Received: 30 November 2024 | Accepted: 11 February 2025

DOI: 10.1111/mice.13446

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Crack segmentation-guided measurement with lightweight distillation network on edge device

Jianqi Zhang[1,2] | **Ling Ding**[3] | **Wei Wang**[1] | **Hainian Wang**[4] | **Ioannis Brilakis**[5] | **Diana Davletshina**[5] | **Rauno Heikkilä**[6] | **Xu Yang**[2,4]

[1]School of Information Engineering, Chang'an University, Xi'an, China

[2]School of Future Transportation, Chang'an University, Xi'an, China

[3]School of Transportation Engineering, Chang'an University, Xi'an, China

[4]School of Highway, Chang'an University, Xi'an, China

[5]Department of Engineering, University of Cambridge, Cambridge, UK

[6]Faculty of Technology, University of Oulu, Oulu, Finland

Correspondence
Ling Ding, School of Transportation Engineering, Chang'an University, Xi'an 710064, China.
Email: dingling@chd.edu.cn

Ioannis Brilakis, Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK.
Email: ib340@cam.ac.uk

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 52378431, 52408454; Fundamental Research Funds for the Central Universities, Grant/Award Numbers: 300102210302, 300102210118, 300102244712; 111 Project of Low Carbon Smart Road Infrastructure Construction and Maintenance Discipline Innovation and Talent Introduction Base of Shaanxi Province, Scientific Innovation Practice Project of Postgraduates of Chang'an University, Grant/Award Number: 300103724053

Abstract

Pavement crack measurement (PCM) is essential for automated, precise road condition assessment. However, balancing speed and accuracy on edge artificial intelligence (AI) mobile devices remains challenging. This paper proposes a real-time PCM framework for edge deployment, incorporating a lightweight distillation network and a surface feature measurement algorithm. Specifically, the proposed instance-aware hybrid distillation module combines feature-based and relation-based knowledge distillation, leveraging crack instance-related information for efficient knowledge transfer from teacher to student networks, which results in a more accurate and lightweight segmentation model. Additionally, a real-time crack surface feature measurement algorithm, based on distance mapping relationships and crack edge coordinate extraction, addresses issues with crack edge branching and loss, enhancing measurement efficiency. Real-time measurement was performed on actual roads utilizing mobile robot equipped with an edge computing unit. The crack segmentation precision reached 84.37%, with a frame per second of 77.72. Compared to the ground truth, the relative error for average crack width ranged from 6.42% to 40.65%, while the relative error for crack length varied between 1.48% and 3.76%. These findings highlight the feasibility of real-time crack assessment and save road maintenance costs.

1 | INTRODUCTION

Pavement crack measurement (PCM) plays a crucial role in precise road condition assessment, especially for pavements with irregular and slender cracks (J. Zhang et al., 2023; Zhang, Yang, Wang, Guan, et al., 2024). The realization of real-time crack measurement hinges on the inference speed of edge-deployed segmentation models

and robust measurement processing (X. Yang et al., 2024). The inference speed of segmentation on edge devices is restricted, and irregular cracks frequently display localized branching and discontinuities, thereby causing measurement errors. At present, the real-time performance and robustness of detection and measurement on mobile devices are relatively weak (Zhang, Yang, Wang, Brilakis, Wang, & Ding, 2024). Efficient pavement condition assessment contributes to guiding the allocation of highly specific pavement condition index or other condition measurements. Consequently, there is an urgent necessity to develop integrated systems that combine embedded deep learning and autonomous mobile robots to accomplish real-time crack measurement.

Recently, profound advancements in visual object segmentation and robotic vision measurement have been propelled by deep convolutional neural networks (CNNs; García-Aguilar et al., 2023; J. Wang et al., 2022). Nevertheless, the majority of deep learning techniques for object segmentation are customized for notably regular objects, and their efficacy in the real-time segmentation of elongated cracks with extreme aspect ratios, which is of utmost importance for automated road condition assessment, has not been thoroughly investigated (Han et al., 2024). Additionally, robotic measurement generally focuses on regular or large-scale objects, overlooking the online measurement of complex crack topologies that are crucial for precise pavement evaluation (J. Chen et al., 2022). Crack measurement is confronted with two principal challenges: the restricted inference speed of edge-computing segmentation models and the proneness to interference during online crack parameter measurement.

The inference speed within the context of edge computing is hampered by high-precision CNN models. These models commonly possess deeper network architectures and necessitate high-resolution input images. There exists a positive correlation between the size of model parameters and the latency of inference (De Nardin et al., 2023). In recent times, the runtime inference pertaining to crack segmentation tasks has garnered enhanced attention from the research community. Zhu et al. (2024) proposed RHACrackNet, a lightweight model designed for road crack recognition. This model integrates hybrid attention modules and residual blocks. Zheng et al. (2024) introduced the T-Seg network, which effectively combines soft and hard labels via knowledge distillation, thereby augmenting computational efficiency. However, the unevaluated incorporation of lightweight methods into crack segmentation tasks may precipitate adverse consequences in terms of potential accuracy loss, thus signifying the necessity for further in-depth research in this domain.

Current online measurement capabilities are restricted and prone to interference, thereby giving rise to substantial measurement errors (Li et al., 2024). Complex and variable environments can exert a significant influence on measurement accuracy. Recent endeavors have addressed certain aspects of crack quantification. Deng et al. (2023) devised a segmentation and measurement method by combining an improved Residual U-Net with a novel crack surface feature measurement algorithm, which enhanced measurement precision. Xuefei Wang et al. (2023) introduced a novel neural network that enables more accurate and efficient crack pixel extraction and geometric feature estimation. Xu et al. (2023) developed a crack parameter measurement algorithm based on the microelement method using the YOLOv5-IDS deep learning model to enhance accuracy and efficiency and ensure the completeness of measurement. Although these advancements have significantly improved measurement accuracy and had a positive impact on crack parameter evaluation, challenges still remain. These challenges involve measurement errors caused by issues such as local branching and pixel loss during the crack skeleton extraction and measurement processes of traditional algorithms, highlighting the urgent need for robust real-time crack parameter measurement methods.

To overcome the challenges posed by the large model sizes of crack segmentation, which restrict edge computing inference speed, and the complexity and inadequate real-time performance of crack measurements, this paper presents a real-time PCM framework. This framework incorporates two key components: a lightweight distillation network and a surface feature measurement algorithm. Specifically, the proposed instance-aware distillation, leveraging crack instance information for effective knowledge transfer from teacher to student. Additionally, a real-time crack surface feature measurement algorithm is designed, utilizing distance mapping relationships and crack edge coordinate extraction based on robot camera calibration. Extensive experiments were conducted using a self-developed robot equipped with edge computing units on actual road. Our contributions are presented as detailed below.

1. Addressing the problem of crack measurement in road precision condition assessment, the real-time PCM framework was proposed.
2. Proposing a crack instance-aware hybrid distillation (IHD) method, which enables the student model to acquire and transfer logical reasoning capabilities similar to the teacher model, reducing computational complexity.
3. Designing a surface feature measurement algorithm to achieve crack measurement. By utilizing the distance mapping relationship calibrated by a robotic camera and extracting crack edge coordinates, the algorithm

effectively addresses issues such as edge branching and discontinuities at crack ends during measurement, significantly enhancing the efficiency of online real-time crack measurement.

The arrangement is: Section 2 discusses the previous research. Section 3 provides our methodology. Section 4 provides the experimental verification. Section 5 offers a summary and future research.

2 | RELATED WORKS

The literature pertinent to the proposed real-time PCM is examined.

2.1 | Lightweight crack segmentation

A lightweight crack segmentation endeavors to diminish model complexity and augment operational efficacy within resource-constrained environments (Hu et al., 2023; Kheradmandi et al., 2022; Lau et al., 2020). Prevailing techniques encompass model compression, network pruning, low-precision quantization, lightweight networks (Gong et al., 2024), and model distillation. These strategies can be utilized either independently or in concert to attain efficient object detection with limited resources.

Prior research frequently neglected the necessity of striking a balance between accuracy and speed. Attaining an apt equilibrium between accuracy and speed is of utmost significance. Hence, this paper puts forward an integrated methodology employing extant single-stage detectors.

2.2 | Knowledge distillation

Knowledge distillation enhances performance by transferring knowledge from a complex model (teacher) to a simplified model (student). Standard methods include response-based approaches (Gou et al., 2021), which use the teacher model's classification predictions as the "target knowledge" for the student model; feature-based approaches (Park et al., 2019), where the student learns from features extracted by the teacher; and relation-based approaches (Cho et al., 2019), which focus on the relationships between samples or feature layers. These methods allow the student model to benefit from the teacher model's expertise.

Response-based knowledge distillation encounters challenges related to incomplete knowledge transfer from teacher to student models (Xin Wang et al., 2024; C. Yang et al., 2023). This inadequacy can result in suboptimal performance in the separation of foreground and background in crack images. To address this issue, the current study integrates relation and feature knowledge distillation, with a focus on decoupling instance-related foreground from instance-unrelated background in crack images.

2.3 | Crack measurement

Crack measurement is a core issue in precise condition assessment. The aim is to measure crack parameters based on extracted segmentation results and utilize these measurements to guide comprehensive evaluation and maintenance (Liu et al., 2024). Common crack measurement methods are as follows: The orthogonal projection method (Nishikawa et al., 2011), which calculates the distance between two edges along a line parallel to the normal vector of the crack's central axis. The crack direction is defined by the tangent vector of the central axis. The shortest distance method is combined with the orthogonal projection method (Ong et al., 2022), which determines crack width by identifying a pair of points in near orthogonal directions that yield the shortest distance. The longitudinal section method (Y. Zhang et al., 2018) easily extracts the contour of the crack area by iterating over all edges. These methods can be applied individually or in combination to achieve efficient path tracking in resource-constrained environments.

Despite the proposed improvements intended to enhance the performance of measurement, challenges still exist, including high computational complexity, significant errors in crack edge, and skeleton line algorithms. To address these challenges, the PCM framework is proposed to enable online measurement of pavement cracks.

3 | METHODOLOGY

This section offers a comprehensive overview of the PCM framework and provides an in-depth explanation of each module (Figure 1).

3.1 | Lightweight crack segmentation

A novel real-time crack measurement framework, namely, PCM, is introduced in this paper, as depicted in Figure 1. This framework encompasses two essential constituents: a lightweight distillation network and a surface feature measurement algorithm. Specifically, the IHD integrates relation-based and feature-based knowledge distillation, capitalizing on crack instance-related semantic information to facilitate efficient knowledge transfer between the teacher and the student. The student model

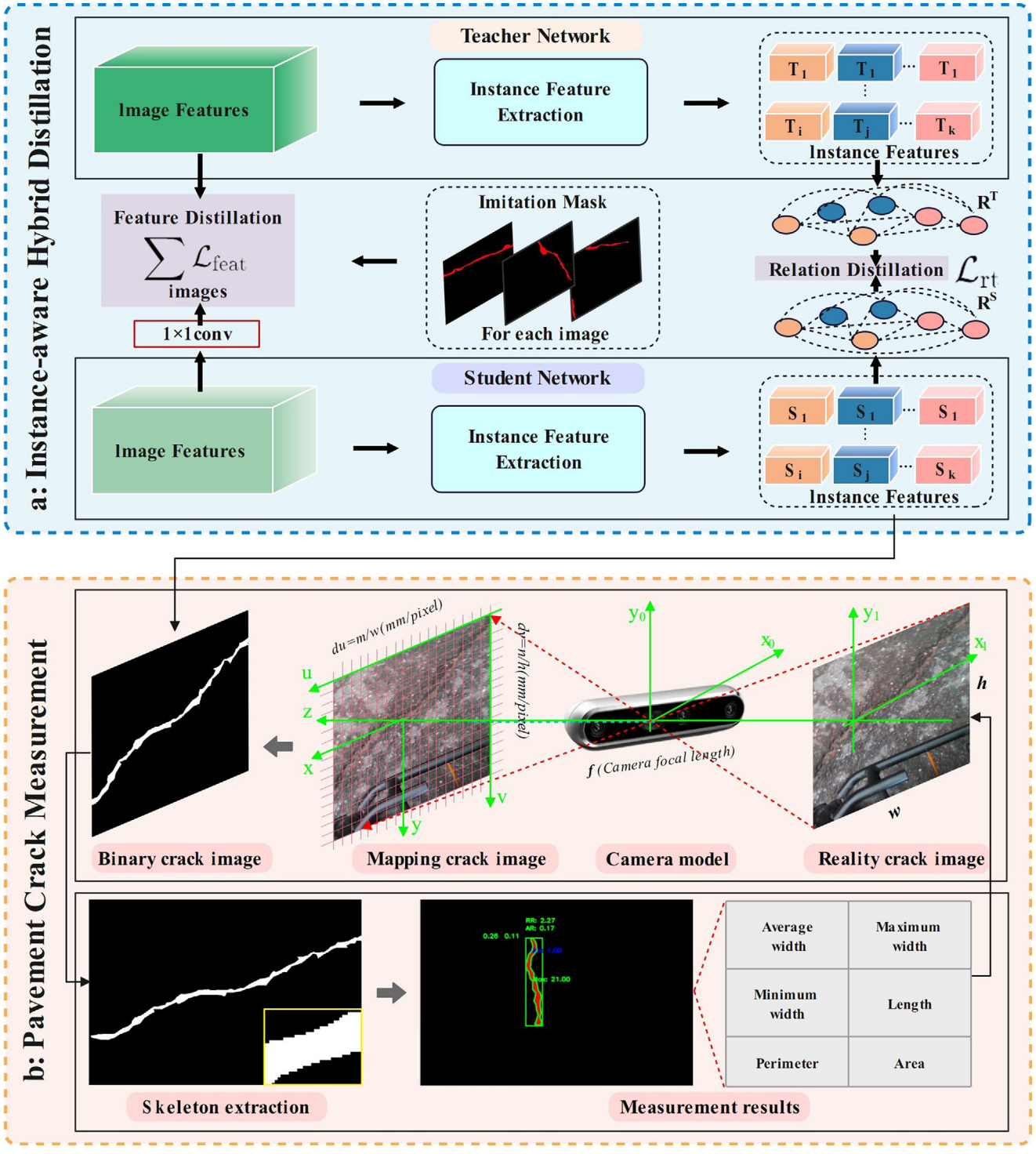

FIGURE 1 Frame work of real-time pavement crack measurement (PCM), which consists of instance-aware hybrid distillation (IHD) and PCM.

comprehensively emulates the features and relationships of the teacher to attain a more precise and lightweight segmentation model. Additionally, a robotic online crack surface feature measurement algorithm is devised, which exploits distance mapping relationships and edge coordinate extraction predicated on robot camera calibration.

3.2 | Lighter model for crack segmentation

In the current section, a more lightweight crack segmentation model is introduced. Subsequently, an elaborate and detailed account of the feature-based knowledge

distillation process is provided. This encompasses a comprehensive and in-depth exploration of the various aspects and mechanisms involved in this particular distillation technique. Additionally, extensive and thorough information regarding the relational knowledge distillation is furnished. Herein, a direct yet highly efficacious approach for the establishment of instance-level relationships is put forward. This approach is carefully designed and refined to ensure its effectiveness and relevance in the context of the overall knowledge distillation framework. Finally, the entirety of the loss function, which plays a crucial role in the optimization and performance of the model, is introduced.

3.2.1 | IHD

In the present analysis, the high-level semantic features $F \in \mathbb{R}^{C\times H\times W}$ outputted by the final layer of the backbone network are taken into consideration. Here, k is used to signify the instances. The vector of the kth instance representation is derived by computing the average of the feature vectors correlated with its corresponding region of interest (RoI). For the facilitation of this process, the notation (h, w) is adopted to denote the spatial domain $H\times W$, and a binary mask $M^k \in \{0,1\}^{H\times W}$ is applied to activate the RoI of the kth instance. Thus, F_I^k is expressed as

$$F_{\mathrm{I}}^k = \frac{1}{N^k}\sum_{h=1}^{H}\sum_{w=1}^{W}\mathbf{M}^k(h,w)\cdot\mathbf{F}(h,w)$$
$$N^k = \sum_{h=1}^{H}\sum_{w=1}^{W}\mathbf{M}^k(h,w) \quad (1)$$

Here, N^k signifies the count of feature vectors.

By averaging the vector representations of all instances of a class that appears, a prototype for each occurrence of class $F_{\mathrm{P}} \in \mathbb{R}^C$ is obtained from each image. Inspired by weakly supervised semantic segmentation, the mimicking mask $\mathbf{M_{IM}} \in \mathbb{R}^{H\times W}$ is achieved through the assessment of the similarity between class prototypes and the pixel embeddings within the original feature F. To be specific, the spatial dimensions of F are first flattened, transforming F from $C\times H\times W$ to $H\times W\times C$. Thereafter, matrix multiplication is conducted between each class prototype F_{P} and F, giving rise to a $H\times W$ similarity map. By traversing all the classes that emerge and integrating their similarity maps via element-wise addition, a $H\times W$ unnormalized mask is attained. This process bears resemblance to the implementation of an "OR" operation on the generated similarity maps. Subsequently, the mask is subjected to a simple min-max normalization, leading to the generation of the final mimic mask $\mathbf{M_{IM}} \in [0,1]^{H\times W}$. The values within the mimic mask correspond to the probabilities of pixel embeddings being considered as instance-related foreground, which ought to be given priority by the student during the knowledge distillation process. The relevant pseudocode process is shown in Algorithm 1.

ALGORITHM 1 Generation of mimicking mask $\mathrm{M_{IM}}$.

1: Input: Feature matrix $F \in \mathbb{R}^{C\times H\times W}$, class prototypes FP
2: Output: Mimicking mask $\mathrm{M_{IM}} \in [0,1]^{\mathrm{H\times W}}$
3: Flatten the spatial dimensions of F:
4: F ← reshape (F, (H × W, C)) ▷ Transform F from $\mathbb{R}^{C\times H\times W}$ to $\mathbb{R}^{H\times W\times C}$
5: **for all** each class prototype FP_i **do**
6: $\mathrm{similarity\_map}_i$ ← matrix_multiplication (FP_i, F) ▷ Get a H × W similarity map for each class
7: **end for**
8: unnormalized_mask ← zeros (H, W)
9: **for all** generated similarity maps similarity_map*i* **do**
10: unnormalized_mask ← unnormalized_mask + $\mathrm{similarity\_map}_i$ ▷ Element-wise addition
11: **end for**
12: $\mathrm{M_{IM}}$ ← min_max_normalize (unnormalized_mask) ▷ Simple min-max normalization to get $\mathrm{M_{IM}} \in [0,1]^{\mathrm{H\times W}}$

For the purpose of clearly segregating pixel embeddings into instance-oriented foreground, the $\varphi \in [0,1]$ is introduced to divide the $\mathbf{M_{IM}}$ into foreground representation ($\mathbf{M_{FG}}$) and background representation ($\mathbf{M_{BG}}$). Precisely, in the case of $\varphi = 0$, all pixel embeddings are treated as foreground; conversely, when $\varphi = 1$, all are considered as background. The $\mathbf{M_{FG}}$ and $\mathbf{M_{BG}}$ are then applied to the multi-scale features generated by a feature pyramid network (FPN). To accomplish this, the mask is resized via interpolation to match the target feature scale.

In an effort to empower students to learn and replicate the behavior of the teacher model predicated on the output features of the FPN, an additional layer is integrated subsequent to each output of the student model, with the objective of ensuring the congruence of feature dimensions. As has been emphasized by G. Chen et al. (2017), this adaptation layer holds significant importance even in scenarios where the features of the teacher and the student exhibit identical dimensionality. The teacher and student are respectively denoted as $\mathbf{F}_{\mathrm{FPN}}^{\mathrm{T}}$ and $\mathbf{F}_{\mathrm{FPN}}^{\mathrm{S}}$. The feature $\mathcal{L}_{\mathrm{feat}}$ loss is expounded as follows:

$$\mathcal{L}_{\mathrm{feat}} = \mathcal{L}_{\mathrm{mse}}\left(\mathbf{F}_{\mathrm{FPN}}^{\mathrm{T}}, \sigma\left(\mathbf{F}_{\mathrm{FPN}}^{\mathrm{S}}\right)\right)\cdot\left(\mathbf{M}_{\mathrm{FG}} + \tau\cdot\mathbf{M}_{\mathrm{BG}}\right) \quad (2)$$

where σ denotes the adaptation layer, $\mathcal{L}_{\mathrm{mse}}$ indicates the mean squared error loss. τ is a constant that balances the importance of foreground and background distillation. In our experiments, τ is set within the range of 2–4.

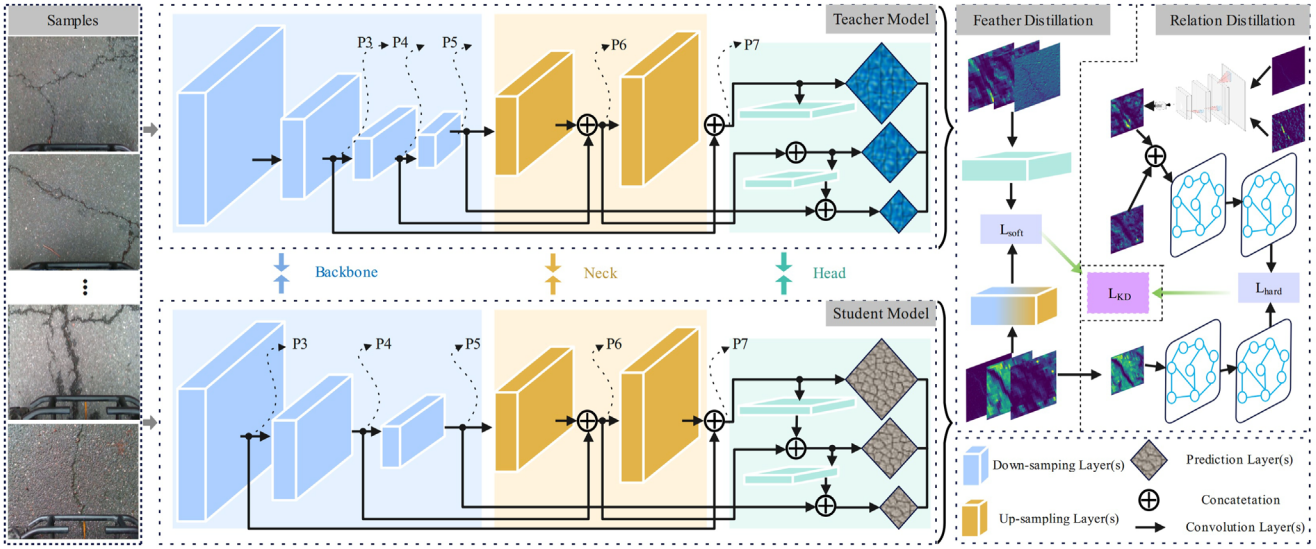

FIGURE 2 Architecture of IHD. The symbols P3–P7 represent the levels of the feature pyramid in YOLOv8. P3 denotes the high-resolution feature map output from the backbone network. P4, P5, P6, and P7 indicate feature maps with progressively lower resolutions, achieved through downsampling operations.

In conjunction with the previously described feature-knowledge distillation, relational knowledge distillation is instituted by scrutinizing the relationships among diverse instances. This methodology endeavors to augment intra-class compactness and inter-class discriminability. It is hypothesized that an efficacious teacher model frequently discloses distinctive correlations both within and between classes. Extending the domain from solitary images to small batches, presume that a batch encompasses K instances, with K differing across various batches. Let i and j serve as indices for the vectors denoting these K instances. The instance-level relationship $R_{i,j}$ is quantified via cosine similarity in the following manner:

$$\mathbf{R}_{i,j} = \frac{F_I^i \cdot F_I^j}{\| F_I^i \|_2 \cdot \| F_I^j \|_2} \tag{3}$$

In the context where the instance representation set $\mathbf{F_I^T} \in \mathbb{R}^{K \times C}$ originating from the teacher and the instance representation set $\mathbf{F_I^S} \in \mathbb{R}^{K \times C'}$ stemming from the student are considered, with C and C' denoting the number of channels which might not be identical, the instance-level relationships $\mathbf{R^T}$ pertaining to the teacher and $\mathbf{R^s}$ associated with the student are separately computed based on Equation (4). Subsequently, the relational loss $\mathcal{L}_{\text{rt}}$ is represented as follows:

$$\mathcal{L}_{\text{rt}} = \frac{1}{K} \| \mathbf{R}^{\text{T}} - \mathbf{R}^{\text{S}} \|_{\text{F}}^2 \tag{4}$$

where the Frobenius norm is denoted by $\| * \|_{\text{F}}$.

Figure 2 depicts the utilization of YOLOv8 (Jocher et al., 2023), a more profound pre-trained model, for the precise representation of sample data features, which are regarded as knowledge from the original model. The final detection model is trained by minimizing the output disparities between the original and smaller models. As illustrated in Figure 2, the upper network represents the teacher network, whereas the bottom network represents the student network. For the transfer of knowledge from the teacher to the student network, the P3–P7 layers extract abundant feature information from the backbone and neck. This process fully exploits instance-related information.

3.2.2 | Loss function

Our proposed method's loss function comprises three components: loss of classification ($\mathcal{L}_{cls}$), loss of confidence ($\mathcal{L}_{obj}$), and loss of bounding box regression for object detection ($\mathcal{L}_{bbox}$), along with segmentation mask loss ($\mathcal{L}_{mask}$), similar to the loss function used in YOLOv8. Additionally, a distillation loss ($\mathcal{L}_{KD}$) is incorporated to evaluate the distillation difference between the teacher model and the student model.

The classification loss ($\mathcal{L}_{cls}$) is specified as

$$\mathcal{L}_{cls} = \mathcal{L}_{BCE} = \begin{cases} -\log p, & \text{y} = 1 \\ -\log(1-p), & \text{y} = 0 \end{cases} \tag{5}$$

where $\mathcal{L}_{BCE}$ represents the binary cross-entropy loss, which is widely used in binary classification problems. 1 indicates a positive sample, and 0 indicates a negative sample. Additionally, the predicted probability assigned by the algorithm for classifying the input as positive is denoted as p.

The loss of confidence ($\mathcal{L}_{obj}$) is calculated using a binary cross-entropy function and incorporates weight coefficients that change according to the size of the detection layers. These coefficients are allocated in decreasing order according to the sizes of the detection layers, with default values established at 4.0, 1.0, and 0.4. Consequently, the loss for small objects on multi-scale feature maps receives higher weight coefficients.

$$\mathcal{L}_{obj} = \mathcal{L}_{cls} \tag{6}$$

$$\begin{aligned} \mathcal{L}_{bbox} &= \mathcal{L}_{CIoU(B,B_{gt})} = IoU(B,B_{gt}) - \frac{\rho^2(B,B_{gt})}{c^2} - \alpha\upsilon \\ \upsilon &= \frac{4}{\pi}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \\ \alpha &= \frac{\upsilon}{1 - IoU(B,B_{gt}) + \upsilon} \end{aligned} \tag{7}$$

where the parameter υ denotes the normalized variation. The parameter a ranges from 0 to $\pi/4$ and is then scaled by $\pi/4$ to ensure it falls within the 0 to 1 range. Additionally, the parameter α is a balancing factor that adjusts the loss contribution from aspect ratio differences relative to the loss generated by the intersection over union (IoU).

The distillation loss ($\mathcal{L}_{KD}$) is defined as the weighted sum of two components as shown below:

$$\mathcal{L}_{KD} = \mathcal{L}_{\text{det}} + \alpha\mathcal{L}_{\text{feat}} + \beta\mathcal{L}_{\text{rt}} \tag{8}$$

where the two hyperparameters, α and β, balance the distillation loss and detection loss. The overall loss is model-independent.

The total loss ($\mathcal{L}_{Total}$) should be represented as follows.

$$\mathcal{L}_{Total} = \mathcal{L}_{cls} + \mathcal{L}_{obj} + \mathcal{L}_{bbox} + \mathcal{L}_{KD} \tag{9}$$

3.3 | Pavement crack surface feature measurement

Robot cameras can be installed on diverse types of robotic platforms and exhibit flexible mobility. They are able to move freely in complex road environments, adapting to a wide variety of detection scenarios. Through analyzing and processing images, they can precisely measure parameters such as the length, width, and area of cracks, offering quantitative metrics for crack detection. A real-time crack surface feature measurement algorithm was developed, based on distance mapping relations and crack edge coordinate extraction under robot camera calibration. This approach effectively alleviates issues related to edge branching and loss during parameter measurement, enhancing the efficiency of online crack measurement. As depicted in Figure 3, the algorithm uses robot vision camera calibration to establish a transformation between the crack image coordinate system and the real-world crack coordinate system of the road scene. It incorporates a crack surface feature skeleton extraction and measurement method, enabling accurate measurement of crack parameters.

3.3.1 | Robotic camera projection geometry

The mapping of points from a scenario in an image is represented through a sequence of geometric transformations as demonstrated in Figure 3a. Typically, when accounting for lens distortion, this process can be decomposed into four distinct transformations.

The initial transformation consists of a rigid body transformation that maps world coordinates from the scenario to the coordinates of the camera. This transformation incorporates three rotations (α, β, γ) along with three translations (t_x, t_y, t_z). These values are generally known as the extrinsic parameters of the camera. The transformation is usually depicted as a 4×4 matrix. The $P^w = (x^w, y^w, z^w)^T$ and $P^c = (x^c, y^c, z^c)^T$ present the coordinates in the world and camera coordinate systems, respectively.

The second transformation is based on the pinhole camera projecting through a perspective projection. The third involves a 2D affine mapping that converts coordinates from the image plane into pixel coordinates.

The last transformation is nonlinear and is intended to address optical distortion. When higher accuracy is not critically necessary, it can sometimes be disregarded. This transformation is typically represented as Equation (14), where $a_c = (c, r)^T$ denotes the correct coordinates, $a_d = (c', r')^T$ represents the distorted coordinates, $\mathcal{F}$ represents the nonlinear distortion function, while δ characterizes the distortion.

$$a_c = a_d + \mathcal{F}(a_d, \delta) \tag{10}$$

These models that simulate optical distortions generally encompass three distortions: radial, tangential, and prismatic distortion. Distortion can also be viewed as a transformation from correct coordinates to distorted coordinates. The intrinsic parameters of the camera are determined entirely by the camera's internal design and are not influenced by its location or orientation.

The process of converting world coordinates into pixel coordinates is referred to as forward projection. The inverse transformation, referred to as backward projection, is intrinsically ambiguous because each pixel aligns with a projection line in world coordinates, allowing numerous points in the world to potentially map to the same pixel.

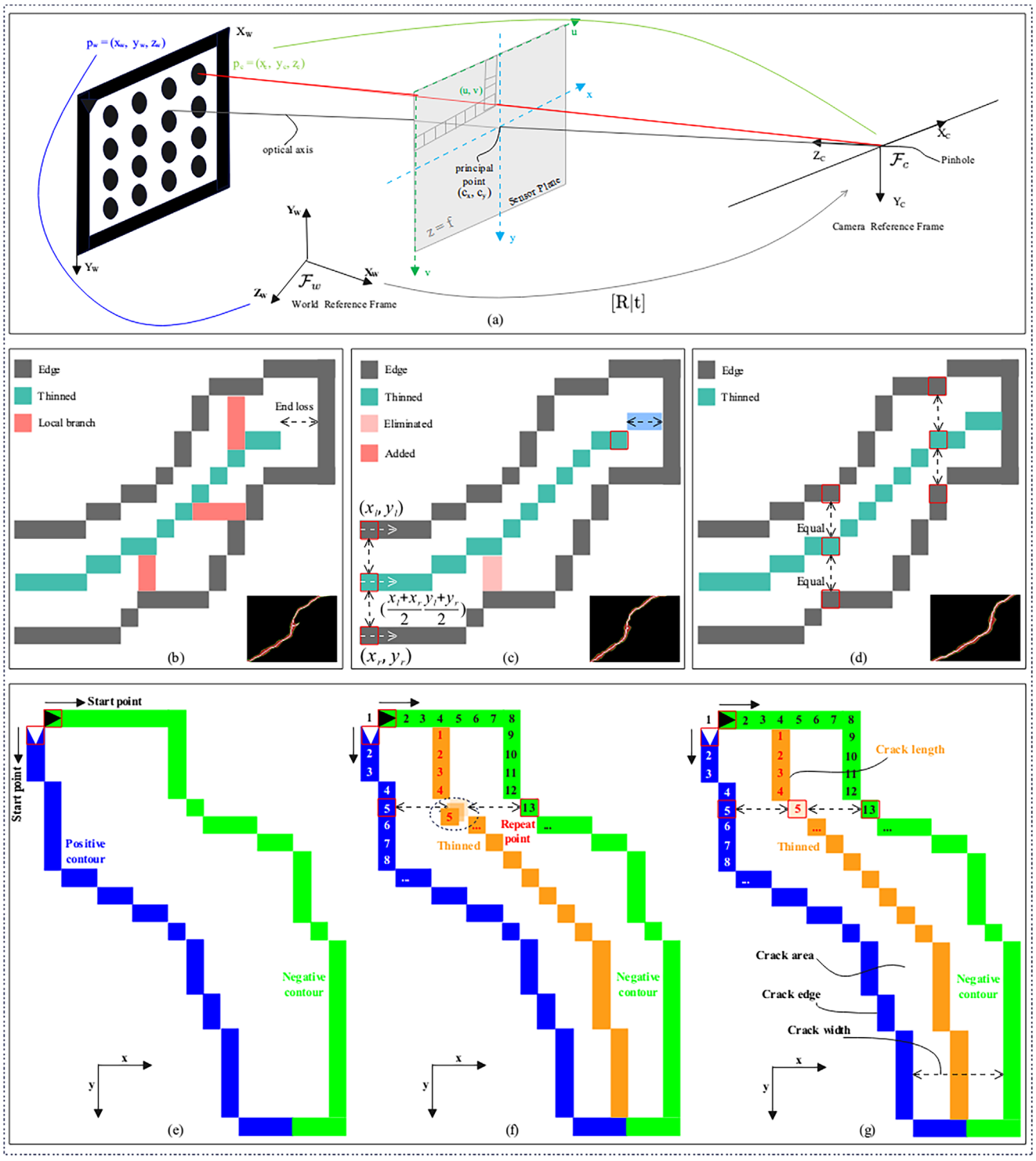

FIGURE 3 Schematic diagram of the proposed real-time crack surface feature measurement algorithm: (a) camera projection model, (b) traditional method suffers from local branch and feature end loss during thinning, (c) overview of the proposed method flow, and (d) skeleton extraction results of the proposed method; (e) extract negative contour, positive contour; (f) obtain midpoint; (g) results of skeleton extraction.

3.3.2 | Crack skeleton extraction and measurement algorithm

Through computational analysis, it was found that traditional parallel thinning algorithms encounter issues of localized branching and pixel loss at crack ends during the extraction of crack center pixels as illustrated in Figure 3b. This can clearly lead to inaccurate crack length calculations. To address this, a new crack measurement algorithm was developed, taking into full account the morphological

characteristics of cracks, as seen in Figure 3c. Through obtaining crack boundaries and calculating their average, the issues are effectively resolved, with the final results displayed in Figure 3d.

Building on the crack segmentation results presented in Section 3.2, the extracted crack contours are categorized into optimal contours and negative contours. The optimal contour for each region is identified by selecting the contour that encloses the biggest count of internal pixels.

$$S = \sum_{x=1} \sum_{y=1} \Gamma(x, y) \tag{11}$$

where S refers to the count of pixels within the object region, and $\Gamma(x, y)$ denotes the gray value associated with the object. Utilizing the extracted (x, y), a set of contour points, denoted as $\mathbb{Z}$, is formed. Note that the coordinate system's origin is established at the top-left corner of the image. The values "1" and "0" correspond to the target and the background, respectively.

The pixels within set $\mathbb{Z}$ were organized in a counterclockwise order to create the complete shape set $\mathbb{Q}$. Its initial segment of $\mathbb{Q}$ is identified as the positive shape set $\mathbb{Q}_{\text{pos}}$ (Figure 3e). In the case of longitudinal pavement crack, the initial point is at the top-left quadrant of the contour line, while for transverse pavement crack, the starting point is the lower left corner (cracks are distinguished by their rotation angle R, with $R \geq 45°$ indicating longitudinal cracks and $R < 45°$ indicating transverse cracks). Negative Contour: Similarly, pixels in $\mathbb{Z}$ are arranged in a clockwise manner to create the complete shape set $\mathbb{N}$. Its initial segment of $\mathbb{N}$ is designated by the negative shape set $\mathbb{N}_{\text{neg}}$. Central Pixel: To every shape, all points are scanned through in both $\mathbb{Q}_{\text{pos}}$ and $\mathbb{N}_{\text{neg}}$. Pairs of points are located that have matching values and compute the centroid coordinates for these pairs.

$$\begin{cases} X^c = x_a + \frac{|x_b - x_a|}{2} \\ Y^c = y_a \,(\text{if } y_a = y_b) \end{cases} \tag{12}$$

where X^c and Y^c denote the coordinates of a pixel in $\mathbb{C}$; x_a, y_a and x_b, y_b represent the coordinates in $\mathbb{Q}_{\text{pos}}$ and $\mathbb{N}_{\text{neg}}$, respectively.

Owing to the irregular nature of crack contours, several centroids can be detected at points where edges change sharply. For example, in Figure 3f, the no. 7, no. 8, and no. 9 edge points produce multiple centroids. To resolve this issue, the point with the lowest length to its nearby points is preserved, but the others are eliminated. The method produces the ultimate collection of centroids, referred to as β. The crack length *Leng* is then determined by calculating and summing the distances between each pair of adjacent centroids in β.

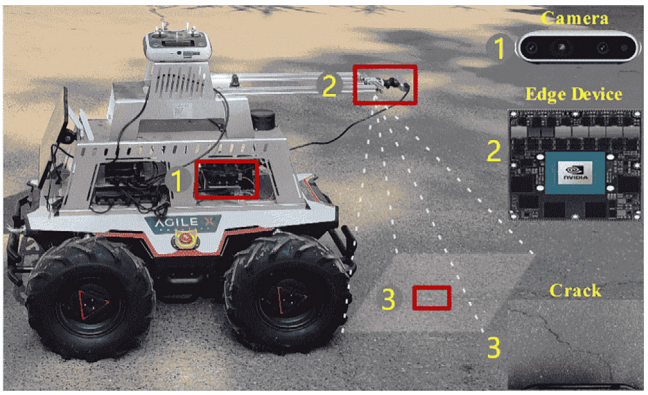

FIGURE 4 Working conditions of our proposed autonomous mobile robot.

TABLE 1 Main components of our equipment.

Main Experiments	Parameters
Edge computing device	Nvidia Jetson AGX Xavier (32G)
RGB-D camera	Intel RealSence D435i
Communication device	HUAWEI wireless router
Communication interface	Standard CAN bus
Vehicular power supply	28 V 30 Ah
DC motor	4*200 w Brushless servo
Driving mode	Four-wheeled differential drive
Motor driver	Agilex
Size	980*730*340 mm
Total weight	67.62 kg

$$Leng = \sum_{i=1} \sqrt{\left(X^c_{i+1} - X^c_i\right)^2 + \left(Y^c_{i+1} - Y^c_i\right)^2} \tag{13}$$

where X^c_{i+1}, Y^c_{i+1}, X^c_i, and Y^c_i represent the pixel coordinates within the center point set β. The crack width W is calculated using the following equation:

$$W = \text{Min}\,(\tau) \times 2 + 1 \tag{14}$$

where τ denotes the length from the crack edge to the relevant central pixel.

4 | EXPERIMENTS

In this section, the crack segmentation findings for S2TCrack are analyzed and discussed. Then, the algorithm is implemented on an edge equipment for measurement.

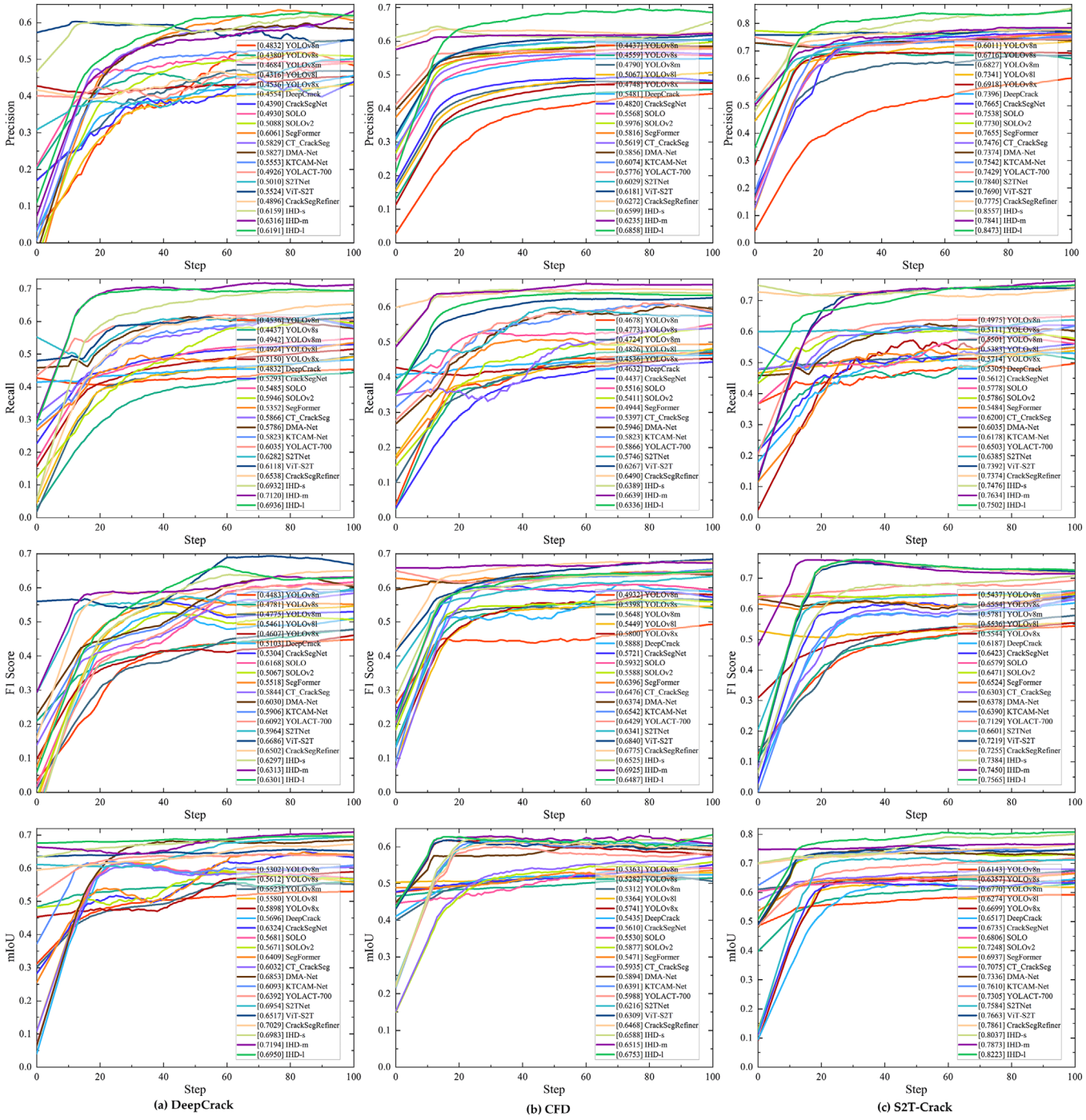

FIGURE 5 The comprehensive comparison of the segmentation training curves across the DeepCrack, CFD, and S2T-Crack datasets is presented.

4.1 | Experimental setting

The segmentation model training experiments were performed on a GPU server equipped with an RTX 4090 GPU. The IHD was evaluated and confirmed on: DeepCrack (Zou et al., 2019), CFD (Shi et al., 2016), and S2TCrack (Zhang, Yang, Wang, Brilakis, Davletshina, Wang & Cao, 2024). These datasets were selected for typical asphalt pavement cracks.

Crack trajectory segmentation is a critical step before quantification. As shown in Figure 4, the autonomous mobile robot employs an RGB-D camera to gather visual data. The detailed parameters of the main components are shown in Table 1. The embedded system comprises

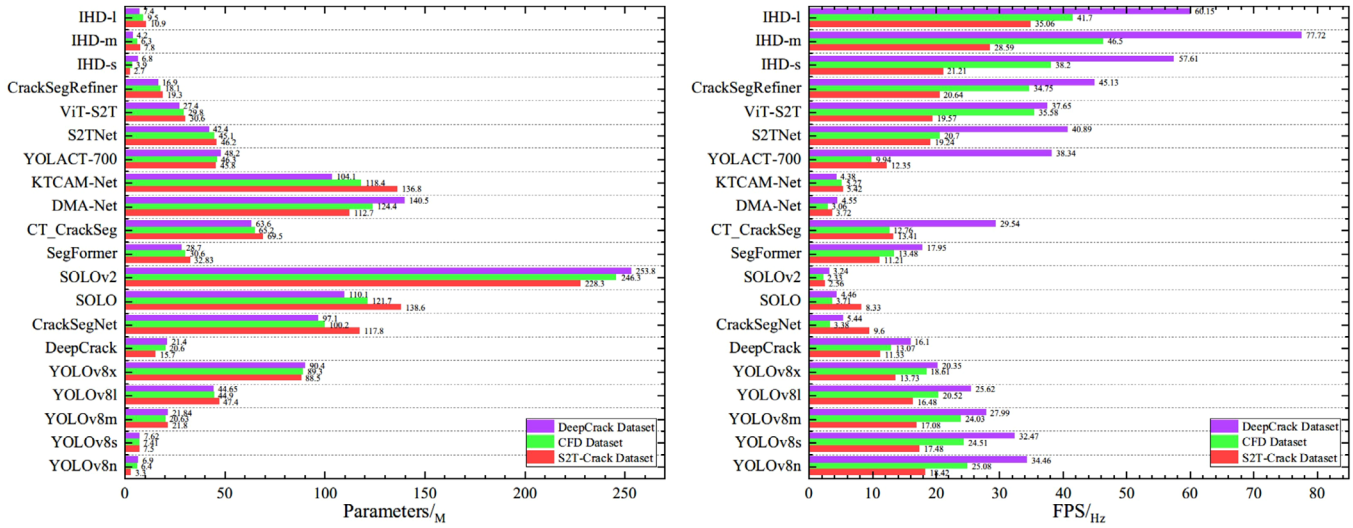

FIGURE 6 Statistical analysis of lightweight evaluation metrics for segmentation models. Different color blocks represent three datasets.

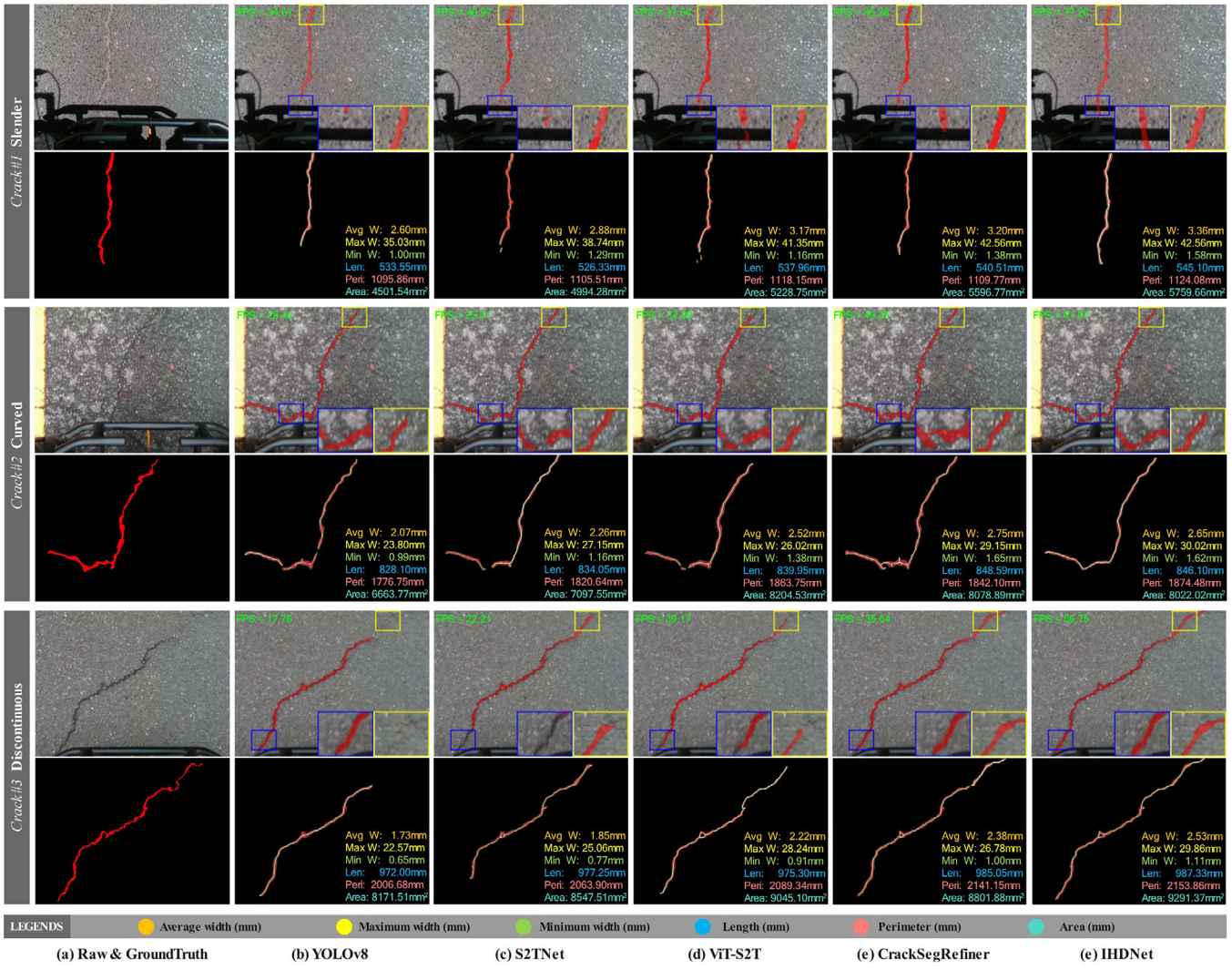

FIGURE 7 Visualization of real-time PCM. Robotic visual interface can display real-time frames per second and crack measurement parameters.

TABLE 2 Comparative evaluation of various methods with instance-aware hybrid distillation (IHD). The highest results are emphasized in red, whereas the second-highest results are shown in blue.

Method	DeepCrack				CFD				S2T-Crack			
	Precision	*Recall*	*F1 Score*	*mIoU*	*Precision*	*Recall*	*F1 Score*	*mIoU*	*Precision*	*Recall*	*F1 Score*	*mIoU*
YOLOv8n	0.4832	0.4437	0.4483	0.5302	0.4437	0.4678	0.4932	0.5363	0.6011	0.4975	0.5437	0.6143
YOLOv8s	0.4380	0.4536	0.4781	0.5612	0.4559	0.4773	0.5398	0.5282	0.6716	0.5111	0.5554	0.6357
YOLOv8m	0.4684	0.4437	0.4775	0.5523	0.4790	0.4724	0.5648	0.5312	0.6827	0.5501	0.5781	0.6770
YOLOv8l	0.4316	0.4942	0.5461	0.5580	0.5067	0.4826	0.5449	0.5364	0.7341	0.5383	0.5536	0.6274
YOLOv8x	0.4536	0.5150	0.4607	0.5898	0.4748	0.4536	0.5800	0.5741	0.6918	0.5714	0.5544	0.6699
DeepCrack	0.4554	0.4832	0.5103	0.5696	0.5481	0.4632	0.5888	0.5435	0.7396	0.5305	0.6187	0.6517
CrackSegNet	0.4390	0.5293	0.5304	0.6324	0.4820	0.4437	0.5721	0.5610	0.7665	0.5612	0.6423	0.6735
SOLO	0.4930	0.5485	0.6168	0.5681	0.5568	0.5516	0.5932	0.5530	0.7538	0.5778	0.6579	0.6806
SOLOv2	0.5088	0.5946	0.5067	0.5671	0.5976	0.5411	0.5588	0.5877	0.7730	0.5786	0.6471	0.7248
SegFormer	0.6061	0.5352	0.5518	0.6409	0.5816	0.4944	0.6396	0.5471	0.7655	0.5484	0.6524	0.6937
CT_CrackSeg	0.5829	0.5866	0.5844	0.6032	0.5619	0.5397	0.6476	0.5935	0.7476	0.6200	0.6303	0.7075
DMA-Net	0.5827	0.5786	0.6030	0.6853	0.5856	0.5946	0.6374	0.5894	0.7374	0.6035	0.6378	0.7336
KTCAM-Net	0.5553	0.5823	0.5906	0.6093	0.6074	0.5823	0.6542	0.6391	0.7542	0.6178	0.6390	0.7610
YOLACT-700	0.4926	0.6035	0.6092	0.6392	0.5776	0.5866	0.6429	0.5988	0.7429	0.6503	0.7129	0.7305
S2TNet	0.5010	0.6282	0.5964	0.6954	0.6029	0.5746	0.6341	0.6216	0.7840	0.6385	0.6601	0.7584
ViT-S2T	0.5524	0.6118	0.6686	0.6517	0.6181	0.6267	0.6840	0.6309	0.7690	0.7392	0.7219	0.7663
CrackSegRefiner	0.4896	0.6538	0.6502	0.6729	0.6272	0.6490	0.6775	0.6468	0.7775	0.7374	0.7255	0.7861
IHD-s	0.6159	0.6932	0.6297	0.6983	0.6599	0.6389	0.6525	0.6588	0.8557	0.7476	0.7384	0.8037
IHD-m	0.6316	0.7120	0.6313	0.7094	0.6235	0.6639	0.6798	0.6515	0.7841	0.7634	0.7450	0.7873
IHD-l	0.6191	0.6936	0.6301	0.6950	0.6858	0.6336	0.6487	0.6753	0.8473	0.7502	0.7565	0.8223

Ubuntu, ROS Melodic, PyTorch 1.8, and Jetpack 4.4. The edge device is the Nvidia Jetson AGX Xavier. This setup controls the robot's navigation and real-time quantification.

To assess the segmentation performance, evaluation metrics such as precision, recall, F1 score, and mean IoU (mIoU) are employed. Additionally, the number of parameters and frames per second (FPS) serve as benchmarks for inference speed. This study utilizes absolute and relative errors to assess the effectiveness and computational accuracy of the proposed method.

4.2 | Segmentation results

In the training phase, various commonly used metrics such as precision, recall, F1 score, and mIoU are comprehensively compared. This discussion helps clarify the compromises associated with various metrics and guides the selection of the most suitable evaluation criterion for particular segmentation. The primary tested algorithms comprise YOLO, DeepCrack (Zou et al., 2019), CrackSegNet (Ren et al., 2020), SOLO (Xinlong Wang et al., 2021), SegFormer (Xie et al., 2021), CT_CrackSeg (Tao et al., 2023), DMA-Net (Sun et al., 2022), KTCAM-Net (Al-Huda et al., 2023), YOLACT-700 (Bolya et al., 2019), S2TNet (Zhang, Yang, Wang, Brilakis, Davletshina, Wang & Cao, 2024), ViT-S2T (Zhang, Yang, Wang, Brilakis, Davletshina, & Wang, 2024), CrackSegRefiner (Zhang, Yang, Wang, Wang, et al., 2024), and IHD.

4.2.1 | DeepCrack dataset

Figure 5a offers a comprehensive overview of various training metrics evaluated on the DeepCrack dataset. All models underwent training for 100 epochs to derive these metrics. Table 2 details a comparison across different models. The findings reveal that ViT-S2T achieved the highest F1 Score of 0.6686, whereas our proposed IHD-m secured the top positions in precision, recall, and mIoU, with respective values of 0.6316, 0.7120, and 0.7194. As shown in Table 3, the IHD-s, notable for its minimal parameter count, significantly reduces parameters, compared to the YOLO series. Additionally, IHD-l achieved the best FPS value at 35.06.

4.2.2 | CFD dataset

Figure 5b illustrates the training curves. Table 2 offers the training metrics. CrackSegRefiner achieved the

TABLE 3 Comparative analysis of lightweight evaluation metrics for various methods. The top results are marked in red, while the second-best results are indicated in blue.

	DeepCrack		CFD		S2T-Crack	
Method	***Params (M)***	***Frames per second (FPS; Hz)***	***Params (M)***	***FPS (Hz)***	***Params (M)***	***FPS (Hz)***
YOLOv8n	3.3	18.42	6.4	25.08	6.9	34.46
YOLOv8s	7.5	17.48	7.4	24.51	7.6	32.47
YOLOv8m	21.8	17.08	20.6	24.03	21.8	27.99
YOLOv8l	47.4	16.48	44.9	20.52	44.7	25.62
YOLOv8x	88.5	13.73	89.3	18.61	90.4	20.35
DeepCrack	15.7	11.33	20.6	13.07	21.4	16.10
CrackSegNet	117.8	9.60	100.2	3.38	97.1	5.44
SOLO	138.6	8.33	121.7	3.71	110.1	4.46
SOLOv2	228.3	2.56	246.3	2.33	253.8	3.24
SegFormer	32.8	11.21	30.6	13.48	28.7	17.95
CT_CrackSeg	69.5	13.41	65.2	12.76	63.6	29.54
DMA-Net	112.7	3.72	124.4	3.06	140.5	4.55
KTCAM-Net	136.8	5.42	118.4	5.27	104.1	4.38
YOLACT-700	45.8	12.35	46.3	9.94	48.2	38.34
S2TNet	46.2	19.24	45.1	20.70	42.4	40.89
ViT-S2T	30.6	19.57	29.8	35.58	27.4	37.65
CrackSegRefiner	19.3	20.64	18.1	34.75	16.9	45.13
IHD-s	2.7	21.21	3.9	38.20	6.8	57.61
IHD-m	7.8	28.59	6.3	46.50	4.2	77.72
IHD-l	10.9	35.06	9.5	41.70	7.4	60.15

second-highest recall rate at 0.6686. IHD-m excelled in both recall rate and F1 score, with 0.6639 and 0.6925, respectively. Additionally, IHD-l showed the best performance in precision and mIoU, significantly improving mIoU by 13.9%, compared to the baseline YOLOv8. From Table 3, IHD-s effectively reduced the number of parameters to 3.9, while IHD-m demonstrated computational efficiency with an FPS of 46.5.

4.2.3 | S2T-Crack dataset

Figure 5c displays the training curves. Table 2 offers the training metrics for various methods. The IHD series shows the best performance for accuracy, recall, F1 score, and mIoU, achieving 0.8557, 0.7634, 0.7565, and 0.8223, respectively. In comparison to the baseline model, YOLOv8, the IHD exhibits an enhancement of more than 13 percentage points across accuracy, recall, F1 score, and mIoU. As shown in Table 3, IHD-m achieves a frame rate of 77.72 with a parameter count of 4.2 M, effectively reducing model parameters meanwhile having good results. In summary, compared to other state of the art (SOTA) methods, IHD offers the most significant accuracy-efficiency balance, presenting the efficacy of the proposed lightweight knowledge distillation. Figure 6 presents a qualitative analysis of lightweight evaluation metrics, illustrating the parameter counts and FPS of the model across three datasets. This visualization effectively depicts the distribution of metrics within the datasets and informs future research on lightweight methodologies.

4.3 | Crack measurement results

This section presents an experimental study of the designed real-time measurement system utilizing a self-developed robot under actual road conditions. The tests aimed to determine the average width, maximum width, minimum width, length, perimeter, and area of the cracks, all measured in millimeters.

The measurement results for the slender-shaped asphalt pavement crack (Crack#1), executed on the self-developed mobile robot, are shown in Figure 7. Table 4 presents the measurement results of six crack measurement parameters under three typical asphalt pavement conditions using five different segmentation models. Among them, the manual refers to the actual measured values obtained manually, while ground truth (GT) refers to the computed values of the manually annotated crack images. When using the

TABLE 4 Comparative assessment of crack measurement performance utilizing the proposed method on five segmentation models with three pavement crack conditions.

		Crack#1 Slender					*Crack#2* Curved					*Crack#3* Discontinuous				
Parameters	**Metric**	**YOLOv8**	**S2TNet**	**ViT-S2T**	**CrackSeg-Refiner**	**IHDNet**	**YOLOv8**	**S2TNet**	**ViT-S2T**	**CrackSeg-Refiner**	**IHDNet**	**YOLOv8**	**S2TNet**	**ViT-S2T**	**CrackSeg-Refiner**	**IHDNet**
Average width (mm)	Ground truth	3.59	3.59	3.59	3.59	3.59	3.01	3.01	3.01	3.01	3.01	2.91	2.91	2.91	2.91	2.91
	Prediction	2.60	2.88	3.17	3.20	3.36	2.07	2.26	2.52	2.75	2.65	1.73	1.85	2.22	2.38	2.53
	Absolute error	0.99	0.70	0.42	0.39	**0.23**	0.93	0.74	0.49	**0.26**	0.35	1.18	1.06	0.69	0.53	**0.38**
	Relative error (%)	27.55	19.62	11.7	10.94	**6.42**	31.37	24.86	16.24	**8.48**	11.56	40.65	36.47	23.18	18.52	**12.94**
Maximum width (mm)	Manual	44.15	44.15	44.15	44.15	44.15	34.35	34.35	34.35	34.35	34.35	31.49	31.49	31.49	31.49	31.49
	Ground truth	46.04	46.04	46.04	46.04	46.04	32.55	32.55	32.55	32.55	32.55	34.47	34.47	34.47	34.47	34.47
	Prediction	35.03	38.74	41.35	43.30	42.56	23.80	27.15	26.02	29.15	30.02	22.57	25.06	28.24	26.78	29.86
	Absolute error	11.01	7.30	4.68	**2.74**	3.48	8.75	5.40	6.53	3.40	**2.53**	11.90	9.41	6.23	7.69	**4.62**
	Relative error (%)	23.91	15.85	10.18	**5.94**	7.56	26.89	16.61	20.04	10.45	**7.78**	34.56	27.28	18.07	22.31	**13.39**
Minimum width (mm)	Manual	1.60	1.60	1.60	1.60	1.60	1.72	1.72	1.72	1.72	1.72	1.44	1.44	1.44	1.44	1.44
	Ground truth	1.62	1.62	1.62	1.62	1.62	1.83	1.83	1.83	1.83	1.83	1.35	1.35	1.35	1.35	1.35
	Prediction	1.00	1.29	1.16	1.38	1.58	0.99	1.16	1.38	1.65	1.62	0.65	0.77	0.91	1.00	1.11
	Absolute error	0.62	0.34	0.46	0.24	**0.04**	0.84	0.66	0.45	**0.18**	0.20	0.70	0.58	0.45	0.35	**0.24**
	Relative error (%)	38.33	20.83	28.33	15	**2.5**	45.61	36.73	24.57	**9.86**	11.32	51.68	43.24	32.94	25.54	**17.57**
Length (mm)	Manual	551.33	551.33	551.33	551.33	551.33	865.53	865.53	865.53	865.53	865.53	991.23	991.23	991.23	991.23	991.23
	Ground truth	554.42	554.42	554.42	554.42	554.42	854.06	854.06	854.06	854.06	854.06	1002.18	1002.18	1002.18	1002.18	1002.18
	Prediction	533.55	526.33	537.96	540.51	545.10	828.10	834.05	839.95	848.59	846.10	972.00	977.25	975.30	985.05	987.33
	Absolute error	20.87	28.09	16.46	13.91	**9.32**	25.96	21.37	14.11	**5.47**	7.97	30.18	24.93	26.88	17.13	**14.85**
	Relative error (%)	3.76	5.07	2.97	2.51	**1.68**	3.04	2.5	1.65	**0.64**	0.93	3.01	2.49	2.68	1.71	**1.48**

(Continues)

TABLE 4 (Continued)

		Crack#1 Slender					*Crack#2* Curved					*Crack#3* Discontinuous				
Parameters	**Metric**	**YOLOv8**	**S2TNet**	**ViT-S2T**	**CrackSeg-Refiner**	**IHDNet**	**YOLOv8**	**S2TNet**	**ViT-S2T**	**CrackSeg-Refiner**	**IHDNet**	**YOLOv8**	**S2TNet**	**ViT-S2T**	**CrackSeg-Refiner**	**IHDNet**
Perimeter (mm)	Manual	1132.50	1132.50	1132.50	1132.50	1132.50	1919.88	1919.88	1919.88	1919.88	1919.88	2272.38	2272.38	2272.38	2272.38	2272.38
	Ground truth	1142.41	1142.41	1142.41	1142.41	1142.41	1986.52	1986.52	1986.52	1986.52	1986.52	2354.71	2354.71	2354.71	2354.71	2354.71
	Prediction	1095.86	1105.51	1118.15	1109.77	1124.08	1776.75	1820.64	1863.75	1842.10	1874.48	2006.68	2063.90	2089.34	2141.15	2153.86
	Absolute error	46.55	36.90	24.26	32.64	**18.33**	209.78	165.88	122.77	144.42	**112.04**	348.03	290.81	265.37	213.57	**200.85**
	Relative error (%)	4.07	3.23	2.12	2.86	**1.6**	10.56	8.35	6.18	7.27	**5.64**	14.78	12.35	11.27	9.07	**8.53**
Area (mm)	Ground truth	6815.35	6815.35	6815.35	6815.35	6815.35	9171.16	9171.16	9171.16	9171.16	9171.16	10134.57	10134.57	10134.57	10134.57	10134.57
	Prediction	4501.54	4994.28	5228.75	5596.77	5759.66	6663.77	7097.55	8204.53	8078.89	8022.02	8171.51	8547.51	9045.10	8801.88	9291.37
	Absolute error	2313.81	1821.07	1586.61	1218.59	**1055.70**	2507.39	2073.61	966.63	**1092.27**	1149.14	1963.06	1587.06	1089.47	1332.69	**843.20**
	Relative error (%)	33.95	26.72	23.28	17.88	**15.49**	27.34	22.61	10.54	**11.91**	12.53	19.37	15.66	10.75	13.15	**8.32**

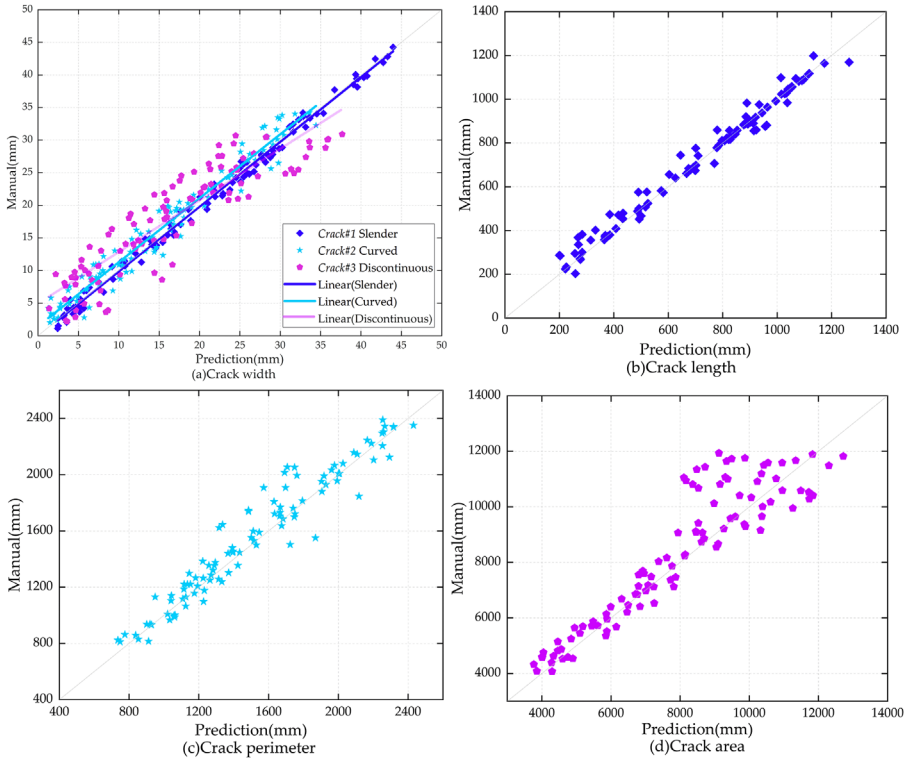

FIGURE 8 Linear correlation between algorithm prediction and manual measurement. (a) Crack width (100 sampling points were taken from three different types of cracks), (b) crack length (100 cracks taken from the S2T-Crack dataset), (C) crack perimeter (100 cracks taken from the S2T-Crack dataset), and (d) crack area (100 cracks taken from the S2T-Crack dataset).

CrackSegRefiner segmentation model, the absolute error for the maximum crack width is 2.74 mm, and the relative error is 5.94%, both of which are the best. The remaining measurement parameters achieved the lowest absolute and relative errors using our proposed IHD segmentation model, with all relative errors within acceptable percentages.

Figure 7 shows the results for Crack#2 (a curved shape crack). When using the CrackSegRefiner segmentation model (Table 4), the relative errors for the average and minimum crack widths were 8.48% and 9.86%, respectively, both being the smallest. Using our IHD segmentation model, the absolute and relative errors for the maximum width, length, and perimeter of the cracks were the lowest, with absolute errors of 2.53, 7.97, and 112.04 mm, and relative errors of 7.78%, 0.93%, and 5.64%, respectively. The proposed algorithm demonstrates very low errors in parameter measurement.

Figure 7 shows the results for Crack#3, executed on our equipment. When using the IHD, the absolute and relative errors for the average width, maximum width, minimum width, perimeter, and area of the cracks were the lowest, with absolute errors of 0.38 mm, 4.62 mm, 0.24 mm, 14.85 mm, 200.85 mm, and 843.20 mm^2, and relative errors of 12.94%, 13.39%, 17.57%, 1.48%, 8.53%, and 8.32%, respectively. The proposed method effectively avoids local branching and endpoint loss of the skeleton during crack extraction. It is effective for analyzing cracks under various conditions.

The correlation between the predicted and actual values of crack characteristics is illustrated in Figure 8. Figure 8a depicts 100 sampling points from three different types of cracks in the S2T-Crack dataset, focusing on crack width measurements. The measurement for Crack#3 Discontinuous exhibits the highest relative error, which increases as the complexity of the crack morphology rises, indicating a need for improved segmentation accuracy for fine cracks. Figure 8b presents the crack length measurements of 100 cracks from the S2T-Crack dataset; notable deviations in the relative error rate are observed, primarily due to insufficient detection accuracy for some fine cracks, resulting in predicted lengths being smaller than the actual lengths. Figure 8c encompasses the perimeter measurements of 100 cracks from the S2T-Crack dataset, while Figure 8d covers area measurements for the same dataset. The quantification results for crack area display the greatest deviation, potentially due to inadequate feature extraction for certain cracks and the misidentification of background pixels as cracks, which leads to broader boundaries for finer cracks. Although the quantification results of the aforementioned metrics deviate from the true values to some extent, they remain within an acceptable range.

5 | CONCLUSION

The present study undertakes an exploration into the efficacy of the real-time PCM framework, which is implemented upon autonomous mobile robots. This paper addresses two principal issues: the deployment of a crack segmentation model involving high-precision CNN models and the complexity and suboptimal real-time performance associated with the quantification of parameters. An IHD method is proposed, amalgamating relation-based and feature-based knowledge distillation to formulate a more precise and lightweight segmentation model. By leveraging the S2TCrack dataset, the IHD method attained an mIoU of 82.23% with an FPS rate of 77.72, thus fulfilling real-time requisites. Moreover, a real-time algorithm for quantifying crack surface features was devised, effectively resolving issues pertaining to crack edge branches and measurement losses. This algorithm enables the online real-time measurement of essential crack parameters, encompassing average width, maximum width, minimum width, length, perimeter, and area. In the forthcoming research endeavors, self-supervised learning will be incorporated to extract features from unlabeled data (Rafiei et al., 2024a, 2024b), consequently diminishing the reliance on extant labeled crack data and augmenting the convenience of crack segmentation. Concurrently, the focus will be on quantifying three-dimensional assessments based on geometric information such as crack depth, aiming to enhance the automation level of road maintenance.

ACKNOWLEDGMENTS

The study presented in the article was partially supported by the National Natural Science Foundation of China (No. 52378431, 52408454), Fundamental Research Funds for the Central Universities, CHD (No. 300102210302, 300102210118, 300102244712), the 111 Project of Low Carbon Smart Road Infrastructure Construction and Maintenance Discipline Innovation and Talent Introduction Base of Shaanxi Province, Scientific Innovation Practice Project of Postgraduates of Chang'an University (No. 300103724053).

REFERENCES

Al-Huda, Z., Peng, B., Algburi, R. N. A., Al-antari, M. A., AL-Jarazi, R., & Zhai, D. (2023). A hybrid deep learning pavement crack semantic segmentation. *Engineering Applications of Artificial Intelligence*, *122*, 106142.

Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). YOLACT: Real-time instance segmentation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, South Korea (pp. 9156–9165).

Chen, G., Choi, W., Yu, X., Han, T., & Chandraker, M. (2017). Learning efficient object detection models with knowledge distillation. *Neural Information Processing Systems*, Long Beach, CA.

Chen, J., & He, Y. (2022). A novel U-shaped encoder–decoder network with attention mechanism for detection and evaluation of road cracks at pixel level. *Computer-Aided Civil and Infrastructure Engineering*, *37*(13), 1721–1736.

Cho, J. H., & Hariharan, B. (2019). On the efficacy of knowledge distillation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* Seoul, South Korea (pp. 4794–4802).

De Nardin, A., Zottin, S., Piciarelli, C., Colombi, E., & Foresti, G. L. (2023). Few-shot pixel-precise document layout segmentation via dynamic instance generation and local thresholding. *International Journal of Neural Systems*, *33*(10), 2350052.

Deng, L., Zhang, A., Guo, J., & Liu, Y. (2023). An integrated method for road crack segmentation and surface feature quantification under complex backgrounds. *Remote Sensing*, *15*(6), 1530.

García-Aguilar, I., García-González, J., Luque-Baena, R. M., López-Rubio, E., & Domínguez, E. (2023). Optimized instance segmentation by super-resolution and maximal clique generation. *Integrated Computer-Aided Engineering*, *30*(3), 243–256.

Gong, H., Liu, L., Liang, H., Zhou, Y., & Cong, L. (2024). A state-of-the-art survey of deep learning models for automated pavement crack segmentation. *International Journal of Transportation Science and Technology*, *13*, 44–57.

Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, *129*(6), 1789–1819.

Han, C., Yang, H., Ma, T., Wang, S., Zhao, C., & Yang, Y. (2024). CrackDiffusion: A two-stage semantic segmentation framework for pavement crack combining unsupervised and supervised processes. *Automation in Construction*, *160*, 105332.

Hu, J., Yu, C., Yi, Z., & Zhang, H. (2023). Enhancing robustness of medical image segmentation model with neural memory ordinary differential equation. *International Journal of Neural Systems*, *33*(12), 2350060.

Jocher, G., Chaurasia, A., & Qiu, J. (2023). *Ultralytics YOLO*. https://github.com/ultralytics/ultralytics

Kheradmandi, N., & Mehranfar, V. (2022). A critical review and comparative study on image segmentation based techniques for pavement crack detection. *Construction and Building Materials*, *321*, 126162.

Lau, S. L. H., Chong, E. K. P., Yang, X., & Wang, X. (2020). Automated pavement crack segmentation using U-Net based convolutional neural network. *IEEE Access*, *8*, 114892–114899.

Li, Z., Zhang, T., Miao, Y., Zhang, J., Torbaghan, M. E., He, Y., & Dai, J. (2024). Automated quantification of crack length and width in asphalt pavements. *Computer-Aided Civil and Infrastructure Engineering*, *39*(21), 33317–33336.

Liu, J., Yang, Z., Qi, H., Jiao, T., Li, D., Wu, Z., Zheng, N., & Xu, S. (2024). Deep learning-assisted automatic quality assessment of concrete surfaces with cracks and bugholes. *Advanced Engineering Informatics*, *62*, 102577.

Nishikawa, T., Yoshida, J., Sugiyama, T., & Fujino, Y. (2011). Concrete crack detection by multiple sequential image filtering. *Computer-Aided Civil and Infrastructure Engineering*, *27*(1), 29–47.

Ong, J. C., Ismadi, M.-Z. P., & Wang, X. (2022). A hybrid method for pavement crack width measurement. *Measurement*, *197*, 111260.

Park, W., Kim, D., Lu, Y., & Cho, M. (2019). Relational knowledge distillation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA (pp. 3962–3971).

Rafiei, M. H., Gauthier, L. V., Adeli, H., & Takabi, D. (2024a). Self-supervised learning for electroencephalography. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(2), 1457–1471.

Rafiei, M. H., Gauthier, L. V., Adeli, H., & Takabi, D. (2024b). Self-supervised learning for near-wild cognitive workload estimation. *Journal of Medical Systems*, *48*(1), 107.

Ren, Y., Huang, J., Hong, Z., Lu, W., Yin, J., Zou, L., & Shen, X. (2020). Image-based concrete crack detection in tunnels using deep fully convolutional networks. *Construction and Building Materials*, *234*, 117367.

Shi, Y., Cui, L., Qi, Z., Meng, F., & Chen, Z. (2016). Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, *17*(12), 3434–3445.

Sun, X., Xie, Y., Jiang, L., Cao, Y., & Liu, B. (2022). DMA-Net: DeepLab with multi-scale attention for pavement crack segmentation. *IEEE Transactions on Intelligent Transportation Systems*, *23*(10), 18392–18403.

Tao, H., Liu, B., Cui, J., & Zhang, H. (2023). A convolutional-transformer network for crack segmentation with boundary awareness. *2023 IEEE International Conference on Image Processing (ICIP)*, Kuala Lumpur, Malaysia (pp. 86–90).

Wang, J., Zhang, L., & Zhang, Y. (2022). Mixture 2D convolutions for 3D medical image segmentation. *International Journal of Neural Systems*, *33*(1), 2250059.

Wang, X., Mao, Z., Liang, Z., & Shen, J. (2024). Multi-scale semantic map distillation for lightweight pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, *25*(10), 15081–15093.

Wang, X., Wang, T., & Li, J. (2023). Advanced crack detection and quantification strategy based on CLAHE enhanced DeepLabv3+. *Engineering Applications of Artificial Intelligence*, *126*, 106880.

Wang, X., Zhang, R., Shen, C., Kong, T., & Li, L. (2021). SOLO: A simple framework for instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(11), 1–1.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). *SegFormer: Simple and efficient design for semantic segmentation with transformers*. arXiv:2105.15203. https://arxiv.org/abs/2105.15203

Xu, G., Yue, Q., & Liu, X. (2023). Deep learning algorithm for real-time automatic crack detection, segmentation, qualification. *Engineering Applications of Artificial Intelligence*, *126*, 107085.

Yang, C., An, Z., Zhou, H., Zhuang, F., Xu, Y., & Zhang, Q. (2023). Online knowledge distillation via mutual contrastive learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(8), 10212–10227.

Yang, X., Zhang, J., Liu, W., Jing, J., Zheng, H., & Xu, W. (2024). Automation in road distress detection, diagnosis and treatment. *Journal of Road Engineering*, *4*(1), S2097049824000064.

Zhang, J., Yang, X., Wang, W., Brilakis, I., Davletshina, D., & Wang, H. (2024). Robust ELM-PID tracing control on autonomous mobile robot via transformer-based pavement crack segmentation. *Measurement*, *242*(Part C), 116045.

Zhang, J., Yang, X., Wang, W., Brilakis, I., Davletshina, D., Wang, H., & Cao, M. (2024). Segment-to-track for pavement crack with lightweight neural network on unmanned wheeled robot. *Automation in Construction*, *161*, 105346.

Zhang, J., Yang, X., Wang, W., Brilakis, I., Wang, H., & Ding, L. (2024). Transformer-based pavement crack tracking with neural-PID controller on vision-guided robot. *ISARC Proceedings*, Lille, France (pp. 699–706).

Zhang, J., Yang, X., Wang, W., Guan, J., Ding, L., & Lee, V. C. (2023). Automated guided vehicles and autonomous mobile robots for recognition and tracking in civil engineering. *Automation in Construction*, *146*, 104699.

Zhang, J., Yang, X., Wang, W., Guan, J., Liu, W., Wang, H., Ding, L., & Lee, V. C. S. (2024). Cross-entropy-based adaptive fuzzy control for visual tracking of road cracks with unmanned mobile robot. *Computer-Aided Civil and Infrastructure Engineering*, *39*(6), 891–910.

Zhang, J., Yang, X., Wang, W., Wang, H., Ding, L., El Badawy, S., & You, Z. (2024). Vision-guided robot for automated pixel-level pavement crack sealing. *Automation in Construction*, *168*, 105783.

Zhang, Y., Chen, C., Wu, Q., Lu, Q., Zhang, S., Zhang, G., & Yang, Y. (2018). A kinect-based approach for 3D pavement surface reconstruction and cracking recognition. *IEEE Transactions on Intelligent Transportation Systems*, *19*(12), 3935–3946.

Zheng, J., Chen, L., Wang, J., Chen, Q., Huang, X., & Jiang, L. (2024). Knowledge distillation with T-Seg guiding for lightweight automated crack segmentation. *Automation in Construction*, *166*, 105585.

Zhu, G., Liu, J., Fan, Z., Yuan, D., Ma, P., Wang, M., & Wang, K. C. P. (2024). A lightweight encoder–decoder network for automatic pavement crack detection. *Computer-Aided Civil and Infrastructure Engineering*, *39*(12), 1743–1765.

Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., & &Wang, S. (2019). DeepCrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, *28*(3), 1498–1512.

How to cite this article: Zhang, J., Ding, L., Wang, W., Wang, H., Brilakis, I., Davletshina, D., Heikkilä, R., & Yang, X. (2025). Crack segmentation-guided measurement with lightweight distillation network on edge device. *Computer-Aided Civil and Infrastructure Engineering*, 1–18. https://doi.org/10.1111/mice.13446