5% for validation, and 3% for testing, providing a balanced distribution for model learning and performance assessment. To improve generalization, extensive data augmentation techniques were applied to simulate real-world variations and enhance robustness. These augmentations included geometric transformations, color jittering, and compositional techniques. The complete list is summarized in Table 1.

Table 1. Data Augmentation Techniques Applied During Training.

Augmentation Type	Parameters/Settings
Flip	Horizontal
Rotation	±15°, 90° (Clockwise, Counter-Clockwise, Upside Down)
Zoom & Crop	0% Minimum Zoom, up to 30% Maximum Zoom
Hue	Range: –10% to +10%
Saturation	Range: –25% to +25%
Brightness	Range: –20% to +20%
Blur	Up to 1 pixel
Others	Mosaic, MixUp, contrast enhancement

The models were trained using Google Colab with an NVIDIA Tesla T4 GPU. Three YOLOv8 variants, i.e., YOLOv8n, YOLOv8s, and YOLOv8m, were evaluated. Each model was trained for 50 epochs with a batch size of 16, using an initial learning rate of 0.01 that gradually decreased to 0.0001 to ensure convergence. Stochastic Gradient Descent (SGD) was used as the optimizer. The Complete IoU (CIoU) loss function was used for bounding box regression, while Distribution Focal Loss was applied for classification. Figure 4 illustrates an example of an annotated training image with bounding boxes marking detected cracks.

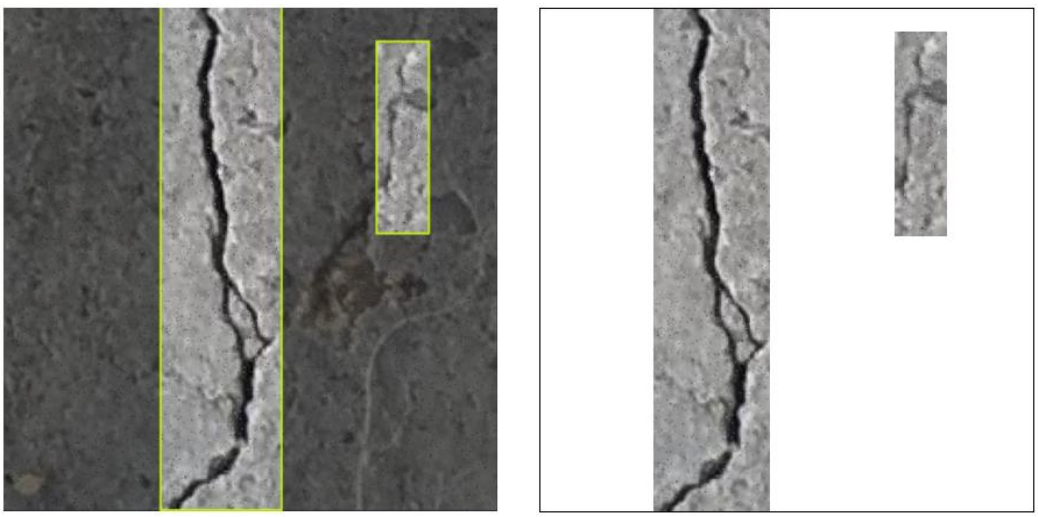

Figure 4. Image with bounding box and road cracks

4.2. Deployment on Embedded Smart Device

The trained YOLOv8 models were deployed onto an embedded smart device for real-time crack detection and road length measurement to test real-world applicability. The embedded system was built around a Jetson Nano 4GB module and equipped with a stereo camera and rotary encoder to capture road imagery and calculate distance simultaneously. The lightweight YOLOv8n model was selected for deployment due to its smaller footprint and

faster inference capability. The model was converted to ONNX format and further optimized using TensorRT to minimize latency and maximize throughput on the embedded GPU.

The deployment pipeline includes image acquisition from the stereo camera, image resizing and normalization, real-time inference using the Ultralytics inference API, and direct display of bounding boxes and class labels on the onboard LCD screen. Processed data and length measurements are also transmitted to a centralized storage system for later analysis. The complete embedded specifications are listed in Table 2.

Table 2. Specifications of Embedded Smart Road Measuring Device.

Component	Description
Processor	Jetson Nano 4GB
Model Deployed	YOLOv8n (converted to ONNX and optimized via TensorRT)
Input Image Size	640 × 640 pixels
Software Stack	Python, OpenCV, Ultralytics Inference API
Inference Speed	Approximately 10–15 FPS
Latency	~100–150 ms
Output	LCD with bounding boxes and labels, data logging enabled
Operation Mode	MAXN (maximum power and performance mode)

4.3. Implementation Results and Analysis

The results from experiments conducted on the Roboflow test dataset are presented in Table 3, comparing precision, recall, mAP50, and mAP50–95 metrics across the three YOLOv8 model variants. The results confirm that model performance improves with size: YOLOv8m outperforms YOLOv8s and YOLOv8n in all metrics, although at the cost of slower inference speed, making it less ideal for embedded real-time applications.

Table 3. Experimental Results on Roboflow Test Set.

Model	Precision	Recall	mAP50	mAP50–95
YOLOv8n	64.4%	57.3%	59.3%	48.9%
YOLOv8s	70.5%	65.4%	68.6%	55.6%
YOLOv8m	79.8%	72.8%	76.6%	62.5%

Figure 5 displays the confusion matrix, highlighting the classification accuracy of crack and background categories. The YOLOv8n model achieved 94 correctly predicted crack instances, with 36 false negatives and 54 false positives. Visual results of the detection performance on test images are shown in Figure 6, which illustrates sample images with bounding boxes and confidence scores generated by the YOLOv8n model on the Roboflow test set.

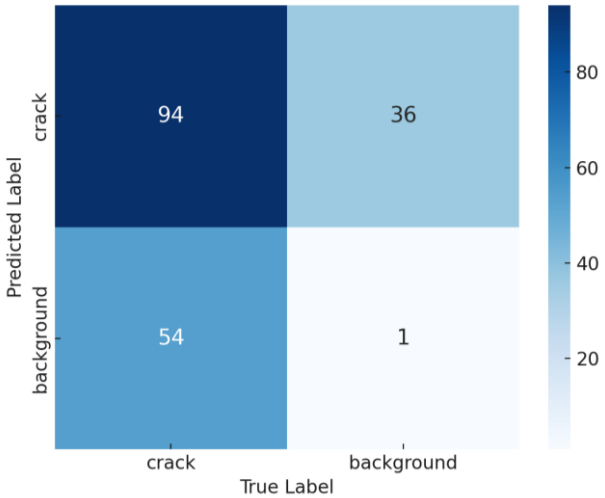

Figure 5. The confusion matrix testing on the Roboflow dataset

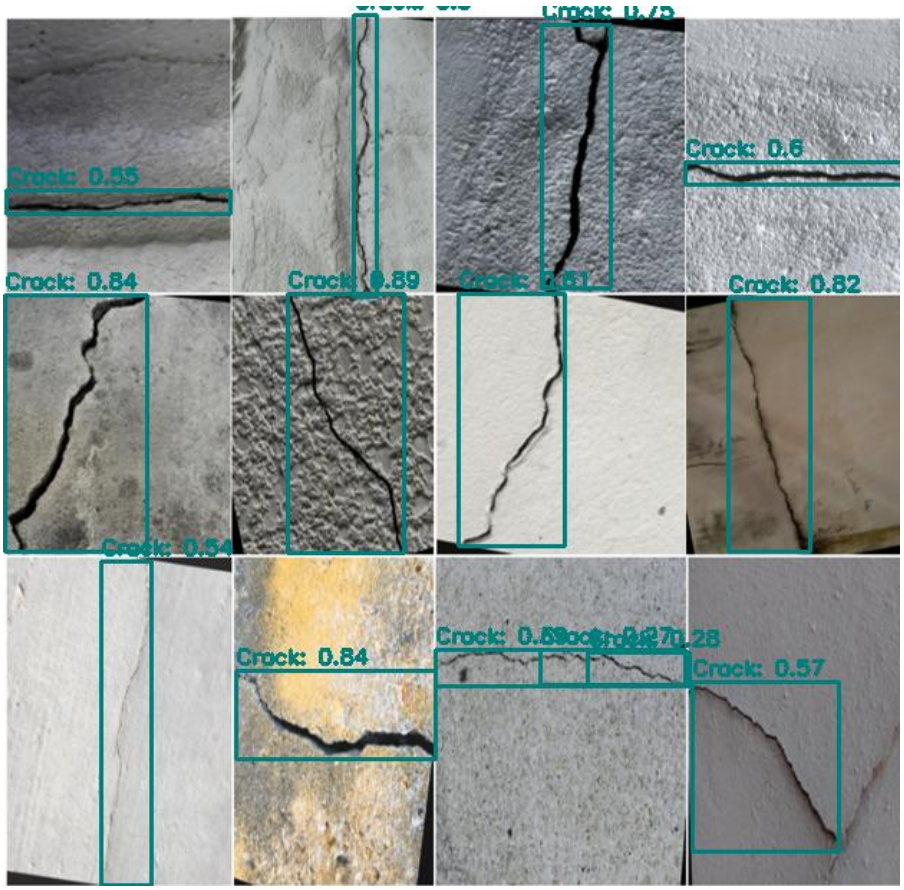

Figure 6. Experimental results of detecting cracks on the road using the developed YOLOv8 model

The system was tested on DongAnh road in Hanoi, Vietnam, under dry weather conditions to evaluate field applicability further. Approximately 58 images were collected in situ using the smart device. The images contained realistic noise—dirt, shadows, dry leaves, and uneven surfaces, not present in the Roboflow training set. These field experiments served as qualitative validation, showing that the YOLOv8n model could still detect the most visible cracks and display results in real time on the device. Although lower than in the Roboflow test set, the field performance was as follows: precision reached 43.3%, recall reached 27.8%, and mAP50 reached 25.8%. These figures confirm the model's ability to operate in challenging, real-world environments despite domain shifts. Visual examples of the field test and interface display are shown in Figure 7 and Figure 8.

Figure 7. Detecting cracks on DongAnh road using the developed YOLOv8 model

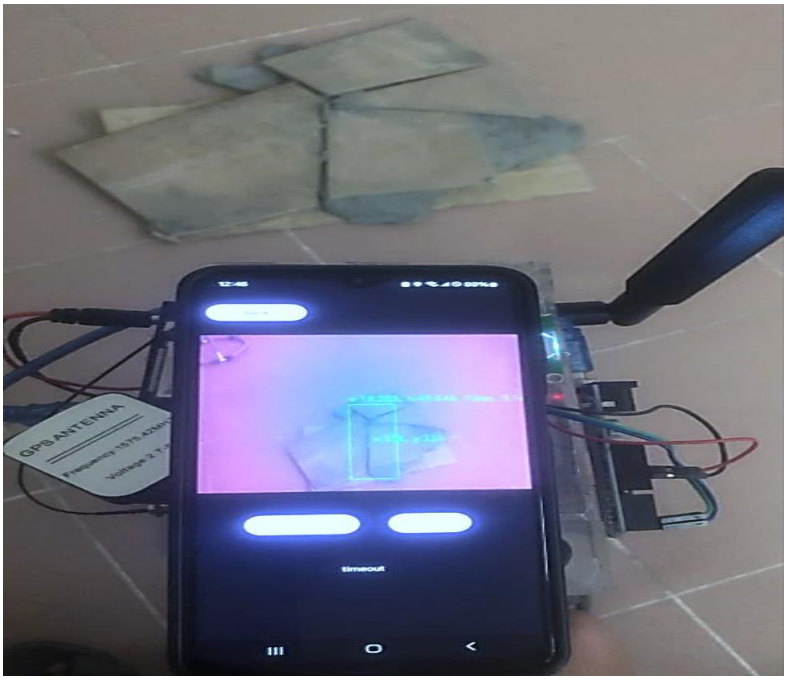

Figure 8. Field experiment on DongAnh road using road length measuring device

In summary, the model's accuracy scales with its complexity, as reflected in the experimental results. However, YOLOv8n offers an ideal trade-off between processing efficiency and detection performance for real-time edge deployment. Future improvements may include fine-tuning the model using locally collected data to increase robustness against context-specific environmental noise.

5. Conclusions

In this work, we have researched and developed an embedded computer program applying the YOLOv8 model to detect road cracks integrated on a smart road length measuring device, helping this measuring device not only measure length, store data, transmit to the centre, but also check the quality of road cracks. Evaluating the performance of the road crack detection model applying YOLOv8 on the open source Roboflow Universe dataset, including 3913 crack images located at many different locations, shows that the model achieves good metrics: recall with 57.3%, mAP50 with 59.3% and precision with 64.4%, with the YOLOv8n model and achieves higher precision, 79.8% with the YOLOv8m model. Some experimental results of detecting road cracks in DongAnh road, Hanoi, Vietnam, under normal weather conditions, show that the developed YOLOv8 model achieves high-performance precision of 43.3%, confirming the effectiveness of implementing the real-time road crack detection model on smart road measuring devices-basic hardware configuration requirements, processing speed meets the requirements.

However, this study also shows some challenges encountered when deploying the YOLOv8 model to detect road cracks, developed on real-world road length measuring devices, affecting the accuracy and stability of the model. Some of the main challenges are changing environmental conditions, uneven lighting, weather conditions, rain, fog, and wet road surfaces. Also creates light reflections, causing interference to the model, dirt, debris or crack size characteristics or limitations in training data, unoptimized field equipment models or vibrations, movements, and device movement speeds when collecting data can blur the image, affecting the input quality of the model. In the future, the research will focus on attention mechanisms, optimizing convolutional blocks of the YOLO model, version v8 or higher, to help the model focus better on crack features, especially small or faint cracks in complex, noisy environments, enhancing the ability to extract features while reducing computational costs, creating a lighter and faster model.

Author Contributions: Conceptualization: T.L.M and N.D.T; Methodology: T.L.M.; Software: T.L.M and N.V.L; Validation: T.L.M and N.D.T; Formal analysis: T.L.M; Investigation: N.V.L; Resources: N.D.T and N.V.L; Data curation: T.L.M and N.V.L; Writing—original draft preparation: T.L.M and N.V.L; Writing—review and editing: T.L.M and N.V.L; Visualization: T.L.M and N.V.L.; Supervision: T.L.M and N.D.T; Project administration: T.L.M; Funding acquisition: T.L.M and N.D.T; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data can be supported upon request.

Acknowledgments: The authors would like to thank the University of Transport and Communications for supporting this research (project numberT2025-DT-005).

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] T. L. Mien, V. Van An, and T. T. Huong, "Research and Development of the Pupil Identification and Warning System using AI-IoT," *J. Robot. Control*, vol. 3, no. 4, pp. 528–534, Jul. 2022, doi: 10.18196/jrc.v3i4.14978.

[2] T. Luong Mien and V. Van Duy, "Development of the System of Monitoring Traffic Vehicle Volume and Density on the Vietnam's Street," in *2022 11th International Conference on Control, Automation and Information Sciences (ICCAIS)*, Nov. 2022, pp. 292–297. doi: 10.1109/ICCAIS56082.2022.9990367.

[3] V. T. Ha, T. N. Tu, N. T. Dung, T. L. Mien, and C. T. Thuy, "Deep Q-Network (DQN) Approach for Automatic Vehicles Applied in the Intelligent Transportation System (ITS)," in *2023 International Conference on System Science and Engineering (ICSSE)*, Jul. 2023, pp. 527–532. doi: 10.1109/ICSSE58758.2023.10227206.

[4] T. L. Mien and V. Van Duy, "Research on Building the System of the Identification and Detection Traffic Violations Vehicles at Vietnamese Intersections," in *Lecture Notes in Networks and Systems*, vol. 602 LNNS, Springer, Cham, 2023, pp. 831–838. doi: 10.1007/978-3-031-22200-9_87.

[5] R. Ali, J. H. Chuah, M. S. A. Talip, N. Mokhtar, and M. A. Shoaib, "Structural crack detection using deep convolutional neural networks," *Autom. Constr.*, vol. 133, p. 103989, Jan. 2022, doi: 10.1016/j.autcon.2021.103989.

[6] Y. Choi, H. W. Park, Y. Mi, and S. Song, "Crack Detection and Analysis of Concrete Structures Based on Neural Network and Clustering," *Sensors*, vol. 24, no. 6, p. 1725, Mar. 2024, doi: 10.3390/s24061725.

[7] Q. Qiu and D. Lau, "Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images," *Autom. Constr.*, vol. 147, p. 104745, Mar. 2023, doi: 10.1016/j.autcon.2023.104745.

[8] M. Nie and C. Wang, "Pavement Crack Detection based on yolo v3," in *2019 2nd International Conference on Safety Produce Informatization (IICSPI)*, Nov. 2019, pp. 327–330. doi: 10.1109/IICSPI48186.2019.9095956.

[9] T. Chen *et al.*, "Pavement crack detection and recognition using the architecture of segNet," *J. Ind. Inf. Integr.*, vol. 18, p. 100144, Jun. 2020, doi: 10.1016/j.jii.2020.100144.

[10] N. H. T. Nguyen, S. Perry, D. Bone, H. T. Le, and T. T. Nguyen, "Two-stage convolutional neural network for road crack detection and segmentation," *Expert Syst. Appl.*, vol. 186, p. 115718, Dec. 2021, doi: 10.1016/j.eswa.2021.115718.

[11] C. M. Yeum and S. J. Dyke, "Vision-Based Automated Crack Detection for Bridge Inspection," *Comput. Civ. Infrastruct. Eng.*, vol. 30, no. 10, pp. 759–770, Oct. 2015, doi: 10.1111/mice.12141.

[12] A. Cubero-Fernandez, F. J. Rodriguez-Lozano, R. Villatoro, J. Olivares, and J. M. Palomares, "Efficient pavement crack detection and classification," *EURASIP J. Image Video Process.*, vol. 2017, no. 1, p. 39, Dec. 2017, doi: 10.1186/s13640-017-0187-0.

[13] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic Road Crack Detection Using Random Structured Forests," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3434–3445, Dec. 2016, doi: 10.1109/TITS.2016.2552248.

[14] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, pp. 3708–3712. doi: 10.1109/ICIP.2016.7533052.

[15] L. Ngo, C. L. Xuan, H. M. Luong, B. N. Thanh, and D. B. Ngoc, "Designing image processing tools for testing concrete bridges by a drone based on deep learning," *J. Inf. Telecommun.*, vol. 7, no. 2, pp. 227–240, Apr. 2023, doi: 10.1080/24751839.2023.2186624.

[16] P. Kaur, B. S. Khehra, and E. B. S. Mavi, "Data Augmentation for Object Detection: A Review," in *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug. 2021, pp. 537–543. doi: 10.1109/MWSCAS47672.2021.9531849.

[17] Y. Liu, Y. Liu, X. Guo, X. Ling, and Q. Geng, "Metal surface defect detection using SLF-YOLO enhanced YOLOv8 model," *Sci. Rep.*, vol. 15, no. 1, p. 11105, Apr. 2025, doi: 10.1038/s41598-025-94936-9.

[18] A. M. Mayya and N. F. Alkayem, "Triple-stage crack detection in stone masonry using YOLO-ensemble, MobileNetV2U-net, and spectral clustering," *Autom. Constr.*, vol. 172, p. 106045, Apr. 2025, doi: 10.1016/j.autcon.2025.106045.

[19] M. Çiftçi, M. U. Türkdamar, and C. Öztürk, "Leveraging YOLO Models for Safety Equipment Detection on Construction Sites," *J. Comput. Theor. Appl.*, vol. 1, no. 4, pp. 492–506, May 2024, doi: 10.62411/jcta.10453.

[20] C.-Y. Wang and H.-Y. M. Liao, "YOLOv1 to YOLOv10: The Fastest and Most Accurate Real-time Object Detection Systems," *APSIPA Trans. Signal Inf. Process.*, vol. 13, no. 1, 2024, doi: 10.1561/116.20240058.

[21] J. Solawetz and Francesco, "What is YOLOv8? A Complete Guide," *Roboflow Blog*, 2024. https://blog.roboflow.com/what-is-yolov8/