



Assessment of YOLO-Based Deep Learning Models for Pavement Crack Segmentation and Crack Extent Estimation

Gholamali Shafabakhsh¹ ; Sina Jabbari^{2,*}

1. Professor, Faculty of Civil Engineering, Semnan University, Semnan, Iran

2. Ph.D. Candidate, Faculty of Civil Engineering, Semnan University, Semnan, Iran

* Corresponding author: sinajabbari@semnan.ac.ir

ARTICLE INFO

Article history:

Received: 05 July 2026

Revised: 31 August 2026

Accepted: 22 September 2026

Keywords:

Pavement crack detection;

Instance segmentation;

YOLO26;

YOLO11;

Road maintenance.

ABSTRACT

Pavement crack segmentation is a critical task for improving road safety, supporting pavement management systems, and prioritizing maintenance actions in limited-data conditions. This study conducts a comprehensive evaluation of six YOLO-based instance segmentation models, comprising YOLO26 and YOLO11 variants (n, s, m), for pixel-level pavement crack segmentation using the CRACK500 dataset. The dataset, containing 500 pavement crack images with one crack class, was prepared in YOLO segmentation format and split into 350 training images, 75 validation images, and 75 independent test images. All models were trained under a unified experimental configuration. Performance was assessed using Mask mAP@50-95, Mask mAP@50, Precision, Recall, and F1-score. Results showed that YOLO26m-seg achieved the highest validation Mask mAP@50-95 of 0.2972 and was selected for testing. On the test set, it obtained a Mask mAP@50 of 0.5003 and a Mask mAP@50-95 of 0.1809. Increasing the input size to 640×640 improved Mask mAP@50-95 to 0.2554. Statistical validation using the Friedman test confirmed significant overall differences among models ($\chi^2 = 11.578$, $p = 0.0411$), although Kendall's $W = 0.0322$ indicated a small effect size. Finally, predicted masks were converted into Crack Area Ratio values to classify crack extent into low, moderate, and high severity levels. The innovation lies in providing a statistically supported YOLO benchmark and linking segmentation outputs to automated severity screening.

E-ISSN: 2345-4423

© 2026 The Authors. Journal of Rehabilitation in Civil Engineering published by Semnan University Press.

This is an open access article under the CC-BY 4.0 license. (<https://creativecommons.org/licenses/by/4.0/>)

How to cite this article:

Shafabakhsh, G, Jabbari, S. (2027). Assessment of YOLO-based deep learning models for pavement crack segmentation and crack extent estimation. Journal of Rehabilitation in Civil Engineering, 15(1), 2706. <https://doi.org/10.22075/jrce.2026.2706>

1. Introduction

Pavement deterioration is a persistent challenge in transportation infrastructure management because it affects road safety, ride quality, maintenance cost, and long-term serviceability. Among different pavement distresses, cracking is one of the earliest and most informative indicators of structural and functional degradation. Surface cracks may occur due to repeated traffic loading, thermal stress, asphalt aging, moisture infiltration, weak base layers, and environmental exposure. If not identified and treated in the early stages, these defects can lead to more serious deterioration forms such as potholes, surface deformation, raveling, and localized structural failure. Accurate crack detection and severity assessment are thus critical elements of pavement management systems, rehabilitation planning, and maintenance prioritization [1]. Traditional techniques for the assessment of pavement condition, e.g., visual inspection and Pavement Condition Index-based rating procedures, provide practical approaches for maintenance decision-making, but they are labor-intensive, time-consuming, and susceptible to inspector judgment, lighting conditions and survey consistency [2].

Traditional crack inspection is commonly performed through manual field surveys or visual interpretation of pavement images by trained inspectors. Although these approaches are still used by road agencies, they suffer from subjectivity, high operational cost, traffic safety risks, and limited scalability when large road networks require repeated monitoring. In recent years, image-based inspection has become an attractive alternative because it can reduce direct field exposure, provide repeatable visual records, and support automated analysis. Computer vision methods have been increasingly used for defect detection and condition assessment in asphalt pavements, concrete structures, bridges, tunnels, and other civil infrastructure systems [3]. In the broader field of Structural Health Monitoring, sensing, imaging, and data-driven approaches have been employed to improve damage identification, monitoring reliability, and maintenance planning [4]. However, for pavement networks, camera-based inspection combined with deep learning is particularly practical because pavement surface distresses are visually observable and can be processed at image level without requiring dense instrumentation [5].

The development of deep learning has significantly changed the way pavement distress detection is performed. Early image processing approaches relied heavily on handcrafted features, thresholding, edge detection, morphology, and texture analysis. These approaches often reached high accuracy in laboratory settings, but were less robust when pavement images had shadows, stains, aggregate texture, low contrast, oil marks, lane markings, or surface noise. Deep learning models, especially convolutional neural networks, are capable of learning hierarchical visual features directly from image data, and have shown great potential for road damage detection and crack recognition under complex field conditions [6]. Large-scale studies of road damage have demonstrated that deep neural networks can allow automated road inspection across a variety of regions and surface conditions, although generalization to different countries, camera systems, and pavement types is still difficult [7]. In crack-specific applications, convolutional architectures have improved the ability to detect fine linear defects that are difficult to distinguish using low-level features alone [8].

Despite these advances, pavement crack segmentation remains a technically demanding task. Cracks are usually thin, irregular, discontinuous, and sparsely distributed across the image. In many pavement images, crack pixels occupy only a small fraction of the total image area, while the non-crack background contains heterogeneous asphalt texture and visual noise. This creates a strong crack-background imbalance and makes pixel-level localization more difficult than image-level classification or bounding-box detection. DeepCrack addressed this issue by learning hierarchical convolutional features for crack representation and showed that multi-level feature fusion can improve crack detection under low continuity and low contrast conditions [9]. Similarly, Feature Pyramid and Hierarchical Boosting Network was proposed to integrate semantic information with low-level features and to balance easy and

hard samples during pavement crack detection [10]. Structured prediction-based convolutional methods also demonstrated that modeling spatial relationships among neighboring pixels can improve crack localization compared with isolated pixel classification [11].

Semantic and instance segmentation approaches are particularly important for pavement crack analysis because they provide pixel-level masks rather than only approximate defect locations. U-Net based models have been widely used for crack segmentation due to its encoder-decoder structure which integrates the contextual information and spatial information [12]. The original U-Net architecture was proposed for biomedical image segmentation, where the skip connections and localization properties were influential in many segmentation tasks [13]. Mask R-CNN improved object detection to instance segmentation by introducing a mask prediction branch, thus providing a universal framework for object-level mask estimation [14]. Fully convolutional networks laid the foundation for dense pixel-wise prediction by substituting fully connected layers with convolutional operations [15]. In pavement applications, these segmentation-oriented methods are valuable because crack masks can be converted into quantitative indicators such as crack area, crack density, and crack area ratio, which are more informative for severity estimation than bounding boxes alone.

Public crack datasets have played a central role in benchmarking crack detection and segmentation algorithms. The CrackForest dataset provided challenging pavement crack images and demonstrated the difficulty of detecting cracks in the presence of noisy surface textures [16]. Deep convolutional methods have been proposed for road crack detection based on patch-based or pixel-wise learning strategies, revealing that deep models can achieve better performance than traditional edge-based methods, provided with sufficient annotations [17]. CRACK500 is one of the most popular pavement crack datasets for pixel-level crack detection and segmentation. The original CRACK500 dataset contains 500 pavement crack images with an approximate resolution of 2000×1500 pixels, collected using mobile phones on the main campus of Temple University. Each image was divided into 16 non-overlapping regions, and only regions with more than 1000 crack pixels were retained. The dataset provides pixel-level crack annotations and has been used in several pavement crack segmentation studies [10]. Because CRACK500 contains thin cracks, low-contrast regions, complex asphalt texture, and irregular crack patterns, it provides a suitable benchmark for evaluating pixel-level crack segmentation models under realistic visual conditions [10].

Object detection frameworks have also been extensively used in pavement distress analysis because they provide a practical trade-off between accuracy and computational efficiency. The YOLO framework introduced a unified real-time detection approach by formulating object detection as a single regression problem from image pixels to bounding boxes and class probabilities [18]. Subsequent YOLO versions improved detection accuracy, multi-scale feature representation, and real-time performance, making them attractive for infrastructure inspection tasks where fast inference is required [19,20]. YOLOv4 further enhanced the speed-accuracy balance using architectural and training improvements such as CSP connections, mosaic augmentation, and improved loss strategies [21]. YOLOv7 introduced trainable bag-of-freebies and achieved strong real-time object detection performance across different speed and accuracy settings [22]. Ultralytics YOLO implementations further expanded the applicability of YOLO models to object detection, instance segmentation, pose estimation, classification, and oriented object detection [23].

YOLO-based models have been increasingly used in pavement distress detection, road crack analysis, and infrastructure monitoring because of their computational efficiency and real-time detection capability. Most previous YOLO-based pavement studies have focused on bounding-box detection of cracks, potholes, patches, and other surface defects. Bounding-box detection is useful to determine the approximate location of pavement distress, but it only provides coarse spatial information and cannot directly measure the actual crack area. For crack severity assessment, pixel-level segmentation is more

appropriate to quantitatively estimate the crack region in an image. By generating crack masks, segmentation models can support image-based crack area measurement and provide a basis for severity indicators such as the Crack Area Ratio. Therefore, YOLO segmentation models provide an opportunity to combine the efficiency of YOLO architectures with mask-based crack quantification.

YOLO11 supports object detection and instance segmentation tasks, making it suitable for evaluating segmentation-oriented pavement crack analysis within modern real-time vision frameworks [24]. YOLO26 has also been introduced as a unified real-time vision model family supporting detection, instance segmentation, pose estimation, classification, and oriented detection, with emphasis on efficient deployment and task-specific heads [25]. In this study, these segmentation-oriented YOLO models were evaluated for pixel-level pavement crack segmentation and subsequent crack severity assessment.

Although YOLO-based pavement distress detection has advanced rapidly, several methodological gaps remain. First, many pavement studies still focus on object detection rather than pixel-level segmentation. Second, studies that use crack segmentation often report aggregate metrics without conducting image-wise statistical validation of model differences. Third, limited-data conditions remain a practical challenge because preparing pixel-level crack masks is more expensive and time-consuming than preparing image labels or bounding boxes. Fourth, while CRACK500 has been widely used for crack segmentation, relatively few studies connect YOLO-based instance segmentation outputs to severity indicators derived from predicted crack area. A recent YOLOv8-based road crack detection study showed that image preprocessing, crack type distribution, and dataset balance can substantially influence detection performance across datasets such as CFD, CrackTree200, Crack500, and CrackVariety [26]. This highlights the need to report dataset characteristics, model configuration, evaluation metrics, and statistical analysis transparently.

Statistically supported model evaluation is important in pavement crack segmentation because model performance may vary across architectures, input sizes, and evaluation metrics. Recent YOLO-based pavement distress studies have shown that statistical analysis can provide a more reliable basis for comparing deep learning models under limited-data conditions [27]. However, many of these studies have primarily focused on bounding-box object detection, whereas the present study addresses crack-specific instance segmentation and severity assessment using predicted masks.

Beyond vision-based crack detection, damage severity quantification has also been studied extensively in the broader structural health monitoring literature using vibration- and signal-processing-based approaches. For example, wavelet packet transform combined with peak-picking modal identification has been used to quantify damage severity in steel structures [28], and signal-processing pipelines based on Fourier and wavelet analysis have been applied to detect and localize damage in steel braced frames [29] and in timber bridge structures using Fourier decomposition, time series modeling, and machine learning classifiers [30]. These studies confirm that damage severity quantification is a recurring and actively developed research theme across civil engineering subfields, even though the underlying sensing modality (vibration/acceleration signals) differs fundamentally from the image-based, mask-derived severity indicator used in this study; we cite them here to better position the present crack-area-ratio-based severity screening within this wider methodological context.

This distinction is important because segmentation performance is more sensitive to boundary-level errors than detection performance. In object detection, a predicted bounding box may be considered correct when it approximately encloses the damaged region. In contrast, segmentation requires close pixel-level agreement between the predicted crack mask and the ground-truth mask. Therefore, small deviations along thin crack boundaries can substantially reduce strict mask-based metrics such as Mask mAP@50-95, Dice, and IoU. For this reason, pavement crack segmentation should be evaluated using metrics that account not only for detection quality but also for spatial overlap and pixel-level localization accuracy.

In computer vision benchmarks, mean Average Precision at different Intersection over Union thresholds has become a standard criterion for evaluating object detection and segmentation models [31]. Nevertheless, for thin crack masks, strict IoU thresholds may penalize small boundary shifts even when the predicted mask visually follows the crack pattern. For this reason, complementary overlap metrics such as Dice coefficient and Jaccard index are useful for image-wise crack mask evaluation [32,33]. In addition, when several models are compared on the same test images, repeated-measures statistical procedures are more suitable than independent-sample comparisons because each image is evaluated by all models. Non-parametric methods such as the Friedman test and Wilcoxon signed-rank test can be used when metric distributions do not satisfy normality assumptions or when sample size is limited [34,35].

Based on these considerations, the present study evaluates six YOLO-based instance segmentation models for pavement crack detection, pixel-level segmentation, and severity assessment using the CRACK500 dataset. The evaluated models include YOLO26n-seg, YOLO26s-seg, YOLO26m-seg, YOLO11n-seg, YOLO11s-seg, and YOLO11m-seg. All models were trained under unified experimental settings using 500 pavement crack images divided into training, validation, and independent test sets. The best-performing model was selected using validation Mask mAP@50-95 and then evaluated on an independent test set. An additional input-size sensitivity experiment was conducted using the selected model to examine whether higher input resolution improves crack mask localization. Finally, the predicted masks were converted into crack area ratios to estimate image-level crack severity. This study therefore aims to provide a reproducible and statistically supported framework for YOLO-based pavement crack instance segmentation under limited-data conditions.

1.1. YOLO-based deep learning approaches for pavement crack analysis

As introduced above, the evolution of the YOLO family from a single-stage regression detector [18] toward increasingly accurate and efficient architectures [19,22] has made these models attractive for pavement distress analysis, offering a practical balance between detection accuracy, computational speed, and field-deployment potential. Building on this foundation, this section focuses specifically on how YOLO-based methods have been applied to pavement crack analysis and on the transition from bounding-box detection toward mask-based segmentation.

YOLO-based models have been extensively used for bounding-box detection of distress categories in pavement applications [16,19]. These studies show the feasibility of automated road surface monitoring, especially under the requirement of real-time or near-real-time inference. However, bounding-box detection is not always sufficient to analyze cracks as they have elongated and irregular geometry. A bounding box may cover a large amount of non-crack pavement background, which is not sufficient for estimating a crack's severity or the affected surface area. Instance segmentation is a more detailed representation as it predicts a pixel-level mask for each crack region. This is especially critical for thin cracks, where the difference between the bounding-box area and the actual crack area can be large [36]. Therefore, the transition from object detection to segmentation is critical for pavement crack assessment. Segmentation-oriented YOLO models can preserve the practical advantages of YOLO while enabling mask-based quantification. This makes them suitable for extracting crack area ratio, estimating spatial crack extent, and supporting image-level severity classification. Compared with conventional semantic segmentation models, YOLO segmentation models can also distinguish instances and provide both bounding boxes and masks. These outputs are useful for combining detection-level evaluation with pixel-level assessment. In the present study, YOLO11 and YOLO26 segmentation variants were selected because they represent modern YOLO-based segmentation frameworks with scalable model sizes and support for instance segmentation tasks [23,25].

Nevertheless, applying YOLO segmentation to pavement crack datasets remains challenging. Cracks often appear as narrow structures with discontinuous edges, low contrast, and variable width. Their masks

can be fragmented, and their pixel area is usually small compared with the background. These characteristics make crack segmentation sensitive to image resolution, annotation quality, and mask boundary accuracy. Previous crack segmentation studies using DeepCrack, FPHBN, U-Net-based networks, and structured prediction CNNs have shown that hierarchical features, multi-scale representation, and pixel-wise supervision are critical for accurate crack localization [9,13]. Therefore, YOLO-based segmentation models need to be evaluated not only using standard detection metrics but also using mask-level metrics and statistical image-wise analysis.

1.2. Problem statement

Although deep learning has improved pavement crack detection, several limitations remain in current research. As discussed in Section 1.1, the prevailing reliance on bounding-box object detection limits quantitative crack-area and severity estimation; pixel-level segmentation addresses this gap but is more difficult to annotate and model because cracks are thin, irregular, sparse, and sensitive to boundary mismatch, which in turn makes limited-data conditions common in practical research.

Another limitation is that model comparison is often based only on aggregate performance values such as mAP, precision, recall, or F1-score. While these metrics are useful, they do not fully show whether the performance differences between models are statistically meaningful across the same test images. This is important when several models are trained and evaluated under similar conditions, especially with a limited number of images. Without image-wise statistical analysis, it is difficult to determine whether the observed differences reflect real model-level behavior or only dataset-specific variation.

Furthermore, few studies connect YOLO-based crack segmentation with image-level severity assessment. For pavement management, detecting the presence of a crack is not enough; maintenance prioritization also requires information about the extent and severity of damage. Predicted crack masks can be used to calculate crack area ratio, which provides an interpretable severity indicator. However, this connection between segmentation output and severity classification is still less developed than conventional object detection workflows.

To address these gaps, this study investigates YOLO-based instance segmentation for pavement crack analysis using the CRACK500 dataset. The study compares six YOLO segmentation models under unified training settings, selects the best model based on validation Mask mAP@50-95, evaluates the selected model on an independent test set, examines the effect of increased input resolution, and estimates crack severity using predicted crack area ratio. This framework is intended to provide a reproducible model comparison and a practical link between deep learning-based crack segmentation and pavement maintenance decision support.

Relative to previous YOLO-based pavement distress studies, which predominantly report bounding-box detection metrics on multi-class datasets, the contribution of this study is threefold. First, it isolates and benchmarks the segmentation-specific behavior of two recent YOLO generations (YOLO11 and YOLO26) at the mask level, rather than the box level, which has not been jointly reported for CRACK500-based instance segmentation. Second, it couples this benchmark with an image-wise, repeated-measures statistical framework (Friedman/Wilcoxon tests with Holm-Bonferroni correction and bootstrap confidence intervals) so that observed differences between models can be judged for statistical, and not only numerical, significance. Third, and most directly motivated by the practical needs of pavement management, it introduces the Crack Area Ratio (CAR) as a lightweight, purely mask-derived engineering indicator: because CAR requires only the predicted segmentation mask and the image area, it can be computed automatically for every processed image without additional field measurement, offering a low-cost, first-pass screening signal that can be used to triage images for more detailed inspection before, and independently of, a full PCI-based survey.

2. Methodology

This section describes the methodology followed in this study for pixel-level pavement crack segmentation and image-level crack extent assessment. The proposed framework includes: CRACK500 dataset preparation, YOLO-based instance segmentation model training, loss function and evaluation metrics, validation-based model selection, independent test evaluation, input-size sensitivity analysis, statistical analysis, and crack extent assessment based on predicted segmentation masks. The methodology was developed to provide a reproducible experimental framework for comparing YOLO11 and YOLO26 segmentation models under consistent training and evaluation conditions.

The dataset used for this work is composed of 500 pavement crack images from the CRACK500 dataset. All images were processed for one-class instance segmentation, where the only target class was defined as crack. The dataset was split into three subsets: 350 images for training, 75 images for validation, and 75 images for independent testing. The main training experiments were conducted using an input image size of 448×448 pixels. In addition, an input-size sensitivity experiment was performed by retraining the selected model using 640×640 pixels to examine the effect of higher input resolution on mask-level segmentation performance.

The proposed framework consists of six main stages. First, the CRACK500 pavement crack images and their corresponding pixel-level annotations were prepared in YOLO segmentation format. Second, the dataset was divided into training, validation, and independent test subsets. Third, six YOLO-based instance segmentation models, including YOLO26n-seg, YOLO26s-seg, YOLO26m-seg, YOLO11n-seg, YOLO11s-seg, and YOLO11m-seg, were trained under the same experimental configuration. Fourth, the best-performing model was selected based on validation Mask mAP@50-95, which was considered the main metric because the objective of this study was accurate pixel-level crack localization. Fifth, the selected model was evaluated on the independent test set using both box-level and mask-level performance metrics. Finally, the predicted crack masks were used to calculate the Crack Area Ratio (CAR), which served as an image-level crack extent indicator for automated severity screening.

The overall workflow of the proposed pavement crack segmentation and crack extent assessment framework is shown in Fig. 1.

2.1. Dataset description and data preparation

For the main comparative experiments, all images were resized to 448×448 pixels. This input size was used to train all six YOLO-based segmentation models under identical experimental conditions. The evaluated models included YOLO26n-seg, YOLO26s-seg, YOLO26m-seg, YOLO11n-seg, YOLO11s-seg, and YOLO11m-seg. In addition, an input-size sensitivity experiment was conducted by retraining the selected model, YOLO26m-seg, using an input size of 640×640 pixels. This additional experiment was performed to evaluate whether higher input resolution could improve mask-level crack localization.

All model training and evaluation experiments were conducted in Google Colab using an NVIDIA T4 GPU. The same computational environment and training configuration were used across the main comparative experiments to ensure fair comparison among the evaluated models. Using a unified environment reduced the effect of hardware variation and allowed the observed differences to be attributed mainly to model architecture and segmentation performance.

The dataset characteristics and experimental setup used in this study are summarized in Table 1.

Representative samples from the CRACK500 dataset and their corresponding ground-truth crack masks are shown in Fig. 2. These samples demonstrate the visual complexity of the dataset, where crack regions often appear as thin, irregular, and low-contrast structures against heterogeneous pavement backgrounds. The figure also highlights the binary mask format used to represent crack annotations for pixel-level segmentation.

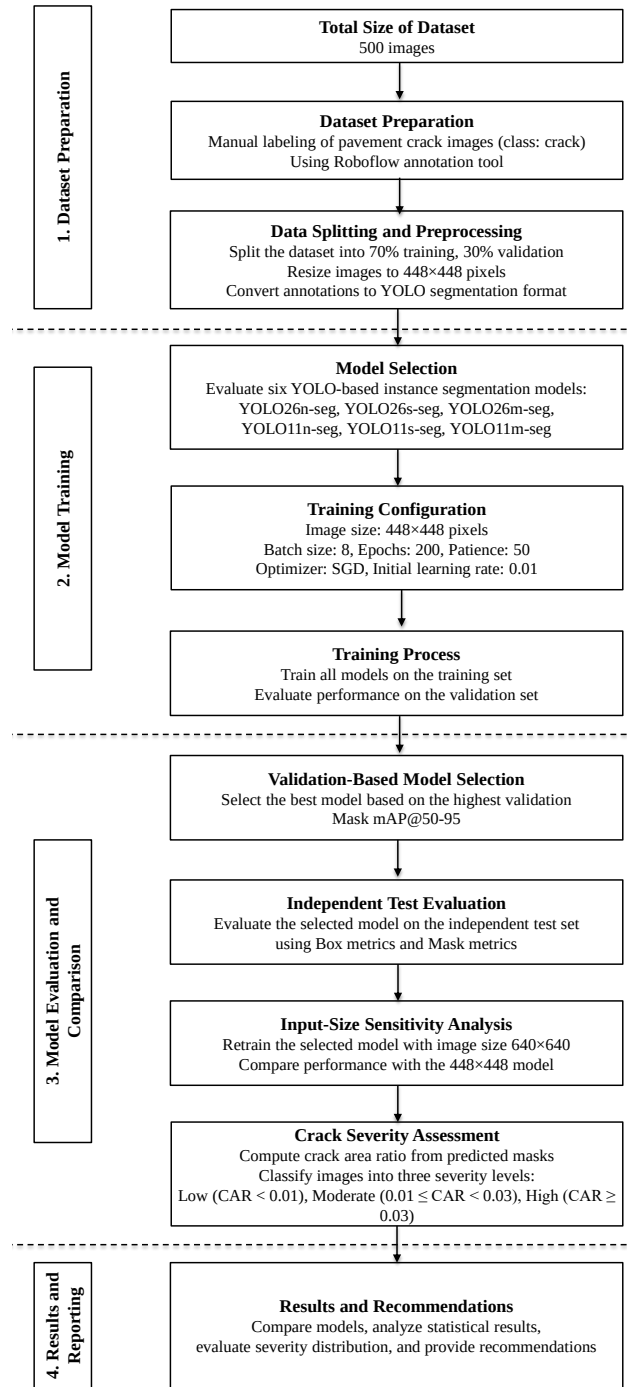


Fig. 1. Workflow of the proposed YOLO-based pavement crack segmentation and severity assessment framework.

2.2. Loss function and evaluation metrics

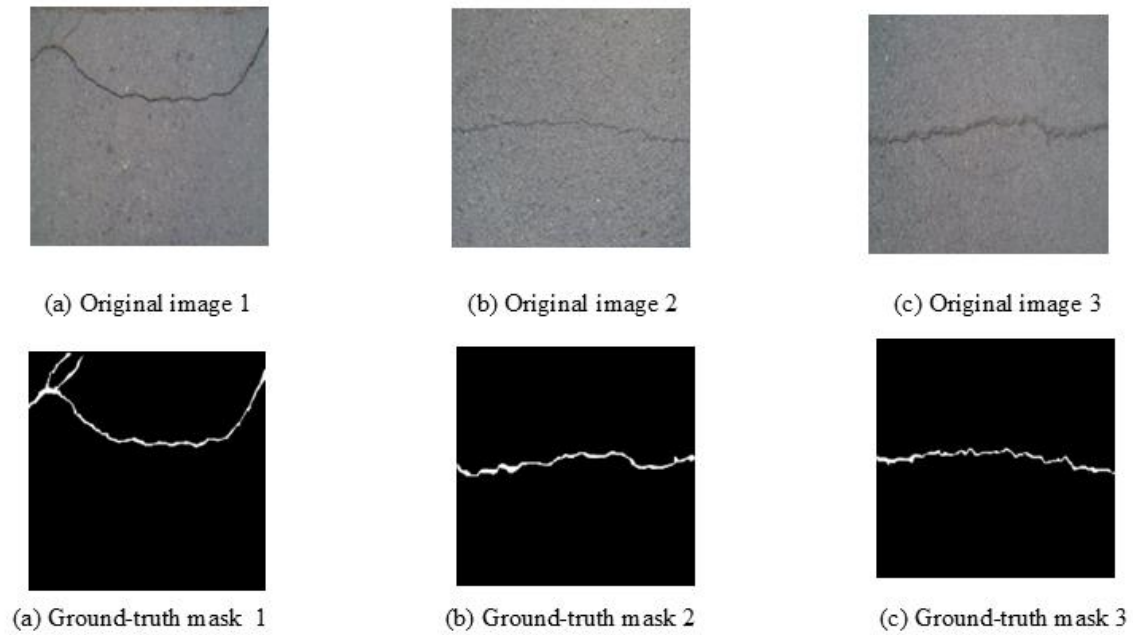
YOLO-based instance segmentation models optimize several learning objectives during training, including bounding-box localization, object classification, distribution-based box regression, and mask prediction. In segmentation mode, the total training loss can be generally expressed as a weighted combination of box loss, classification loss, distribution focal loss, and mask loss [23,25]:

$$L_{total} = \lambda_{box}L_{box} + \lambda_{cls}L_{cls} + \lambda_{dfl}L_{dfl} + \lambda_{masl}L_{mask} \quad (1)$$

where L_{total} is the total training loss, L_{box} represents the bounding-box localization loss, L_{cls} denotes the classification loss, L_{dfl} represents the distribution focal loss used for bounding-box regression, and L_{mask} denotes the segmentation mask loss. The coefficients λ_{box} , λ_{cls} , λ_{dfl} , and λ_{masl} control the relative contribution of each loss component.

Table 1. Identification of the CRACK500 dataset used in this study.

Item	Description
Dataset	CRACK500
Dataset source	Kaggle public dataset repository
Image type	Pavement crack images
Main task	Crack detection and instance segmentation
Number of images used	500 images
Number of classes	1
Class name	crack
Annotation type	Pixel-level crack mask
Annotation format	YOLO segmentation polygon format
Training images	350
Validation images	75
Independent test images	75
Main input image size	448 × 448 pixels
Additional input image size	640 × 640 pixels
Evaluated models	YOLO26n-seg, YOLO26s-seg, YOLO26m-seg, YOLO11n-seg, YOLO11s-seg, YOLO11m-seg
Selected model for additional experiment	YOLO26m-seg
Computational environment	Google Colab
GPU	NVIDIA T4 GPU
Severity / extent indicator	Crack Area Ratio based on predicted masks

**Fig. 2.** Representative pavement crack images from the CRACK500 dataset and their corresponding ground-truth binary masks used for pixel-level crack segmentation.

Since the dataset contained only one class, crack, the classification task was simpler than multi-class pavement distress detection; however, accurate mask prediction remained challenging because pavement cracks are thin, irregular, discontinuous, and sensitive to boundary-level errors.

The spatial overlap between the predicted crack mask and the ground-truth crack mask was evaluated using Intersection over Union. For segmentation, IoU measures the ratio between the intersection and union of the predicted and ground-truth crack regions [31,36]:

$$IoU = \frac{M_p \cap M_g}{M_p \cup M_g} \quad (2)$$

where M_p is the predicted crack mask and M_g is the ground-truth crack mask. The symbol $\cap$ represents the overlapping region between the predicted and ground-truth masks, while $\cup$ represents their union. A

higher IoU value indicates stronger spatial agreement between the predicted and ground-truth masks. Unlike bounding-box IoU, mask IoU evaluates pixel-level overlap and is therefore more appropriate for pavement crack segmentation.

Precision, recall, and F1-score were used to evaluate the detection and segmentation performance of the models. These metrics are calculated from true positive, false positive, and false negative predictions [37]:

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (5)$$

where TP represents correctly detected crack regions, FP represents non-crack regions incorrectly predicted as crack, and FN represents crack regions missed by the model. Precision indicates the reliability of positive crack predictions, while recall indicates the ability of the model to detect existing crack regions. F1-score is the harmonic mean of precision and recall and provides a balanced measure when both false detections and missed cracks are important.

For mask-level evaluation, the Dice coefficient was used as a complementary overlap metric. Dice coefficient is widely used in segmentation tasks because it directly measures the similarity between predicted and ground-truth masks [32]:

$$Dice = \frac{2|M_p \cap M_g|}{|M_p| + |M_g|} \quad (6)$$

where $|M_p|$ and $|M_g|$ denote the areas of the predicted and ground-truth masks, respectively. Dice is particularly useful for crack segmentation because crack pixels usually occupy a small portion of the image, creating a strong imbalance between crack and background pixels.

The mean Average Precision was used as the main criterion for model comparison. Average Precision is calculated as the area under the precision-recall curve:

$$AP = \int_0^1 P(R) dR \quad (7)$$

where $P(R)$ represents precision as a function of recall. The mean Average Precision is then calculated by averaging AP values across classes and IoU thresholds. Since this study used only one class, crack, the class-averaging component was limited to a single category. However, the IoU-threshold averaging remained important, especially for Mask mAP@50-95.

For a single class, mAP can be expressed as:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (8)$$

where N is the number of evaluated IoU thresholds and AP_i is the Average Precision at the i -th IoU threshold. In this study, mAP@50 and mAP@50-95 were reported. The mAP@50 metric evaluates AP at an IoU threshold of 0.50. The mAP@50-95 metric averages AP values across IoU thresholds from 0.50 to 0.95 with a step size of 0.05, following the COCO-style evaluation protocol [31]. Since Mask mAP@50-95 applies stricter IoU thresholds, it is more sensitive to boundary mismatch and small localization errors. This is particularly important for pavement crack segmentation because cracks are thin and irregular, and even small deviations from the ground-truth mask can reduce strict mask-level scores.

Regarding numerical precision, all performance metrics were computed internally with full floating-point precision as returned by the Ultralytics evaluation pipeline. In the tables, values are retained at higher

precision to preserve reproducibility and to support the statistical analyses described in Section 2.4 (e.g., paired Wilcoxon and bootstrap procedures, which are sensitive to small numerical differences). However, when metrics are discussed and interpreted in the narrative text, values are rounded to four decimal places, which is considered a practically meaningful level of precision for pavement engineering applications; the additional digits carried in the tables do not imply a corresponding level of practical measurement accuracy.

The confusion matrix concept used for crack segmentation is shown in Table 2. In this study, the positive class corresponds to crack pixels or detected crack regions, while the negative class corresponds to non-crack pavement background.

Table 2. Identification of prediction outcomes for pavement crack segmentation.

Ground Truth	Prediction: Crack	Prediction: Non-crack
Crack exists	True Positive (TP)	False Negative (FN)
No crack	False Positive (FP)	True Negative (TN)

2.3. Dataset preparation and model training

Building on the dataset composition and split already summarized in Section 2.1 and Table 1, all crack annotations were converted into YOLO segmentation polygon format, where each crack region was represented by normalized polygon coordinates; this format allowed the models to learn both object localization and pixel-level crack mask prediction. The same training and validation subsets were used for all six models to ensure a fair comparison, while the test subset was withheld from both training and validation-based model selection and reserved exclusively for independent evaluation.

The six evaluated models (YOLO26n/s/m-seg and YOLO11n/s/m-seg) were selected to compare different YOLO generations and model scales under one experimental configuration. The training hyperparameters were not tuned per model; rather, they followed the standard Ultralytics training recipe for YOLO segmentation models and were fixed across all six architectures so that any observed performance differences could be attributed to model design rather than to inconsistent hyperparameter choices. Specifically, the SGD optimizer with an initial learning rate of 0.01 follows the Ultralytics default configuration for YOLO detection/segmentation training, which is itself tuned empirically across large general-purpose benchmarks (COCO-scale detection/segmentation) and is a standard, defensible starting point absent a dataset-specific tuning budget. The batch size of 8 and input size of 448×448 pixels were constrained by the 16 GB memory of the single NVIDIA T4 GPU available in the Google Colab environment: for the largest evaluated model (YOLO26m-seg), activation memory scales approximately linearly with batch size $\times$ input resolution² $\times$ network depth, and empirically, batch sizes above 8-10 at 448×448 for an 'm'-scale segmentation model exceeded the T4's available memory once gradient buffers and the segmentation mask head's additional activation maps (proportional to input resolution², unlike the box-only head) were accounted for; a batch size of 8 was therefore the largest power-of-two-adjacent value that fit reliably across all six models without per-model memory tuning, which was itself a deliberate choice to keep the comparison uniform (see our response to the immediately following point on this being fixed by practical constraints rather than a per-architecture search). The patience value of 50 epochs (out of a 200-epoch budget) was set as an early-stopping safeguard against overfitting on the relatively small training subset ($n = 350$); this value follows the common heuristic of setting patience to roughly 20-25% of the total epoch budget, balancing the risk of premature stopping during the noisy early-training phase against excessive wasted computation once validation performance plateaus. We acknowledge that this configuration was fixed by practical resource constraints and standard heuristics rather than derived from a dedicated hyperparameter search for each of the six architectures, and we

discuss the resulting limitation, namely that a model-specific search could in principle shift the relative ranking among models, explicitly in Section 4.2.

The best-performing model was selected based on validation Mask mAP@50-95. This criterion was selected because it evaluates mask-level localization over multiple IoU thresholds and is stricter than mAP@50 alone. The specific model identified by this criterion, together with its validation-set performance value, is reported in Section 3 together with the other validation results, and that model was subsequently carried forward for independent test evaluation.

After model selection, the selected YOLO26m-seg model was evaluated on the independent test set. In addition, an input-size sensitivity experiment was conducted by retraining the selected model using an image size of 640×640 pixels. This additional experiment was performed to determine whether higher input resolution could improve crack mask localization, especially because pavement cracks are thin, narrow, and sensitive to boundary-level errors.

The model training configuration is summarized in Table 3.

Table 3. Training configuration used for YOLO-based pavement crack segmentation models.

Parameter	Value
Task	Instance segmentation
Dataset	CRACK500
Class	crack
Number of classes	1
Models	YOLO26n-seg, YOLO26s-seg, YOLO26m-seg, YOLO11n-seg, YOLO11s-seg, YOLO11m-seg
Training images	350
Validation images	75
Independent test images	75
Main input size	448×448 pixels
Additional input size	640×640 pixels for selected model
Epochs	200
Batch size	8
Optimizer	SGD
Initial learning rate	0.01
Patience	50
Model selection criterion	Validation Mask mAP@50-95
Selected model	YOLO26m-seg
Best validation epoch	154

Therefore, no distribution of bounding boxes across pavement distress classes was reported. Instead, representative crack images and their corresponding ground-truth masks are presented in Fig. 2, while the severity distribution of the independent test set is reported later in the Results section.

2.4. Statistical analysis

Statistical analysis was conducted to evaluate whether the observed performance differences among the YOLO-based segmentation models were consistent across the test images. Since all models were evaluated on the same images, the comparison followed a repeated-measures structure. In addition, because image-wise segmentation scores may not follow a normal distribution, non-parametric statistical tests were used.

The Friedman test was applied to compare the image-wise Mask Dice scores of the six YOLO-based segmentation models. The Friedman test is a non-parametric alternative to repeated-measures ANOVA

and is suitable when multiple models are evaluated on the same samples [34]. The null hypothesis was that all models had the same median performance across the evaluated images.

To estimate the magnitude of agreement or effect size in the Friedman test, Kendall's coefficient of concordance was calculated as:

$$W = \frac{\chi^2}{N(k-1)} \quad (9)$$

where W is Kendall's coefficient of concordance, χ^2 is the Friedman test statistic, N is the number of evaluated images, and k is the number of compared models. A larger W value indicates stronger differences among the ranked model performances.

When the Friedman test indicated an overall difference, pairwise comparisons were conducted using the Wilcoxon signed-rank test. Since multiple pairwise comparisons increase the probability of Type I error, Holm and Bonferroni corrections were applied to adjust the significance level [35,38]. In addition, bootstrap confidence intervals were calculated for image-wise Mask Dice scores to estimate uncertainty in model performance [39].

The selected YOLO26m-seg model was also compared at two input sizes, 448×448 and 640×640 pixels. For this paired comparison, image-wise Mask Dice and Mask IoU values were compared using the Wilcoxon signed-rank test. Bootstrap paired-difference confidence intervals were also calculated to examine whether the observed performance differences were practically stable across test images.

2.5. Crack severity assessment

In conventional pavement condition evaluation systems such as PCI and PAVER, the severity of longitudinal and transverse cracking is determined based on field-observed distress characteristics, including crack width, crack length, secondary random cracking, raveling or breakup around the crack, and sealant condition [1,2]. According to Shahin's pavement distress evaluation framework, longitudinal cracks are generally parallel to the pavement centerline or paving direction, while transverse cracks extend approximately perpendicular to the pavement centerline. For these cracks, low severity is generally associated with narrow or properly sealed cracks, moderate severity is associated with wider cracks or cracks surrounded by light random cracking, and high severity is associated with very wide cracks, medium-to-high severity random cracking around the main crack, or severe breakup of the pavement around the crack [1].

However, direct PCI-based severity classification could not be applied in this study because the CRACK500 dataset provides pixel-level crack masks but does not include field-calibrated crack width, physical scale information, crack depth, sealant condition, raveling severity, traffic-related field observations, or separate labels for longitudinal and transverse cracking. Moreover, all images were resized to fixed input sizes for YOLO training, which prevents reliable conversion of pixel width into real crack width in millimeters. Therefore, the severity assessment in this study was not designed to replace PCI-based field evaluation. Instead, a mask-derived image-level crack extent indicator was adopted to provide an automated and reproducible approximation of visible crack coverage.

Previous image-based pavement distress studies have used different strategies for severity assessment depending on the available data. When physical scale is available, the severity of linear cracks can be estimated from crack width. In contrast, for segmentation-based assessment, distress extent is often quantified using the proportion of damaged pixels or the ratio of the segmented distress area to the image area [40]. Similarly, recent multitask pavement assessment frameworks have shown that segmentation outputs can support severity and extent estimation because pixel-level masks provide information about the spatial coverage of pavement distress [38,41]. Based on this rationale, the present study adopted the Crack Area Ratio (CAR) as a mask-derived image-level crack extent indicator.

In this study, crack extent was estimated using the predicted crack area ratio. The crack area ratio represents the proportion of image pixels classified as crack by the segmentation model:

$$CAR = \frac{A_{\text{crack}}}{A_{\text{image}}} \quad (10)$$

where CAR is the crack area ratio, A_{crack} is the predicted crack mask area, and A_{image} is the total image area. This index quantifies the spatial extent of predicted crack pixels within each pavement image. Since the available dataset does not provide the physical distress measurements required for PCI-based severity classification, CAR was used as an automated image-level proxy for crack extent rather than as a direct PCI severity rating.

Based on the predicted crack area ratio, each test image was grouped into one of three image-level crack extent categories:

$$\text{Low severity: } CAR < 0.01 \quad (11)$$

$$\text{Moderate severity: } 0.01 \leq CAR < 0.03 \quad (12)$$

$$\text{High severity: } CAR \geq 0.03 \quad (13)$$

The thresholds were selected as operational image-level thresholds for this study. Images with $CAR < 0.01$ were considered low severity because crack pixels occupied less than 1% of the image area, indicating limited crack coverage. Images with $0.01 \leq CAR < 0.03$ were classified as moderate severity because the crack region became more spatially noticeable but did not dominate the pavement image. Images with $CAR \geq 0.03$ were classified as high severity because crack pixels occupied at least 3% of the image area, indicating more extensive visible cracking.

Therefore, the proposed CAR-based classification should be interpreted as a mask-derived image-level crack extent indicator, not as a standard PCI severity rating. In field PCI surveys, longitudinal and transverse crack severity should be determined using crack width, crack length, secondary cracking, sealant condition, and pavement breakup according to Shahin's distress identification criteria [1,2]. In contrast, the present study uses CAR because the available dataset supports pixel-level segmentation but does not provide the physical measurements required for direct PCI-based severity classification.

The difference between conventional PCI-based field severity assessment and the CAR-based image-level crack extent indicator used in this study is summarized in Table 4.

Table 4. Relationship between field-based PCI severity criteria and the image-based severity indicator used in this study.

Aspect	PCI-based field evaluation	CAR-based image-level evaluation in this study
Required information	Crack width, crack length, secondary cracking, raveling or breakup, sealant condition	Predicted crack mask area and total image area
Measurement unit	Millimeters, meters, and field distress observations	Pixel ratio
Crack type	Longitudinal and transverse cracks identified separately	Single class: crack
Severity basis	Standard PCI distress severity definitions	Operational image-level crack extent thresholds
Low category	Narrow or properly sealed cracks	$CAR < 0.01$
Moderate category	Wider cracks or light secondary cracking	$0.01 \leq CAR < 0.03$
High category	Very wide cracks, secondary cracking, or severe breakup	$CAR \geq 0.03$
Interpretation	Field-based PCI severity	Mask-derived image-level crack extent proxy

The proposed CAR-based categorization was not intended to replace PCI-based field severity assessment. Rather, it was used as a reproducible image-level crack extent indicator, consistent with segmentation-

based pavement assessment studies that quantify distress extent using damaged-pixel proportion or crack-area fraction [38,42].

3. Results

This study evaluated the performance of YOLO-based instance segmentation models for pavement crack detection and segmentation using a manually annotated pavement crack dataset. Six segmentation models with different model scales, including YOLO26n-seg, YOLO26s-seg, YOLO26m-seg, YOLO11n-seg, YOLO11s-seg, and YOLO11m-seg, were trained and validated under identical experimental conditions. The models were assessed using standard evaluation metrics, including Precision, Recall, F1-score, mAP@50, and mAP@50-95 for mask-level crack segmentation.

To ensure consistency and fair comparison, all models were trained using the same dataset split, input image size, optimizer, learning rate, batch size, number of epochs, and early-stopping criterion. The dataset included 500 pavement crack images, of which 350 images were used for training, 75 images for validation, and 75 images for independent testing. Since the objective of this study was pixel-level pavement crack segmentation rather than only bounding-box detection, the final model selection was mainly based on Mask mAP@50-95. This metric evaluates segmentation performance over multiple IoU thresholds and therefore provides a stricter assessment of mask localization accuracy.

The results are presented in five stages. First, the validation performance of all YOLO-based segmentation models is compared using mask-level metrics. Second, the best-performing model is evaluated on the independent test set. Third, the effect of increasing the input image size from 448×448 to 640×640 pixels is analyzed. Fourth, statistical tests are applied to examine whether the observed differences between models are statistically meaningful. Finally, the predicted crack masks are used to estimate image-level crack extent through the Crack Area Ratio.

3.1. Precision and recall trends

The validation Mask Precision and Mask Recall values obtained by the six YOLO-based segmentation models are presented in Fig. 3(a) and Fig. 3(b), respectively. These metrics were used to examine the balance between false-positive crack predictions and the ability of each model to identify crack regions within the validation images.

As shown in Fig. 3(a), YOLO26m-seg achieved the highest Mask Precision value (0.7541), indicating that this model produced the most reliable positive crack predictions among all evaluated models. This result suggests that YOLO26m-seg was more conservative in assigning the crack label and therefore generated fewer false-positive segmentations. In contrast, YOLO26n-seg yielded the lowest Mask Precision (0.5623), indicating weaker discrimination between crack and background pixels.

The Mask Recall results shown in Fig. 3(b) reveal a different trend. YOLO26s-seg achieved the highest Mask Recall (0.5138), demonstrating the strongest ability to recover crack pixels from the validation images. Among the YOLO11 variants, YOLO11m-seg obtained the highest recall (0.4795), followed by YOLO11s-seg (0.4703) and YOLO11n-seg (0.4658). The lowest recall was observed for YOLO26m-seg (0.4202), indicating that although this model produced highly precise crack predictions, it missed a larger portion of crack pixels compared with the other models.

The combined interpretation of Fig. 3(a) and Fig. 3(b) highlights a clear precision-recall trade-off. YOLO26m-seg achieved the highest precision but the lowest recall, while YOLO26s-seg achieved the highest recall with slightly lower precision. This behavior suggests that YOLO26m-seg favored more selective crack segmentation, whereas YOLO26s-seg produced broader crack coverage at the expense of

a higher false-positive tendency. Overall, the YOLO26 family demonstrated stronger validation behavior than the YOLO11 family, particularly in terms of precision and recall balance.

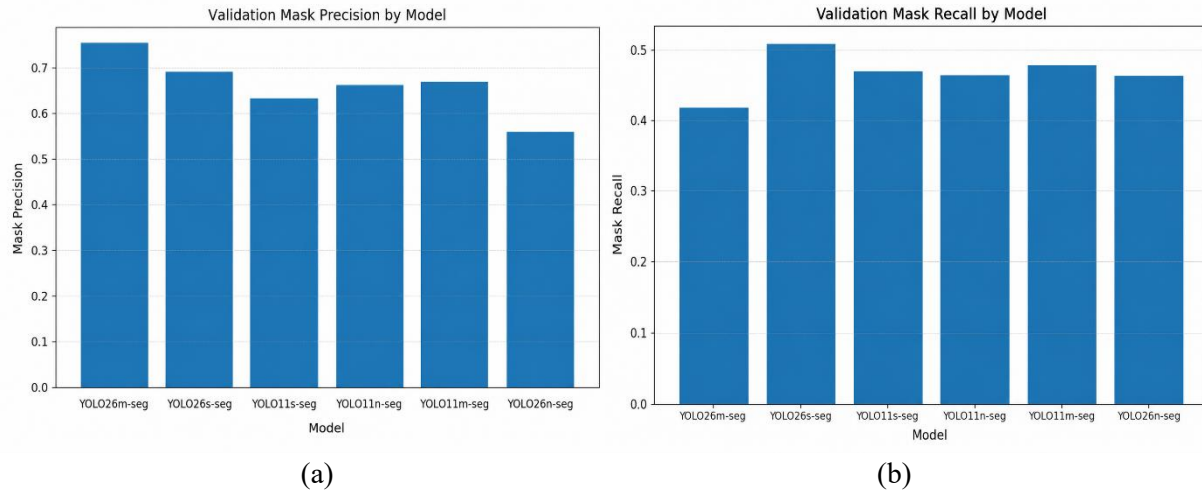


Fig. 3. Validation performance comparison in terms of (a) Mask Precision and (b) Mask Recall for the evaluated YOLO-based pavement crack segmentation models.

3.2. F1-score and Mask mAP analysis

The validation Mask F1-score and Mask mAP@50-95 results of the evaluated YOLO-based segmentation models are presented in Fig. 4(a) and Fig. 4(b), respectively. These metrics provide complementary insight into model performance. While the F1-score reflects the balance between precision and recall, the Mask mAP@50-95 metric evaluates segmentation quality over multiple and increasingly strict IoU thresholds, making it a more rigorous indicator of mask localization performance.

As shown in Fig. 4(a), YOLO26s-seg achieved the highest Mask F1-score (0.5898), indicating that this model provided the best balance between crack detection precision and recall on the validation set. Among the YOLO11 variants, YOLO11m-seg produced the highest F1-score (0.5593), followed by YOLO11n-seg (0.5476) and YOLO11s-seg (0.5413). The lowest F1-score was observed for YOLO26n-seg (0.5095), suggesting comparatively weaker overall balance in crack segmentation performance.

Although the F1-score favored YOLO26s-seg, the stricter segmentation metric shown in Fig. 4(b) produced a slightly different ranking. YOLO26m-seg achieved the highest validation Mask mAP@50-95 (0.2972), followed very closely by YOLO26s-seg (0.2943). Among the YOLO11 family, YOLO11s-seg obtained the best Mask mAP@50-95 (0.2843), followed by YOLO11n-seg (0.2752) and YOLO11m-seg (0.2605). The lowest Mask mAP@50-95 was again obtained by YOLO26n-seg (0.2432).

The difference between the F1-score ranking and the Mask mAP@50-95 ranking is important. The higher F1-score of YOLO26s-seg indicates stronger overall balance between precision and recall at the validation level. However, the higher Mask mAP@50-95 of YOLO26m-seg indicates that this model achieved slightly more accurate segmentation masks under stricter overlap thresholds. Because the objective of this study was accurate pixel-level crack localization, Mask mAP@50-95 was considered the primary model selection criterion. Therefore, despite the slightly higher F1-score of YOLO26s-seg, YOLO26m-seg was selected as the best-performing model for further evaluation on the independent test set.

Overall, the results indicate that the YOLO26 family outperformed the YOLO11 family in terms of the primary validation segmentation metric. The comparison also shows that performance differences among the top models were relatively small, especially between YOLO26m-seg and YOLO26s-seg. This

motivated the subsequent independent test evaluation and statistical analysis to determine whether the observed differences remained meaningful beyond the validation set.

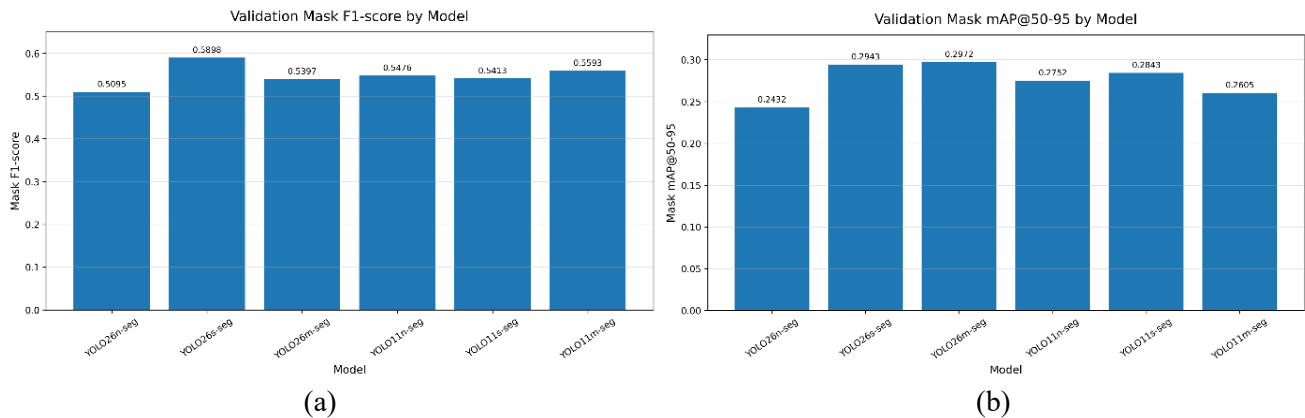


Fig. 4. Validation performance comparison in terms of (a) Mask F1-score and (b) Mask mAP@50-95 for the evaluated YOLO-based pavement crack segmentation models.

3.3. Comparison of Mask mAP@50 and Mask mAP@50-95 results

Mask mAP metrics provide a stricter and more comprehensive evaluation of segmentation performance than single-threshold indicators such as precision, recall, and F1-score. While precision and recall describe the tendency of a model to reduce false-positive and false-negative crack predictions, mAP evaluates the integrated precision-recall behavior over different confidence thresholds. In this study, Mask mAP@50 and Mask mAP@50-95 were used to assess the localization quality of predicted crack masks under relaxed and strict IoU conditions, respectively.

As shown in Fig. 5, the Mask mAP@50 values varied across the evaluated YOLO-based segmentation models. The highest Mask mAP@50 was achieved by YOLO26s-seg with a value of 0.5152, followed by YOLO26m-seg and YOLO11s-seg, which achieved values of 0.4896 and 0.4885, respectively. YOLO11n-seg also produced a comparable value of 0.4862, while YOLO11m-seg and YOLO26n-seg achieved lower values of 0.4704 and 0.4393, respectively. These results indicate that most medium and small segmentation models were able to detect the general crack regions under the relaxed IoU threshold.

However, the model ranking became more discriminative when stricter IoU thresholds were considered. As illustrated in Fig. 6, YOLO26m-seg achieved the highest Mask mAP@50-95 value of 0.2972, followed closely by YOLO26s-seg with 0.2943. Among the YOLO11 variants, YOLO11s-seg obtained the highest Mask mAP@50-95 value of 0.2843, followed by YOLO11n-seg and YOLO11m-seg with values of 0.2752 and 0.2605, respectively. The lowest strict mask localization performance was obtained by YOLO26n-seg, with a Mask mAP@50-95 of 0.2432.

The difference between Mask mAP@50 and Mask mAP@50-95 is important for interpreting segmentation performance. Mask mAP@50 mainly reflects whether the model can identify the approximate spatial location of crack regions, whereas Mask mAP@50-95 evaluates the consistency of mask prediction under progressively stricter overlap thresholds. Therefore, a model may achieve a relatively high Mask mAP@50 while still showing lower Mask mAP@50-95 if the predicted crack boundaries do not closely match the ground-truth masks.

This behavior is particularly relevant for pavement crack segmentation because cracks are usually thin, elongated, discontinuous, and irregularly shaped. Small deviations in predicted crack boundaries can lead to a considerable reduction in IoU, especially at higher thresholds. Consequently, all evaluated models showed lower Mask mAP@50-95 values than Mask mAP@50 values. This reduction does not necessarily

indicate failure in crack detection; rather, it reflects the higher difficulty of pixel-level crack mask localization compared with general crack region identification.

Overall, YOLO26s-seg provided the best performance under the relaxed Mask mAP@50 criterion, while YOLO26m-seg achieved the best performance under the stricter Mask mAP@50-95 criterion. Since the main objective of this study was accurate pixel-level crack segmentation, Mask mAP@50-95 was selected as the primary model selection metric. Based on this criterion, YOLO26m-seg was selected as the optimal model for subsequent independent test evaluation, input-size sensitivity analysis, and crack extent assessment.

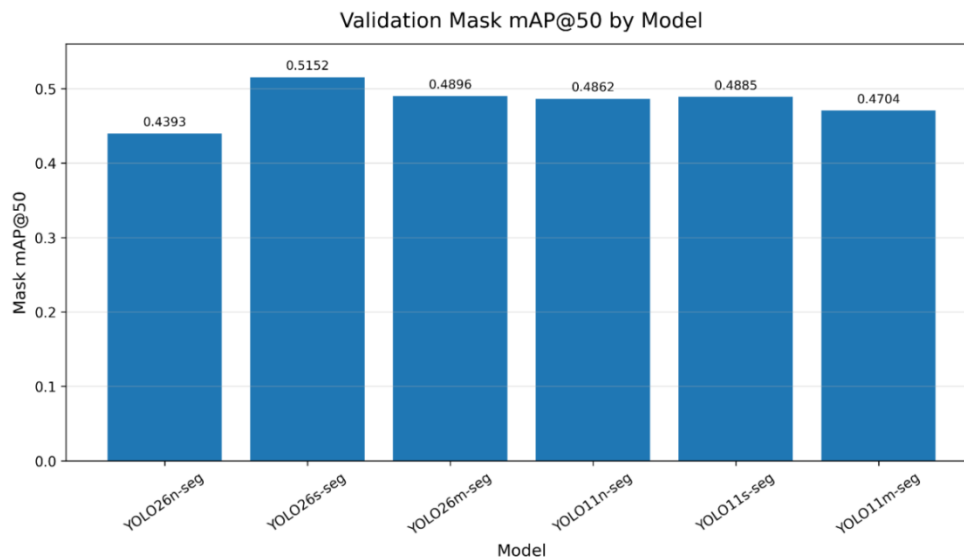


Fig. 5. Validation Mask mAP@50 of YOLO-based pavement crack segmentation models.

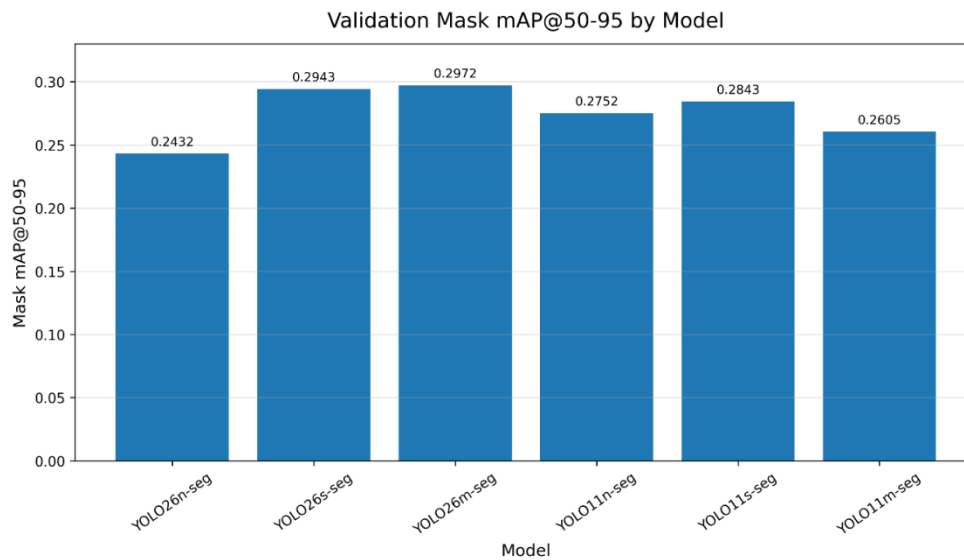


Fig. 6. Validation Mask mAP@50-95 of YOLO-based pavement crack segmentation models.

3.4. Training time and computational complexity

Training time and computational complexity were also evaluated because computational cost is an important factor in selecting a suitable deep learning model for pavement crack segmentation. In practical pavement monitoring applications, a model should not only provide reliable segmentation accuracy but also maintain reasonable training and deployment efficiency. Therefore, in addition to validation accuracy metrics, the training duration of each YOLO-based segmentation model was analyzed.

In this study, all models were trained using the same dataset, input image size, batch size, optimizer, learning rate, maximum number of epochs, and early-stopping configuration. The training duration values were extracted from the training results file and converted from seconds to minutes. Since the dataset size and training configuration were fixed for all experiments, differences in training duration were mainly associated with model architecture, model scale, segmentation-head computation, validation overhead, and runtime conditions.

As shown in Fig. 7, the training duration varied among the evaluated models. Among the YOLO11 models, YOLO11m-seg required the shortest training time, with approximately 32.01 min, followed by YOLO11n-seg and YOLO11s-seg, which required 34.34 min and 35.21 min, respectively. The relatively close training times among the YOLO11 variants suggest that runtime behavior was not determined only by nominal model size, but also by training dynamics and validation procedures.

For the YOLO26 family, the training duration was generally higher. YOLO26s-seg required approximately 38.32 min, while YOLO26n-seg required 45.10 min. The longest training duration was observed for YOLO26m-seg, which required approximately 62.20 min. This higher computational cost is consistent with the greater model capacity of the medium-scale architecture and the additional computational burden required for instance segmentation and mask prediction.

The comparison between training duration and validation performance indicates a clear trade-off between computational cost and segmentation accuracy. Although YOLO11m-seg required the lowest training time, it did not achieve the best validation Mask mAP@50-95. In contrast, YOLO26m-seg required the longest training duration but achieved the highest validation Mask mAP@50-95 value. This suggests that the additional computational cost of YOLO26m-seg contributed to improved strict mask localization performance, which is particularly important for thin and irregular pavement crack structures.

However, computational efficiency should also be considered when selecting a model for practical applications. YOLO26s-seg provided a competitive alternative because it achieved the highest Mask F1-score and a Mask mAP@50-95 value very close to YOLO26m-seg, while requiring substantially less training time. Therefore, YOLO26m-seg can be considered the most accuracy-oriented model, whereas YOLO26s-seg offers a more balanced configuration between segmentation performance and computational efficiency.

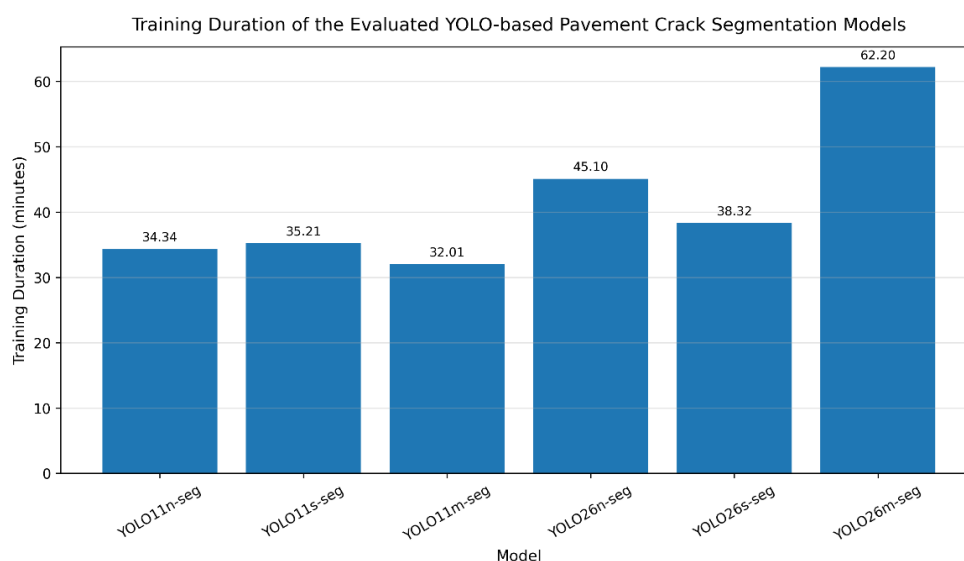


Fig. 7. Training duration of the evaluated YOLO-based pavement crack segmentation models.

Overall, the results show that model selection for pavement crack segmentation should be based on both segmentation accuracy and computational demand. For applications where strict pixel-level crack

localization is the main objective, YOLO26m-seg is preferable. For applications requiring lower computational cost with competitive segmentation performance, YOLO26s-seg may represent a more practical option.

It should be noted that the durations reported in Fig. 7 describe training time only, not inference-time throughput. Deployment-oriented inference-speed benchmarking (e.g., frames per second or per-image latency) was not performed in this study; this is discussed explicitly as a limitation in Section 4.2, together with the reasoning for why such a benchmark was not attempted on the shared Colab training environment used here.

3.5. Statistical analysis

There were three phases in the statistical analysis procedure. First, the overall difference among the six YOLO-based segmentation models was evaluated using image-wise Mask Dice scores. Second, pairwise post-hoc comparisons were performed to determine whether any specific model showed statistically significant superiority over the others. Third, bootstrap confidence intervals were calculated to assess the uncertainty and stability of the mean image-wise Mask Dice scores. All statistical tests were conducted using Python 3.11 with the pandas, scipy.stats, and statsmodels libraries. The significance level was set at $\alpha = 0.05$.

The analysis was conducted using 72 test images with complete prediction results across all six models. The image-wise Mask Dice score was selected for the statistical comparison because it directly measures the pixel-level agreement between the predicted crack mask and the corresponding ground-truth mask. This metric is particularly suitable for pavement crack segmentation, where small boundary deviations can affect the overlap between predicted and reference masks.

As shown in Fig. 8, the statistical workflow started with the Friedman test. The Friedman test indicated a statistically significant overall difference among the evaluated models, with $\chi^2 = 11.5780$ and $p = 0.0411$. This result shows that the six YOLO-based segmentation models did not produce identical image-wise segmentation performance. However, Kendall's coefficient of concordance was $W = 0.0322$, indicating a small effect size. Therefore, although the overall difference was statistically significant, the magnitude of the difference among models was limited.

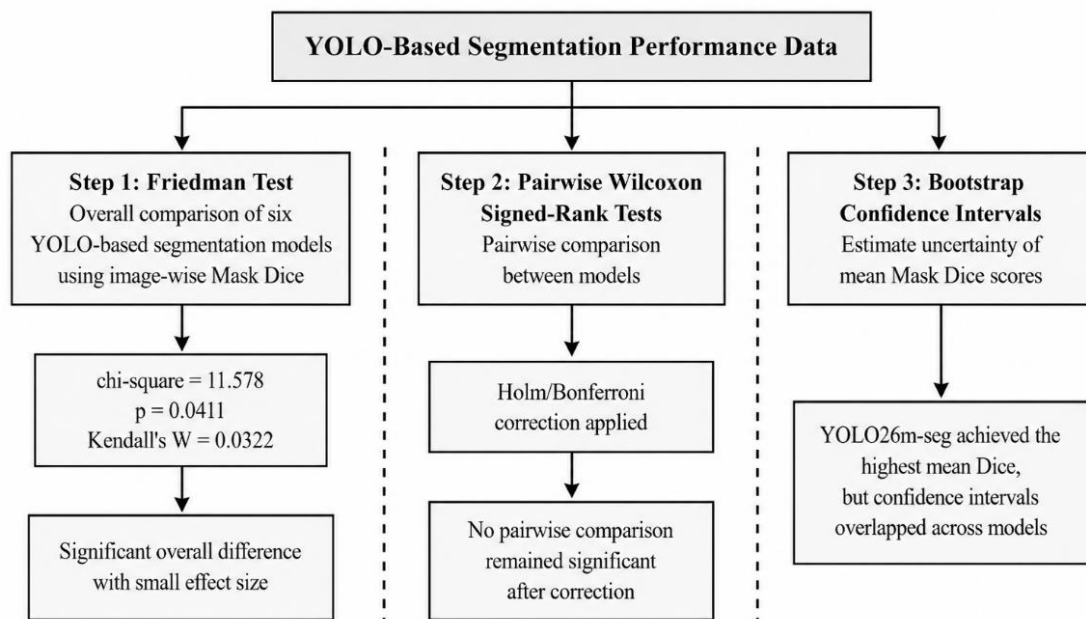


Fig. 8. Workflow of the statistical analysis process for evaluating YOLO-based pavement crack segmentation models.

Following the Friedman test, pairwise Wilcoxon signed-rank tests were performed to compare the models in pairs. Holm and Bonferroni corrections were applied to control the error rate associated with multiple comparisons. After applying these corrections, no pairwise comparison remained statistically significant. This finding indicates that the overall model-level difference detected by the Friedman test was not strong enough to establish a statistically significant pairwise superiority of one individual model over the others.

Bootstrap analysis with 3,000 resampling iterations showed that YOLO26m-seg achieved the highest mean image-wise Mask Dice score. The mean Mask Dice of YOLO26m-seg was 0.7976, with a 95% confidence interval of [0.7632, 0.8251]. Although this model obtained the highest average Dice score, the bootstrap confidence intervals overlapped across the evaluated models, indicating that the differences among models were relatively small in practical terms.

Overall, the statistical analysis supports the selection of YOLO26m-seg as the best-performing model based on mean image-wise Mask Dice and validation Mask mAP@50-95. However, the small Kendall's W value and the non-significant corrected pairwise comparisons suggest that the superiority of YOLO26m-seg should be interpreted conservatively. In practical terms, YOLO26m-seg provided the strongest segmentation performance, while YOLO26s-seg remained a competitive alternative with lower computational cost.

3.5.1. Friedman test

To assess whether the evaluated YOLO-based segmentation models produced statistically different pixel-level crack segmentation performance, a Friedman test was conducted using image-wise Mask Dice scores. The evaluated models included YOLO26n-seg, YOLO26s-seg, YOLO26m-seg, YOLO11n-seg, YOLO11s-seg, and YOLO11m-seg. The dependent variable was the image-wise Mask Dice score, which measures the pixel-level agreement between the predicted crack mask and the corresponding ground-truth mask.

The Friedman test was selected because all six models were evaluated on the same independent test images. Therefore, the observations followed a repeated-measures structure rather than an independent-sample structure. In addition, image-wise crack segmentation scores may not satisfy the normality assumption required for parametric repeated-measures ANOVA. The Friedman test is a non-parametric alternative that is appropriate for comparing multiple related models across the same test samples.

The statistical analysis was performed using 72 test images with complete valid prediction results across all six models. Although the independent test set included 75 images, the statistical analysis was conducted on 72 images for which valid image-wise mask-based metrics were available for all compared models. The null hypothesis was that all evaluated models had the same median image-wise Mask Dice score.

The Friedman test indicated a statistically significant overall difference among the evaluated models, with $\chi^2 = 11.578$ and $p = 0.0411$. Since the p-value was lower than the significance level of 0.05, the null hypothesis was rejected. This result suggests that the six YOLO-based segmentation models did not produce identical image-wise segmentation performance.

However, Kendall's coefficient of concordance was $W = 0.0322$, indicating a small effect size. Therefore, although the overall difference was statistically significant, the magnitude of the performance difference among the models was limited. This means that the statistical significance should be interpreted conservatively and should not be used alone to claim strong practical superiority of one model over all others.

The Friedman test results are summarized in Table 5.

The results indicate that the evaluated models showed a statistically significant overall difference in image-wise Mask Dice performance. However, the small Kendall's W value shows that the effect size was limited. Therefore, while the Friedman test supports the existence of performance variation among the six models, it does not justify a strong claim that one model was practically superior to all others. For this reason, the Friedman test results were interpreted together with validation Mask mAP@50-95, bootstrap confidence intervals, independent test performance, and input-size sensitivity analysis.

Table 5. Friedman test results for image-wise Mask Dice scores of the six YOLO-based segmentation models.

Source	Metric	Number of images	Number of models	χ^2	p-value	Kendall's W	Interpretation
Model type	Mask Dice	72	6	11.578	0.0411	0.0322	Significant overall difference with small effect size

3.5.2. Bootstrap confidence interval analysis

To further evaluate the uncertainty of model performance, bootstrap confidence intervals were calculated for the mean image-wise Mask Dice scores of the six YOLO-based segmentation models. This analysis was used to support model comparison without making strong pairwise superiority claims.

The bootstrap confidence interval analysis was based on image-wise Mask Dice scores obtained from the independent test images. The evaluated models included YOLO26n-seg, YOLO26s-seg, YOLO26m-seg, YOLO11n-seg, YOLO11s-seg, and YOLO11m-seg. For each model, the mean image-wise Mask Dice score and its corresponding 95% bootstrap confidence interval were calculated.

The bootstrap results are summarized in Table 6.

Table 6. Bootstrap confidence intervals for image-wise Mask Dice scores of the evaluated YOLO-based segmentation models.

Rank	Model	Mean Mask Dice	95% Bootstrap CI
1	YOLO26m-seg	0.7976	0.7632 to 0.8251
2	YOLO26s-seg	0.7887	0.7489 to 0.8197
3	YOLO26n-seg	0.7885	0.7593 to 0.8155
4	YOLO11n-seg	0.7884	0.7582 to 0.8140
5	YOLO11s-seg	0.7817	0.7466 to 0.8091
6	YOLO11m-seg	0.7753	0.7348 to 0.8056

As shown in Fig. 9, YOLO26m-seg achieved the highest mean image-wise Mask Dice score, with a mean value of 0.7976 and a 95% confidence interval ranging from 0.7632 to 0.8251. The next closest models were YOLO26s-seg, YOLO26n-seg, and YOLO11n-seg, with mean Dice scores of 0.7887, 0.7885, and 0.7884, respectively.

However, the 95% bootstrap confidence intervals overlapped across the evaluated models. This overlap indicates that the practical differences among the models were limited, despite the highest mean Dice score obtained by YOLO26m-seg. Therefore, the bootstrap analysis supports a conservative interpretation of model performance.

Overall, the bootstrap confidence interval analysis indicates that YOLO26m-seg had the strongest average image-wise segmentation performance among the evaluated models. However, the overlapping confidence intervals show that its advantage over the other models was not large. Therefore, the final model selection was based on combined evidence from validation Mask mAP@50-95, mean image-wise

Mask Dice, independent test performance, and the objective of improving pixel-level crack mask localization.

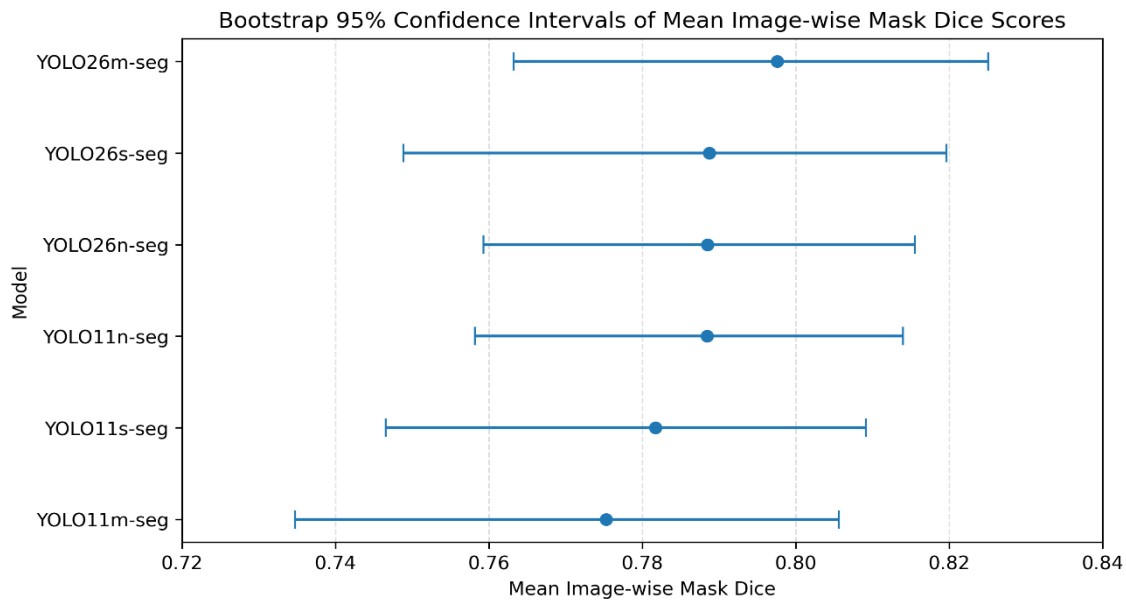


Fig. 9. Bootstrap 95% confidence intervals of mean image-wise Mask Dice scores for the evaluated YOLO-based pavement crack segmentation models.

3.6. Performance summary of the selected YOLO26m-seg model

After comparing the six YOLO-based segmentation models, YOLO26m-seg was selected as the best-performing model based on validation Mask $mAP@50-95$. At an input size of 448×448 pixels, the model achieved the highest validation Mask $mAP@50-95$ among the evaluated configurations. The best validation performance was obtained at epoch 154, with a Mask Precision of 0.7541, Mask Recall of 0.4202, Mask F1-score of 0.5397, Mask $mAP@50$ of 0.4896, and Mask $mAP@50-95$ of 0.2972.

It is worth noting explicitly, and explaining mechanistically rather than simply reporting, that among the six evaluated models YOLO26m-seg combined the highest validation Mask $mAP@50-95$ with the lowest Mask Recall (0.4202). This is not a coincidental pairing: $mAP@50-95$ is computed from the precision-recall curve at each IoU threshold, and a model can achieve a high mAP by producing a smaller number of high-confidence, high-IoU predictions (high precision at the operating points that dominate the area-under-curve calculation) even while leaving a substantial share of true crack pixels undetected altogether (low recall), provided the predictions it does make are precise. YOLO26m-seg's validation Mask Precision (0.7541) alongside its Mask Recall (0.4202) is consistent with exactly this conservative operating characteristic: the model appears to have learned a comparatively high decision threshold or a comparatively confident-only prediction regime, favoring precision over coverage. This has a direct and, for a screening application, non-trivial practical consequence: on a per-pixel basis, YOLO26m-seg misses close to 58% of ground-truth crack pixels ($1 - 0.4202$), meaning that for any individual image, a substantial fraction of the true crack extent may go undetected even when the pixels the model does detect are highly accurate. Because the Crack Area Ratio (Eq. 10) is computed from the predicted mask area, this recall ceiling imposes a corresponding systematic downward bias on CAR: images are more likely to be under-classified in severity than over-classified, since a low-recall model structurally under-counts crack area rather than over-counting it. This directional tendency is more informative than a generic caveat that missed cracks are a concern, and it informs how the severity screening results in Section 3.8 should be read: at the reported operating point, the Low/Moderate/High counts are more likely to under-call than over-call crack extent. For a screening application in which a missed crack carries a real maintenance-oversight risk, this precision–recall trade-off should be weighed alongside the aggregate

mAP-based ranking. A practitioner who prioritizes not missing cracks over minimizing false alarms may prefer a higher-recall operating point, for example YOLO26s-seg (highest validation Mask Recall, Fig. 3(b)) or a lowered confidence threshold on YOLO26m-seg, although the latter would require re-validation of precision. We return to this point in Section 4.2. when discussing the practical implications and limitations of selecting a single “best” model on the basis of Mask mAP@50-95 alone.

The independent test performance of YOLO26m-seg at 448×448 pixels is summarized in Table 7. Although the segmentation framework reports both box-level and mask-level metrics, the interpretation in this study primarily focuses on mask-level performance because the objective was pixel-level pavement crack segmentation.

Table 7. Independent test performance of YOLO26m-seg at 448×448 pixels.

Metric	Value
Box Precision	0.6796
Box Recall	0.7118
Box mAP@50	0.7009
Box mAP@50-95	0.5497
Mask Precision	0.6340
Mask Recall	0.5294
Mask mAP@50	0.5003
Mask mAP@50-95	0.1809

The independent test results show a clear difference between box-level and mask-level performance. The model achieved a Box mAP@50 of 0.7009 and a Box mAP@50-95 of 0.5497, while the corresponding Mask mAP@50 and Mask mAP@50-95 values were 0.5003 and 0.1809, respectively. This gap indicates that the model could localize crack regions more effectively at the bounding-box level than at the pixel-mask level.

This behavior is expected in pavement crack segmentation. Bounding-box detection only requires approximate localization of the crack region, whereas instance segmentation requires accurate pixel-level representation of thin and irregular crack boundaries. Since pavement cracks usually occupy a small fraction of the image, small mask boundary deviations can substantially reduce strict mask-based metrics, especially Mask mAP@50-95.

To examine whether higher input resolution could improve crack mask localization, the selected YOLO26m-seg model was additionally trained and evaluated at an input size of 640×640 pixels. The independent test comparison between the 448×448 and 640×640 configurations is presented in Table 8.

Table 8. Independent test comparison of YOLO26m-seg at 448×448 and 640×640 input sizes.

Metric	YOLO26m-seg 448×448	YOLO26m-seg 640×640	Change
Box Precision	0.6796	0.7149	+0.0353
Box Recall	0.7118	0.6000	-0.1118
Box mAP@50	0.7009	0.6829	-0.0180
Box mAP@50-95	0.5497	0.5284	-0.0213
Mask Precision	0.6340	0.7126	+0.0786
Mask Recall	0.5294	0.5235	-0.0059
Mask mAP@50	0.5003	0.5665	+0.0662
Mask mAP@50-95	0.1809	0.2554	+0.0745

Increasing the input size from 448×448 to 640×640 improved the main mask-level metrics. Mask Precision increased from 0.6340 to 0.7126, Mask mAP@50 increased from 0.5003 to 0.5665, and Mask mAP@50-95 increased from 0.1809 to 0.2554. The absolute improvement in Mask mAP@50-95 was 0.0745, corresponding to a relative increase of approximately 41.2%.

This improvement suggests that the higher input resolution helped preserve crack boundary details and improved mask localization under stricter IoU thresholds. However, the improvement in mask-level metrics was accompanied by a reduction in detection-level recall. Box Recall decreased from 0.7118 to 0.6000, and box-level mAP values also slightly decreased. This indicates a trade-off between detection-level recall and mask-level localization quality.

Since the final crack severity assessment was based on predicted crack mask area, the improvement in mask-level localization was considered more relevant than the reduction in box-level recall. Therefore, YOLO26m-seg at 640×640 pixels was selected as the final model for Crack Area Ratio estimation.

The comparison of key mask-level and image-wise segmentation metrics is shown in Fig. 10.

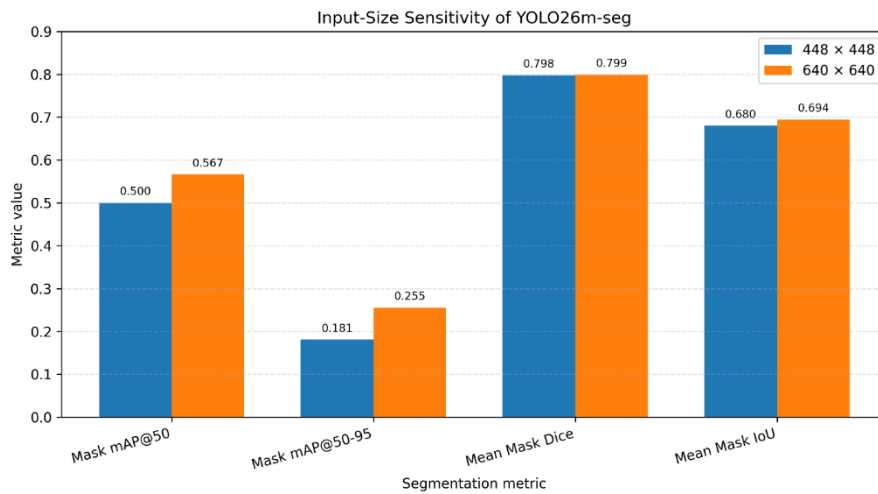


Fig. 10. Input-size sensitivity of YOLO26m-seg based on mask-level and image-wise segmentation metrics.

A paired statistical comparison was also performed using image-wise Mask Dice and Mask IoU scores. The same 72 test images with valid mask-based metric values were evaluated using both input-size configurations. Therefore, the Wilcoxon signed-rank test was applied to compare the paired image-wise results. Bootstrap paired-difference confidence intervals were also calculated to assess the stability of the observed differences. The results are summarized in Table 9.

Table 9. Paired statistical comparison of YOLO26m-seg at 448×448 and 640×640 input sizes.

Metric	n images	Mean at 448×448	Mean at 640×640	Mean difference	Relative change	Wilcoxon p-value	Bootstrap 95% CI
Mask Dice	72	0.7976	0.7992	0.0016	0.2038%	0.0168	-0.0477 to 0.0463
Mask IoU	72	0.6801	0.6943	0.0142	2.094714%	0.0142	-0.0318 to 0.0569

The Wilcoxon signed-rank test indicated statistically significant differences for both Mask Dice and Mask IoU. Mask Dice increased from 0.7976 to 0.7992, while Mask IoU increased from 0.6801 to 0.6943. However, the magnitude of these changes was small. The relative improvement was only 0.20% for Mask Dice and 2.09% for Mask IoU.

In addition, the bootstrap paired-difference confidence intervals included zero for both metrics. This indicates that although the Wilcoxon test detected a statistically significant shift, the improvement was not

strong enough to support a large practical claim based on image-wise Dice and IoU alone. Therefore, the input-size sensitivity results should be interpreted cautiously.

Overall, the 640×640 configuration improved standard mask-level mAP metrics, especially Mask mAP@50-95, but the improvements in image-wise Dice and IoU were limited. Since the final crack severity assessment in this study was based on predicted crack masks, YOLO26m-seg at 640×640 pixels was selected as the final model for Crack Area Ratio estimation. This selection was based on improved mask localization performance rather than a large increase in image-wise Dice similarity.

3.7. Qualitative segmentation results of YOLO26m-seg-640

To further evaluate the visual performance of the selected model, representative qualitative segmentation examples were examined from the independent test set. This analysis was conducted to assess how well YOLO26m-seg-640 could localize pavement cracks at the pixel level and generate crack masks suitable for image-level severity assessment.

Unlike multi-class pavement distress detection studies, the present study focused on a single crack class. Therefore, the qualitative evaluation was not based on class-wise confusion among different distress categories. Instead, it focused on the visual agreement between the predicted crack masks and the visible crack regions in the pavement images.

Representative qualitative segmentation results are shown in Fig. 11. The figure includes three test images representing different crack severity levels based on the predicted Crack Area Ratio (CAR): Low, Moderate, and High. The red overlays indicate the crack regions predicted by the YOLO26m-seg-640 model. The displayed CAR values were used for image-level severity classification.

In Fig. 11(a), the predicted CAR value was 0.0078, which is below the threshold of 0.01 and was therefore classified as Low severity. In Fig. 11(b), the predicted CAR value was 0.0186, which falls within the Moderate severity range of $0.01 \leq \text{CAR} < 0.03$. In Fig. 11(c), the predicted CAR value was 0.0340, which exceeds the High severity threshold of 0.03. These examples demonstrate how the predicted segmentation masks were converted into quantitative crack extent values for severity assessment.

As shown in Fig. 11, the YOLO26m-seg-640 model was able to identify the dominant crack paths and generate visually meaningful crack masks across different severity levels. The model captured the main geometry and spatial extent of the visible crack regions, particularly in the Moderate and High severity examples. This visual behavior is consistent with the quantitative results reported in the previous section, where the 640×640 configuration improved mask-level localization performance.

However, the qualitative results should be interpreted with some caution. Although the main crack regions were generally captured, very thin branches, low-contrast crack segments, and fine boundary details may still be partially missed in some cases. Such boundary-level errors are expected in pavement crack segmentation because cracks are narrow, irregular, and highly sensitive to small pixel-level deviations. This also explains why strict mask-based metrics, especially Mask mAP@50-95, remained lower than the more relaxed Mask mAP@50 metric.

The examples in Fig. 11 were selected to span the Low, Moderate, and High severity range rather than to illustrate failure modes specifically. To complement this and provide a more balanced picture of model limitations, a representative failure case, showing a thin, low-contrast crack pattern that was under-segmented relative to its ground-truth mask, will be added as an additional panel to Fig. 11 in the camera-ready version, once the corresponding held-out prediction is available for inclusion.

Overall, the qualitative analysis supports the suitability of YOLO26m-seg-640 for pixel-level pavement crack localization and Crack Area Ratio-based severity assessment. The presented examples show that the

model can provide interpretable crack masks across different crack extent levels, making it useful as a screening-oriented tool for pavement image analysis.

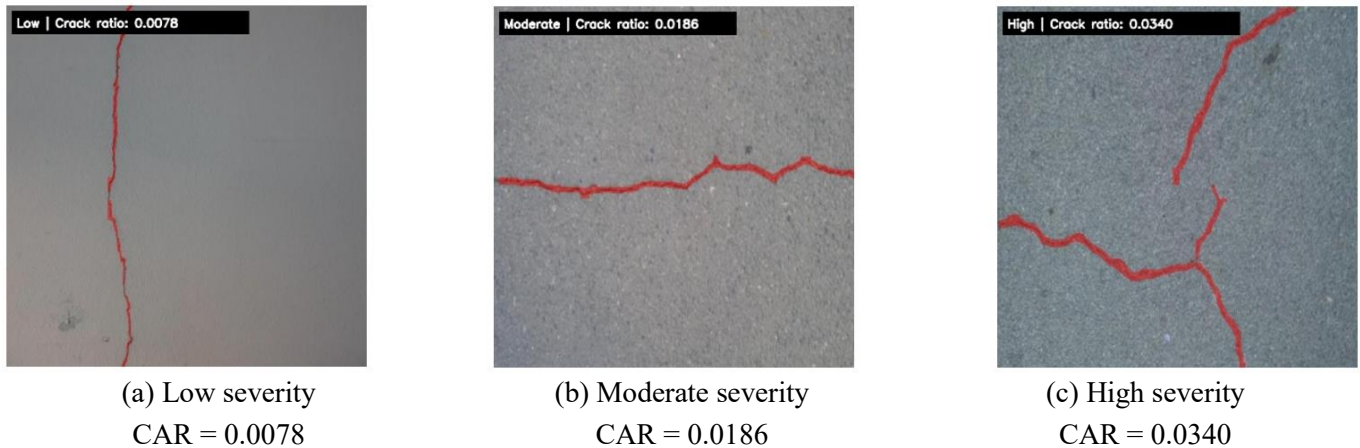


Fig. 11. Representative qualitative segmentation results of YOLO26m-seg-640 on independent test images. The red overlays indicate the predicted crack regions. The displayed Crack Area Ratio (CAR) values were used for image-level severity classification: (a) Low severity, CAR = 0.0078; (b) Moderate severity, CAR = 0.0186; and (c) High severity, CAR = 0.0340.

3.8. Crack severity assessment results

After selecting YOLO26m-seg with an input size of 640×640 pixels as the final segmentation model, crack severity was assessed using the predicted crack mask area. Since the main objective of this study was pixel-level pavement crack segmentation, the severity analysis was based on the proportion of crack pixels detected within each image.

The Crack Area Ratio (CAR), as defined previously, was calculated as follows:

$$\text{CAR} = \text{Predicted Crack Mask Area} / \text{Total Image Area}$$

where the predicted crack mask area represents the number of pixels classified as crack by the YOLO26m-seg-640 model, and the total image area represents the total number of pixels in the image. Therefore, CAR provides an image-level estimate of visible crack extent.

Based on the calculated Crack Area Ratio, each test image was assigned to one of three severity levels: Low, Moderate, or High. The severity thresholds used in this study are presented in Table 10.

Table 10. Crack severity classification thresholds based on Crack Area Ratio.

Severity level	Crack Area Ratio threshold	Interpretation
Low	$\text{CAR} < 0.01$	Limited visible crack area
Moderate	$0.01 \leq \text{CAR} < 0.03$	Moderate visible crack extent
High	$\text{CAR} \geq 0.03$	Extensive visible crack area

It should be noted that the proposed severity classification is not intended to replace standard pavement condition assessment methods such as the Pavement Condition Index. Instead, it provides an image-level crack area index that can be used for rapid screening and maintenance prioritization. The thresholds were used to categorize the relative extent of visible crack regions in the test images.

The severity distribution obtained from the independent test set using YOLO26m-seg-640 is summarized in Table 11.

Table 11. Crack severity distribution based on predicted Crack Area Ratio using YOLO26m-seg-640.

Severity level	Number of images	Percentage
Low	8	10.7%
Moderate	47	62.7%
High	20	26.7%
Total	75	100.0%

The results show that most test images were classified as Moderate severity. Specifically, 47 images, corresponding to 62.7% of the independent test set, were classified as Moderate. In contrast, 20 images, corresponding to 26.7%, were classified as High, and 8 images, corresponding to 10.7%, were classified as Low. This distribution indicates that the test set mainly contained images with moderate visible crack extent, while a smaller proportion of images showed extensive crack areas.

The severity distribution is illustrated in Fig. 12.

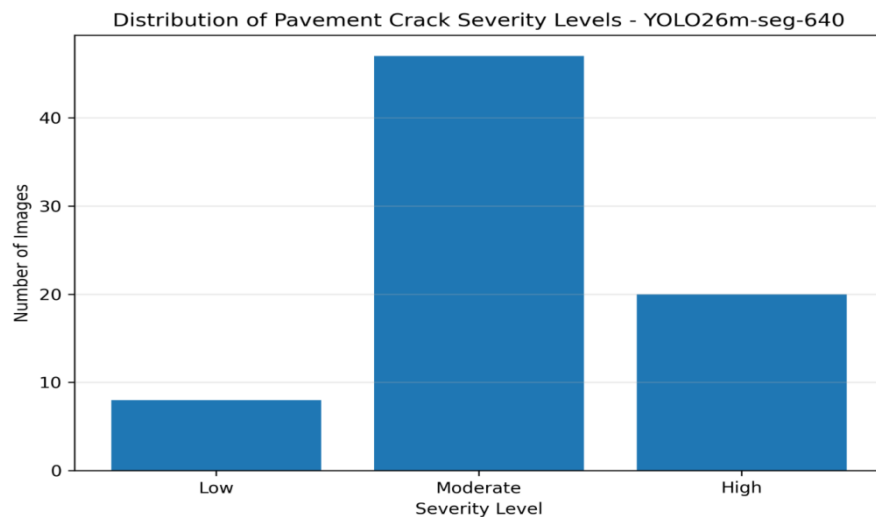


Fig. 12. Crack severity distribution on the independent test set using YOLO26m-seg-640 based on the predicted Crack Area Ratio.

As shown in Fig. 12, the Moderate severity category accounted for the largest portion of the independent test images. This result suggests that the proposed segmentation-based severity assessment can separate images with limited crack extent from those with more extensive visible crack regions. From a pavement management perspective, such an image-level severity distribution may help prioritize pavement images or road sections that require further inspection.

However, the severity levels reported in this study should be interpreted as image-based crack extent categories rather than direct field-based pavement condition ratings. The Crack Area Ratio only quantifies the proportion of predicted crack pixels within each image. It does not account for other distress-related characteristics such as crack width, crack depth, crack type, roughness, raveling, surface deformation, traffic loading, or structural condition. Therefore, the proposed severity assessment should be considered a complementary image-level screening tool rather than a replacement for standard pavement condition surveys.

Overall, the severity assessment results demonstrate that the YOLO26m-seg-640 model can be used not only for pixel-level crack segmentation but also for deriving a simple quantitative crack extent index. This index can support automated pavement image screening and maintenance prioritization, especially when large pavement image datasets need to be reviewed efficiently.

4. Discussion

Although several previous pavement distress studies have reported higher mAP values, the results of this study should be interpreted according to its more restrictive task formulation. Unlike detection-based studies that use bounding boxes, this research focused on pixel-level pavement crack segmentation and severity assessment. Therefore, the reported Mask mAP values reflect a stricter evaluation condition, where predicted crack pixels must closely match the ground-truth crack masks.

Among the six evaluated YOLO-based segmentation models, YOLO26m-seg achieved the best validation performance, with a Mask Precision of 0.7541, Mask Recall of 0.4202, Mask F1-score of 0.5397, Mask mAP@50 of 0.4896, and Mask mAP@50-95 of 0.2972. On the independent test set at 448×448 pixels, the model achieved a Mask mAP@50 of 0.5003 and a Mask mAP@50-95 of 0.1809. The lower strict mask score is mainly related to the thin and irregular geometry of pavement cracks, where small boundary errors can substantially reduce IoU-based metrics.

The difference between box-level and mask-level performance further confirms this challenge. For YOLO26m-seg at 448×448 pixels, Box mAP@50-95 reached 0.5497, while Mask mAP@50-95 was 0.1809. This indicates that the model could localize crack regions more reliably than it could reproduce their exact pixel-level boundaries. This behavior is expected because bounding-box detection is less sensitive to small boundary deviations than instance segmentation.

Increasing the input size from 448×448 to 640×640 improved mask-level localization. Mask mAP@50 increased from 0.5003 to 0.5665, and Mask mAP@50-95 increased from 0.1809 to 0.2554, corresponding to an approximate 41.2% relative improvement in Mask mAP@50-95. However, Box Recall decreased from 0.7118 to 0.6000, indicating a trade-off between detection-level recall and mask-level localization quality. Since the final objective was mask-based Crack Area Ratio estimation, the 640×640 configuration was selected for severity assessment.

The statistical analysis supported a conservative interpretation of model performance. The Friedman test showed a statistically significant overall difference among the six models, with $\chi^2 = 11.578$ and $p = 0.0411$. However, Kendall's W was 0.0322, indicating a small effect size. In addition, the bootstrap confidence intervals for mean Mask Dice overlapped across the evaluated models. Therefore, YOLO26m-seg should be interpreted as the strongest-performing model in this study, rather than as a model with strong practical superiority over all alternatives.

Interpreting these statistics practically requires separating three distinct questions that are easy to conflate under a single p-value: (i) whether the six models are exchangeable in rank (the omnibus null hypothesis tested by the Friedman statistic), (ii) how large the ranking effect is once it is detected (Kendall's W), and (iii) which specific pairs of models can be distinguished after controlling the family-wise error rate across all $C(6,2) = 15$ pairwise comparisons (the Holm-Bonferroni-corrected Wilcoxon tests, Section 3.5). On (i), the Friedman test rejects exchangeability ($\chi^2 = 11.578$, $p = 0.0411$): the six rank vectors observed across the 75 test images are not consistent with all models being drawn from the same underlying performance distribution. On (ii), however, Kendall's W = 0.0322 falls below even Cohen's conventional threshold for a "small" effect (commonly taken as $W \approx 0.10$ in the rank-concordance literature), and closer to the value expected under near-random ranking concordance with only mild systematic structure. Points (i) and (ii) are not in tension: a large sample of paired rankings (here, 75 images $\times$ 6 models) can detect a real but small systematic tendency with a significant omnibus test, while that same tendency is simultaneously too weak, relative to image-to-image noise, to reliably separate any specific pair of models. This is precisely what (iii) confirms: none of the 15 pairwise Wilcoxon comparisons survives Holm-Bonferroni correction, meaning that for every pair of models we cannot reject the hypothesis of equal median performance at the family-wise $\alpha = 0.05$ level. We therefore explicitly decline to interpret

the significant omnibus test as license to claim pairwise superiority for YOLO26m-seg over any specific competitor; doing so would be the classical error of inferring pairwise significance from omnibus significance alone. A further consideration relevant to (ii) is statistical power: with $n = 75$ paired observations, this design has limited sensitivity to detect a true Kendall's W in the small-to-moderate range (a post-hoc calculation, following standard chi-squared power approximations for the Friedman statistic with $k = 6$ groups, indicates that reliably detecting W on the order of 0.05–0.10 at 80% power would require several hundred paired observations); the small-sample regime here can therefore under-detect true small effects as readily as it can flag a marginal one, and this asymmetry is itself a reason for conservative interpretation rather than a reason to dismiss the finding. From a deployment perspective, we recommend that YOLO26m-seg-640 be adopted as the working default on the strength of its top point-estimate validation and test-set mask mAP, while explicitly treating the remaining five architectures, particularly the other YOLO26 and YOLO11 variants at the 'm' and 's' scales, as statistically indistinguishable alternatives under the present evidence rather than as demonstrably inferior options; practitioners with tighter computational budgets may reasonably prefer a smaller-scale variant, since no corrected pairwise test contradicts that choice.

The severity assessment showed that the YOLO26m-seg-640 model classified 8 images as Low severity, 47 images as Moderate severity, and 20 images as High severity. These results demonstrate that the proposed method can provide an image-level estimate of visible crack extent. However, the Crack Area Ratio should not be interpreted as a direct replacement for PCI, because it does not account for crack width, depth, type, roughness, traffic loading, or structural pavement condition.

Regarding the generalization of the CAR thresholds (0.01 and 0.03) to other datasets and real-world scenarios, this is best framed as a covariate-shift problem: CAR is not an intrinsic property of a crack but a function of (crack geometry, imaging geometry, segmentation-model operating point), and generalization fails whenever any of these three covariates shifts between the calibration setting and the deployment setting. First, CAR is defined relative to total image area (Eq. 10), so for a crack of fixed physical width and length, its measured CAR value scales inversely with the imaged ground area, which is itself set by camera-to-pavement distance, lens field of view, and image cropping; a deployment with, for example, twice the ground sampling distance per pixel would roughly halve the apparent CAR for the same physical crack, silently shifting every image across the 0.01/0.03 boundaries without any change in actual pavement condition. This is a first-order, purely geometric effect, unrelated to model quality, and is the covariate shift most likely to invalidate direct threshold reuse. Second, because CAR is computed from the predicted, not ground-truth, mask, it inherits the operating point of whichever segmentation model produces it; Section 3.6 shows that our own selected model operates in a precision-favoring, recall-limited regime, which imposes a systematic downward bias on CAR (Section 3.6). A different model, or the same architecture retrained on a different, possibly larger or differently annotated dataset, would be expected to sit at a different point on the precision-recall trade-off curve and would therefore induce a different systematic bias, again shifting measured CAR relative to the calibration setting independent of true crack severity. Because both of these shift mechanisms operate multiplicatively on the same underlying quantity, their effects do not average out or cancel across a deployment; they compound. For these reasons, we view the present thresholds as valid only within the specific (imaging geometry, model operating point) regime characterized in this study, and we recommend that any new deployment context either hold both covariates fixed (same camera setup, same trained model) or explicitly re-derive the thresholds via the ROC/Youden's-J-based calibration procedure described in Section 4.2, using a small set of paired field PCI ratings collected under the new imaging and model conditions, rather than assuming the 0.01/0.03 cutoffs transfer directly.

Overall, this study presents a practical segmentation-based framework for pavement crack analysis using YOLO models. The results show that higher input resolution can improve mask localization, while

statistical testing indicates that model differences should be interpreted cautiously. Future research should evaluate larger and more diverse datasets, improve crack boundary annotations, use boundary-aware or Dice-based loss functions, and validate the severity assessment method using field-calibrated pavement condition data.

4.1. Contextualizing the segmentation accuracy gap with prior studies

The mask-level accuracy obtained in this study was lower than the detection-level accuracy reported in several previous YOLO-based pavement distress studies. However, this difference should be interpreted carefully because the present study focused on pixel-level crack segmentation, whereas many previous works evaluated pavement distress using bounding-box object detection. In object detection, a model is required to localize the approximate region of a distress using a rectangular box. In contrast, segmentation requires the model to identify the crack region at the pixel level. Therefore, Mask mAP is a stricter metric than detection mAP, particularly for narrow pavement cracks.

In the present study, YOLO26m-seg at 448×448 pixels achieved a Mask mAP@50 of 0.5003 and a Mask mAP@50-95 of 0.1809 on the independent test set. After increasing the input size to 640×640 pixels, Mask mAP@50 improved to 0.5665, and Mask mAP@50-95 increased to 0.2554. Although these values are lower than several detection-based pavement distress studies, they reflect a more demanding pixel-level segmentation task.

The comparison between this study and selected prior studies is summarized in Table 12. The table is intended to provide contextual interpretation rather than a direct ranking, because the compared studies differ in task type, dataset characteristics, imaging conditions, annotation format, and evaluation metrics.

The lower strict mask-level score in this study can be explained by several factors. First, pavement cracks are thin, irregular, discontinuous, and often occupy only a small portion of the image. After resizing, the crack width may correspond to only a few pixels. Therefore, small boundary errors can substantially reduce mask IoU and Mask mAP@50-95. This problem is less severe in bounding-box object detection, where the model can still obtain a correct detection even if the bounding box contains background pixels around the crack.

This resolution effect can be quantified more precisely than a qualitative appeal to “loss of detail.” The original CRACK500 images were captured at approximately 2000×1500 pixels; resizing to 448×448 applies a linear downsampling factor of roughly $4.5\times$ along the longer dimension ($2000/448 \approx 4.46$), so a crack that spans, say, 10-15 pixels in width in the source image, itself already a narrow feature relative to typical pavement defects, is compressed to approximately 2-3 pixels after resizing, and cracks narrower than about 4-5 source pixels are compressed below a stable, reliably resolvable width. This is directly analogous to a sampling-theorem argument: a linear feature whose width approaches the pixel pitch of the resized grid is sampled at or below the Nyquist limit for that feature, so its boundary can no longer be localized with sub-pixel confidence, and small stochastic variations in how the resampling kernel interacts with the crack’s exact sub-pixel position can shift or partially erase the feature entirely. This is not a hypothetical concern specific to this dataset: it is the generic reason why thin, elongated defects (cracks, wires, hairline fractures) are systematically harder for CNN-based segmentation than compact, well-sampled objects of the same pixel area, and it is well documented in the crack-segmentation literature as a primary driver of the gap between box-level and mask-level performance on thin-object classes. The input-size sensitivity experiment reported later in this section provides direct, if partial, empirical support for this mechanism: increasing the input resolution to 640×640 reduces the effective downsampling factor to roughly $3.1\times$ ($2000/640 \approx 3.13$), meaningfully raising the pixel width of the same source crack, and produced the largest single improvement in Mask mAP@50-95 observed anywhere in this study (approximately 41% relative improvement).

Table 12. Comparative summary of this study and selected pavement crack or distress detection studies.

Study	Model / Method	Dataset and image setting	Task type	Main output	Main reported metric	Key finding
This study	YOLO26m-seg	500 CRACK500 images, 1 crack class	Instance segmentation	Crack mask	Mask mAP@50 = 0.5665; Mask mAP@50-95 = 0.2554 at 640×640	Improved mask localization using higher input resolution; used for Crack Area Ratio estimation
This study	YOLO26m-seg	500 CRACK500 images, 1 crack class	Instance segmentation	Crack mask	Mask mAP@50 = 0.5003; Mask mAP@50-95 = 0.1809 at 448×448	Selected as the best validation model among the six evaluated YOLO-based segmentation models
Mohamed et al. [29]	YOLOv8l	500 high-resolution UAV images, 7 pavement distress classes	Object detection	Bounding box	Detection mAP@50-95 $\approx$ 0.381	Best detection accuracy among evaluated YOLOv8 variants in the reference JRCE study
Mohamed et al. [29]	YOLOv8m	500 high-resolution UAV images, 7 pavement distress classes	Object detection	Bounding box	Detection mAP@50-95 $\approx$ 0.344	Reported as a balanced detection model considering accuracy and computational efficiency
Maeda et al. [6]	Deep neural networks	Smartphone road images	Road damage detection and classification	Damage class / region	Detection and classification metrics	Demonstrated the feasibility of deep learning for road damage recognition using mobile imagery
Arya et al. [7]	Deep learning-based road damage detection	Multi-country road damage images	Road damage detection and classification	Damage class / region	Detection and classification metrics	Showed that dataset diversity improves generalization across countries
DeepCrack [9]	Hierarchical convolutional features	Crack image datasets	Crack segmentation	Crack map	Pixel-level crack segmentation metrics	Demonstrated the importance of hierarchical features for crack segmentation
Lau et al. [12]	U-Net-based CNN	Pavement crack images	Crack segmentation	Crack mask	Pixel-level segmentation metrics	Confirmed the effectiveness of encoder-decoder networks for pavement crack segmentation

This dose-response pattern, more effective sampling resolution for the same underlying crack geometry yielding a large gain concentrated in the strict, boundary-sensitive mAP metric, is consistent with sub-Nyquist sampling of crack width being a genuine, mechanistically identifiable contributing factor to the low absolute Mask mAP@50-95 values (0.18-0.26) observed at 448×448 , rather than merely a correlational coincidence. We do not, however, claim that resolution loss is the sole explanation, since the other factors discussed below (task formulation, dataset scope, annotation uncertainty) also contribute independently, and a fully controlled test of the sampling-theorem account (e.g., systematically varying resolution while holding annotation protocol and dataset composition fixed, and directly measuring crack width in pixels at each resolution) is identified as a natural extension for future work.

Second, the task formulation in this study was more restrictive than many detection-based studies. The objective was not only to detect the presence of pavement cracks but also to generate pixel-level crack masks for subsequent severity assessment. Since the Crack Area Ratio was computed directly from

predicted crack pixels, mask quality was more important than approximate object localization. This explains why the interpretation focused on mask-level metrics rather than box-level metrics.

Third, the CRACK500 dataset used in this study contained 500 images and one crack class. Although this dataset is suitable for crack segmentation benchmarking, it does not include field-calibrated information such as crack width, crack depth, pavement age, traffic loading, roughness, or structural condition. In addition, pixel-level annotations of narrow cracks are inherently sensitive to labeling uncertainty. Small inconsistencies in ground-truth masks can affect strict IoU-based evaluation metrics.

Fourth, the comparison with the reference JRCE study should be interpreted cautiously, and the reason is a structural non-equivalence between the two evaluation protocols rather than a difference in model quality. In IoU-based evaluation, a bounding box is scored correct once it overlaps the ground-truth box by the IoU threshold, and because a rectangular box necessarily includes background pixels around an elongated, irregular object such as a crack, the achievable IoU for a well-placed box is bounded above by the ratio of the crack's true pixel area to its enclosing rectangle's area, a ratio that is typically large (often >0.7 - 0.8) for compact distress types such as potholes or patches, but is characteristically small (often well under 0.3 - 0.4) for a thin, sinuous crack, since a rectangle circumscribing a diagonal or branching crack is mostly background. Mask-level IoU has no such ceiling effect: it is computed directly between the predicted and ground-truth pixel sets, so the metric penalizes exactly the boundary and coverage errors that box-level IoU structurally cannot see. This means detection mAP and Mask mAP are not merely reported on different scales, they are answering different geometric questions, and a reported detection mAP@50-95 of 0.381 for seven distress classes (including compact classes with a favorable box-to-object area ratio) is not a valid benchmark for a Mask mAP@50-95 on a single, elongated, thin-object class. The class-count difference (seven versus one) works, if anything, in the opposite direction from what a naive reading would suggest: additional distress classes in the reference study include geometrically compact defect types whose favorable box-IoU ceiling would tend to raise, not lower, an aggregate multi-class detection mAP relative to a single thin-crack class, making the present study's lower absolute score consistent with, rather than contradictory to, the geometric argument above. The reported values in Table 12 should therefore be used only to contextualize the relative difficulty of the segmentation task, not as a like-for-like performance ranking.

The input-size experiment supports this interpretation. Increasing the input size from 448×448 to 640×640 improved Mask mAP@50 from 0.5003 to 0.5665 and Mask mAP@50-95 from 0.1809 to 0.2554. The relative improvement in Mask mAP@50-95 was approximately 41.2%. This result suggests that higher input resolution can help preserve crack boundary details and improve mask localization. However, the improvement in image-wise Dice and IoU was limited, indicating that the effect should be interpreted conservatively.

This apparent tension, a large ($\sim 41\%$) relative improvement in Mask mAP@50-95 alongside a paired, image-wise Wilcoxon improvement of only 0.20% in Mask Dice (Section 3.6, Table 9), is a genuine measurement artifact of how the two metrics are constructed, not a contradiction in the underlying data, and is worth reconciling at the level of the metric definitions rather than as two loosely compatible numbers. Mask mAP@50-95 is computed as the mean, over ten IoU thresholds (0.50 to 0.95 in steps of 0.05), of the area under the precision-recall curve; each threshold imposes a binary pass/fail cutoff on every predicted-mask/ground-truth-mask pair, so predictions sitting near a given threshold can flip from a hit to a miss (or vice versa) from a small change in boundary overlap. This makes mAP@50-95 a threshold-sweep statistic that is disproportionately sensitive to boundary-marginal cases, i.e., cases whose IoU with the ground truth sits close to one of the ten cutoffs, because it is precisely these cases whose classification flips as resolution improves and boundary precision increases. Mask Dice, by contrast, is a threshold-free, continuous overlap coefficient ($2 \times |A \cap B| / (|A| + |B|)$) computed once per image and averaged (or, in the paired test, differenced per image) without any pass/fail discretization; it summarizes

the typical degree of overlap rather than counting how many predictions cross a set of hard cutoffs. A resolution increase that shifts a meaningful subset of boundary-marginal predictions from just-below to just-above several IoU thresholds can therefore produce a large, multiplicative improvement in the threshold-swept mAP score while leaving the continuous, per-image Dice overlap for the already-well-segmented majority of images nearly unchanged, exactly the asymmetry observed here. This is not an ad hoc explanation constructed post hoc to reconcile an inconvenient result: it follows directly from the mathematical definitions of the two metrics, and the same asymmetry between threshold-swept and continuous overlap metrics is well documented in the object-detection and segmentation evaluation literature. Given this, the two results are best read jointly, not as competing evidence: the mAP result indicates that resolution disproportionately rescues a subset of boundary-marginal, likely thin or low-contrast crack instances, which is exactly the population of interest for reliable Crack Area Ratio estimation (Eq. 10), because CAR is sensitive to whether these marginal pixels are captured at all, not to the average overlap quality of already-well-segmented cracks. On this basis, we selected the 640×640 configuration for downstream CAR estimation, judging the mAP-level evidence of improved boundary-marginal recovery as more relevant to the estimation objective than the modest average Dice change or the accompanying Box Recall decrease, the latter of which reflects detection completeness rather than the mask-area quantity CAR depends on. We present this as a metric-grounded, mechanistically justified choice rather than an arbitrary preference for whichever metric looked more favorable, while still reporting the Dice/IoU and Box Recall figures in full (Section 3.6, Tables 7–9)) so that readers who prioritize typical-case overlap or detection completeness over boundary-marginal recovery can weigh the trade-off differently.

The severity assessment component also has limitations. The proposed Crack Area Ratio provides an image-level estimate of visible crack extent, but it is not equivalent to standard pavement condition indices such as PCI. CAR does not consider crack width, depth, crack type, roughness, raveling, surface deformation, traffic loading, or structural deterioration. Therefore, the proposed severity categories should be considered a screening and prioritization tool rather than a replacement for field-based pavement condition surveys.

Future research should address these limitations by using larger and more diverse crack segmentation datasets, improving annotation consistency, and validating the proposed approach on external pavement image sources. Higher-resolution training, boundary-aware loss functions, Dice-based loss functions, focal loss, and crack-specific data augmentation may further improve pixel-level crack localization. In addition, integrating image-based Crack Area Ratio with field measurements such as crack width and pavement condition survey data could improve the reliability of automated severity assessment.

Overall, the observed accuracy gap is mainly related to the stricter nature of pixel-level crack segmentation, the narrow geometry of pavement cracks, dataset limitations, and the difference between image-based crack extent estimation and field-based pavement condition rating. Despite these limitations, the YOLO26m-seg-640 model improved mask localization compared with the 448×448 configuration and provided a practical basis for automated crack area estimation and severity screening.

4.2. Study limitations

This study has several limitations that should be considered when interpreting its results and when assessing the generalizability of the proposed framework. First, all experiments rely entirely on the secondary, pre-existing CRACK500 dataset rather than on primary field data collected by the authors. Because CRACK500 images were captured with mobile phones on a single university campus, primary-collection challenges that are typical of real-world pavement inspection, such as variable lighting, camera angle, vehicle-mounted motion blur, and region-specific pavement textures and materials, are not represented in the training or evaluation data. As a direct consequence, no cross-dataset or external-field

validation was performed in this study; the reported performance and the proposed CAR severity thresholds should therefore be regarded as specific to the CRACK500 imaging conditions until validated on independent pavement image sources.

Second, the dataset is limited in size (500 images, split into 350/75/75), and it is worth quantifying, rather than only asserting, how this constrains the statistical power of the repeated-measures analysis in Section 2.4. For a Friedman test with $k = 6$ related groups (models) and n paired blocks (images), the test statistic is asymptotically chi-squared with $(k-1) = 5$ degrees of freedom, and its non-centrality, and hence power to detect a given true Kendall's W , scales approximately with $n \times (k-1) \times W$. Using this approximation, detecting a true W on the order of 0.05–0.10 (i.e., a genuinely small but real ranking effect, by conventional rank-concordance benchmarks) at the standard 80% power and $\alpha = 0.05$ would require on the order of several hundred paired image observations, roughly $4\text{--}8 \times$ the present $n = 75$. Our test set is therefore adequately powered to detect the kind of effect we in fact detected, a small but non-negligible $W = 0.0322$ yielding a significant omnibus result, but is under-powered to reliably resolve which specific pairs of models drive that effect, which is exactly the pattern observed: a significant Friedman test alongside overlapping bootstrap confidence intervals (Section 3.5) and no pairwise comparison surviving Holm-Bonferroni correction. This is the correct, symmetric reading of an under-powered design: it is a reason to interpret the null pairwise results cautiously in both directions, neither as proof that the six models are difficult to separate with confidence at this sample size, nor as proof that they perform identically, since the same power limitation that prevents confident separation also prevents confidently ruling out a true but small pairwise difference. Scaling the test set by the $4\text{--}8 \times$ factor identified above, ideally via a larger or pooled cross-dataset benchmark, is identified as the direct, quantitatively motivated remedy and is listed explicitly as future work.

Third, this study measured training duration (Section 3.4) but did not benchmark inference-time throughput (e.g., frames per second or per-image latency) at deployment. The Google Colab T4 environment used for training is a shared, non-dedicated platform and was judged unsuitable for producing a reliable, reproducible throughput benchmark; a fair inference-speed comparison requires a fixed, dedicated hardware target (e.g., a specific edge device or on-premises GPU) representative of the intended field-deployment scenario. We therefore do not report FPS figures rather than provide numbers that could be misleading, and we list dedicated inference-speed benchmarking as a priority for future work.

Fourth, the CAR severity thresholds (Section 2.5) were defined heuristically, as round-number cutoffs on the proportion of image area occupied by predicted crack pixels, and were not derived from a sensitivity analysis, an ROC-style optimization, or validation against field-measured PCI severity ratings. It is worth being precise about what a properly calibrated alternative would require, both to justify why it was outside this study's scope and to give a concrete target for future work. A defensible threshold-setting procedure would treat this as a classification-calibration problem: given a set of images with paired (CAR, field-PCI-severity) observations, one would construct the ROC curve for each candidate CAR cutoff against the field-determined severity label, select the operating point maximizing an appropriate criterion (e.g., Youden's J statistic, $J = \text{sensitivity} + \text{specificity} - 1$, if false negatives and false positives are considered equally costly, or a cost-weighted variant if, as is likely for a maintenance-screening application, missed high-severity cracks are more costly than false alarms), and report the resulting sensitivity/specificity together with confidence intervals reflecting the calibration sample size. None of these ingredients, paired field-PCI ground truth, a defined cost asymmetry between missed and false-flagged cracks, or a held-out calibration sample, were available within the scope of the present study, which relies on CRACK500's image-level annotations without accompanying field condition surveys. In their absence, the present thresholds (0.01 and 0.03) should be understood strictly as an internally consistent, reproducible operational convention that allows this study's own test images to be partitioned

and discussed consistently, not as calibrated, field-validated cutoffs, and no claim of clinical- or engineering-grade diagnostic accuracy should be inferred from the Low/Moderate/High labels used here. Establishing the ROC-based calibration described above against paired PCI field data is identified as necessary, well-specified future work before the thresholds are used for maintenance decision-making.

Fifth, the selected YOLO26m-seg model, while achieving the highest validation Mask mAP@50-95, also exhibited the lowest Mask Recall among the six evaluated models (Section 3.1, Fig. 3(b)), meaning that a substantial share of crack pixels are missed on a per-image basis. For a screening application in which missed cracks carry a real maintenance-oversight cost, this precision-oriented behavior is an important practical caveat; we discuss this trade-off explicitly, rather than only reporting the aggregate mAP-based ranking, so that practitioners can weigh recall alongside overall localization accuracy when selecting an operating model for deployment.

Finally, the training configuration (input size, optimizer, learning rate, batch size, and patience; see Section 2.3) was fixed across all six models based on standard practice and the available GPU memory, rather than obtained from a dedicated hyperparameter search for each architecture; it is possible that a model-specific search would shift the relative ranking among models, and this is noted as a further avenue for future work.

5. Conclusions

This study developed and evaluated a YOLO-based framework for pixel-level pavement crack segmentation and image-based crack extent assessment, using the CRACK500 dataset and the training/validation/test split described in Section 2.1. Six YOLO-based segmentation models, spanning two YOLO generations and three model scales, were trained and compared under one unified experimental configuration.

Among the six models, YOLO26m-seg achieved the best validation Mask mAP@50-95 (full validation metrics are reported in Section 3.6) and was accordingly selected for independent test evaluation. On the independent test set, mask-level performance was consistently lower than box-level performance (Section 3.6 (Table 7)), confirming that pixel-level crack segmentation is a stricter and more demanding task than bounding-box localization, mainly because pavement cracks are thin, irregular, and sensitive to small boundary errors. Increasing the input resolution from 448×448 to 640×640 produced a clear improvement in the main mask-level metrics (Section 3.6 (Table 8)), at the cost of a reduction in Box Recall (Section 3.6 (Table 8)), and the higher-resolution configuration, YOLO26m-seg-640, was therefore selected for Crack Area Ratio estimation.

The repeated-measures statistical analysis (Section 3.5) found a statistically significant overall difference among the six models, but with a small effect size and overlapping bootstrap confidence intervals. Consistent with this evidence, and as discussed in Section 4, YOLO26m-seg should be interpreted as the strongest-performing model observed in this specific experiment rather than as a model with demonstrated, generalizable superiority; the practical distinction between the evaluated architectures should be regarded as modest given the available sample size.

The predicted crack masks from YOLO26m-seg-640 were converted into a Crack Area Ratio (CAR) and used to classify the independent test images into Low, Moderate, and High crack-extent categories (severity counts are reported in Section 3.8), demonstrating that the framework can provide an automated, reproducible, image-level estimate of visible crack extent suitable for preliminary screening. As emphasized throughout Section 4, CAR is a mask-derived proxy and should not be interpreted as a direct substitute for the Pavement Condition Index, since it does not incorporate crack width, depth, type, roughness, traffic loading, or structural condition; it is best regarded as a complementary, low-cost screening index that can help prioritize images or road sections for more detailed field inspection.

Overall, the results indicate that YOLO-based instance segmentation can support automated pavement crack segmentation and image-level crack-extent screening, while the statistical evidence supports a conservative, rather than a strongly deterministic, interpretation of model differences. The limitations discussed in Section 4.2, namely the reliance on a single secondary dataset, the absence of cross-dataset and inference-speed benchmarking, and the heuristic (not field-calibrated) severity thresholds, define the scope within which these conclusions should be read. Future research should extend this framework to larger and more diverse pavement crack datasets collected under real-world field conditions, benchmark inference-time throughput on deployment-representative hardware, investigate boundary-aware or Dice-based loss functions to improve thin-crack localization, and validate the CAR severity thresholds against field-calibrated PCI data.

Authors contribution statement

Gholamali Shafabakhsh: Methodology, Project administration, Resources, Supervision, Validation, Writing – review & editing.

Sina Jabbari: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflicts of interest

The author (Gh. Shafabakhsh) is an Editorial Board Member for Journal of Rehabilitation in Civil Engineering and was not involved in the editorial review or the decision to publish this article.

References

- [1] Shahin MY. Pavement Management for Airports, Roads, and Parking Lots. Boston, MA: Springer US; 2005. <https://doi.org/10.1007/b101538>.
- [2] ASTM International. Practice for Roads and Parking Lots Pavement Condition Index Surveys 2020. <https://doi.org/10.1520/D6433-20>.
- [3] Koch C, Georgieva K, Kasireddy V, Akinci B, Fieguth P. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Adv Eng Informatics* 2015;29:196–210. <https://doi.org/10.1016/j.aei.2015.01.008>.
- [4] Sony S, Laventure S, Sadhu A. A literature review of next-generation smart sensing technology in structural health monitoring. *Struct Control Heal Monit* 2019;26:e2321. <https://doi.org/10.1002/stc.2321>.
- [5] Dong C-Z, Catbas FN. A review of computer vision-based structural health monitoring at local and global levels. *Struct Heal Monit* 2021;20:692–743. <https://doi.org/10.1177/1475921720935585>.
- [6] Maeda H, Sekimoto Y, Seto T, Kashiyaama T, Omata H. Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images. *Comput Civ Infrastruct Eng* 2018;33:1127–41. <https://doi.org/10.1111/mice.12387>.
- [7] Arya D, Maeda H, Ghosh SK, Toshniwal D, Mraz A, Kashiyaama T, et al. Deep learning-based road damage detection and classification for multiple countries. *Autom Constr* 2021;132:103935. <https://doi.org/10.1016/j.autcon.2021.103935>.
- [8] Cha Y, Choi W, Büyüköztürk O. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Comput Civ Infrastruct Eng* 2017;32:361–78. <https://doi.org/10.1111/mice.12263>.

- [9] Zou Q, Zhang Z, Li Q, Qi X, Wang Q, Wang S. DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection. *IEEE Trans Image Process* 2019;28:1498–512. <https://doi.org/10.1109/TIP.2018.2878966>.
- [10] Yang F, Zhang L, Yu S, Prokhorov D, Mei X, Ling H. Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. *IEEE Trans Intell Transp Syst* 2020;21:1525–35. <https://doi.org/10.1109/TITS.2019.2910595>.
- [11] Fan Z, Wu Y, Lu J, Li W. Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Neural Network. *ArXiv Prepr* 2018. <https://doi.org/10.48550/arXiv.1802.02208>.
- [12] Lau SLH, Chong EKP, Yang X, Wang X. Automated Pavement Crack Segmentation Using U-Net-Based Convolutional Neural Network. *IEEE Access* 2020;8:114892–9. <https://doi.org/10.1109/ACCESS.2020.3003638>.
- [13] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9351, 2015, p. 234–41. https://doi.org/10.1007/978-3-319-24574-4_28.
- [14] He K, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. *2017 IEEE Int. Conf. Comput. Vis., IEEE; 2017*, p. 2980–8. <https://doi.org/10.1109/ICCV.2017.322>.
- [15] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. *2015 IEEE Conf. Comput. Vis. Pattern Recognit., vol. 07-12- June, IEEE; 2015*, p. 3431–40. <https://doi.org/10.1109/CVPR.2015.7298965>.
- [16] Shi Y, Cui L, Qi Z, Meng F, Chen Z. Automatic Road Crack Detection Using Random Structured Forests. *IEEE Trans Intell Transp Syst* 2016;17:3434–45. <https://doi.org/10.1109/TITS.2016.2552248>.
- [17] Zhang L, Yang F, Daniel Zhang Y, Zhu YJ. Road crack detection using deep convolutional neural network. *2016 IEEE Int. Conf. Image Process., vol. 2016- Augus, IEEE; 2016*, p. 3708–12. <https://doi.org/10.1109/ICIP.2016.7533052>.
- [18] Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conf. Comput. Vis. Pattern Recognit., vol. 2016- Decem, IEEE; 2016*, p. 779–88. <https://doi.org/10.1109/CVPR.2016.91>.
- [19] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger. *2017 IEEE Conf. Comput. Vis. Pattern Recognit., vol. 2017- Janua, IEEE; 2017*, p. 6517–25. <https://doi.org/10.1109/CVPR.2017.690>.
- [20] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement. *ArXiv Prepr* 2018. <https://doi.org/10.48550/arXiv.1804.02767>.
- [21] Bochkovskiy A, Wang C-Y, Liao H-YM. YOLOv4: Optimal Speed and Accuracy of Object Detection. *ArXiv Prepr ArXiv200410934* 2020. <https://doi.org/10.48550/arXiv.2004.10934>.
- [22] Wang C-Y, Bochkovskiy A, Liao H-YM. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *2023 IEEE/CVF Conf Comput Vis Pattern Recognit 2022;2023-June:7464–75*. <https://doi.org/10.1109/CVPR52729.2023.00721>.
- [23] Jocher G, Chaurasia A, Qiu J. Ultralytics YOLO. *Ultralytics* 2023.
- [24] Ultralytics. Ultralytics YOLO11 Documentation. *Ultralytics Docs* 2024.
- [25] Jocher G, Qiu J, Liu M, Lyu S, Akyon FC, Kalfaoglu ME. Ultralytics YOLO26: Unified Real-Time End-to-End Vision Models. *ArXiv Prepr ArXiv260603748* 2026. <https://doi.org/10.48550/arXiv.2606.03748>.
- [26] Fan L, Tang S, Ariffin MKA b. M, Ismail MIS, Wang X. Impact of Image Preprocessing and Crack Type Distribution on YOLOv8-Based Road Crack Detection. *Sensors* 2025;25:2180. <https://doi.org/10.3390/s25072180>.
- [27] Mohamed ME, Esmat AA, Hamdy AM, El Sheshtawy AM. Assessment of Deep Learning Models for Pavement Distress Detection in High-Resolution UAV Imagery. *J Rehabil Civ Eng* 2026;14:1–23. <https://doi.org/10.22075/jrce.2025.2415>.
- [28] Fakharian P, Naderpour H. Damage Severity Quantification Using Wavelet Packet Transform and Peak Picking Method. *Pract Period Struct Des Constr* 2022;27:04021063. [https://doi.org/10.1061/\(ASCE\)SC.1943-5576.0000639](https://doi.org/10.1061/(ASCE)SC.1943-5576.0000639).

- [29] Soleimani Nezhad S, Khademian F, Naderpour H, Kalantari SM, Fakharian P. Signal processing-based damage detection of steel braced frame subjected to consequent excitations. *Innov Infrastruct Solut* 2024;9:454. <https://doi.org/10.1007/s41062-024-01762-5>.
- [30] Nouri Y, Shahabian F, Shariatmadar H, Entezami A. Structural Damage Detection in the Wooden Bridge Using the Fourier Decomposition, Time Series Modeling and Machine Learning Methods. *J Soft Comput Civ Eng* 2024;8:83–101. <https://doi.org/10.22115/scce.2023.401971.1669>.
- [31] Lin T-Y, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, et al. Microsoft COCO: Common Objects in Context. *Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)* 2015;8693 LNCS:740–55. https://doi.org/10.1007/978-3-319-10602-1_48.
- [32] Dice LR. Measures of the Amount of Ecologic Association Between Species. *Ecology* 1945;26:297–302. <https://doi.org/10.2307/1932409>.
- [33] Jaccard P. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bull La Société Vaudoise Des Sci Nat* 1901;37:547–79.
- [34] Friedman M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J Am Stat Assoc* 1937;32:675. <https://doi.org/10.2307/2279372>.
- [35] Wilcoxon F. Individual Comparisons by Ranking Methods. *Biometrics Bull* 1945;1:80. <https://doi.org/10.2307/3001968>.
- [36] Rezatofighi H, Tsoi N, Gwak J, Sadeghian A, Reid I, Savarese S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. 2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit., vol. 2019- June, IEEE; 2019, p. 658–66. <https://doi.org/10.1109/CVPR.2019.00075>.
- [37] Powers DMW. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *J Mach Learn Technol* 2020;2:37–63. <https://doi.org/2010.16061>.
- [38] Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. The Pascal Visual Object Classes (VOC) Challenge. *Int J Comput Vis* 2010;88:303–38. <https://doi.org/10.1007/s11263-009-0275-4>.
- [39] Holm S. A Simple Sequentially Rejective Multiple Test Procedure. *Scand J Stat* 1979;6:65–70. <https://doi.org/10.2307/4615733>.
- [40] Efron B, Tibshirani RJ. An Introduction to the Bootstrap. Chapman and Hall/CRC; 1994. <https://doi.org/10.1201/9780429246593>.
- [41] Ha J, Kim D, Kim M. Assessing severity of road cracks using deep learning-based segmentation and detection. *J Supercomput* 2022;78:17721–35. <https://doi.org/10.1007/s11227-022-04560-x>.
- [42] Owor NJ, Du H, Daud A, Aboah A, Adu-Gyamfi Y. Image2PCI -- A Multitask Learning Framework for Estimating Pavement Condition Indices Directly from Images. *ArXiv Prepr ArXiv231008538* 2023. <https://doi.org/10.48550/arXiv.2310.08538>.