

RESEARCH

Open Access



Crack SAM: enhancing crack detection utilizing foundation models and Detectron2 architecture

R Rakshitha^{1*}, S Srinath¹, N Vinay Kumar², S Rashmi¹ and B V Poornima¹

Abstract

Accurate crack detection is crucial for maintaining pavement integrity, yet manual inspections remain labor-intensive and prone to errors, underscoring the need for automated solutions. This study proposes a novel crack segmentation approach utilizing advanced visual models, specifically Detectron2 and the Segment Anything Model (SAM), applied to the CFD and Crack500 datasets, which exhibit intricate and diverse crack patterns. Detectron2 was tested with four configurations—mask_rcnn_R_50_FPN_3x, mask_rcnn_R_101_FPN_3x, faster_rcnn_R_50_FPN_3x, and faster_rcnn_R_101_FPN_3x—while SAM was compared using Focal Loss, DiceCELoss, and DiceFocalLoss. SAM with DiceFocalLoss outperformed Detectron2, achieving mean IoU scores of 0.69 and 0.59 on the CFD and Crack500 datasets, respectively. The integration of Detectron2 with faster_rcnn_R_101_FPN_3x and SAM using DiceFocalLoss involves generating bounding boxes with Detectron2, which serve as prompts for SAM to produce segmentation masks. This approach achieves mIoU scores of 0.83 for CFD dataset and 0.75 for Crack500 dataset. These results highlight the potential of combining foundation models with Detectron2 for advancing crack detection technologies, offering valuable insights for enhancing highway maintenance systems.

Keywords Crack detection, Crack segmentation, SAM, Detectron2 model

Introduction

Roads are one of the most critical infrastructures, which must be maintained at a high quality of service. Cost-effective road pavement assessment is crucial for this purpose. Cracks frequently occur as a type of damage in asphalt pavement caused by environmental factors or traffic [1]. Engineering consensus indicates that untreated cracks can cause severe structural damage, shortening the pavement's service life and leading to premature overhaul or reconstruction [2]. Periodic crack detection and assessment are thus vital for asphalt pavement operation and maintenance. Classical techniques, such

as onsite visual inspections, are labour-intensive, time-consuming, ineffective, and could compromise inspector safety. Furthermore, the proficiency and experience of the inspectors have a major role in their efficacy. Therefore, in order to improve pavement performance and help managers cut budget expenses, safer and more effective pavement crack detection and assessment procedures are desperately needed. The automatic recognition of road cracks is vital for early detection, mitigating potential economic losses despite the inherent complexity and variability of crack characteristics. Leveraging advancements in computer vision and machine learning, several detection methodologies have emerged, encompassing image processing [3], traditional machine learning, deep learning [4], transfer learning [5], and more recently, foundation models. However, the challenge of selecting the most appropriate model for specific crack datasets persists, necessitating ongoing iterative experimentation.

*Correspondence:

R Rakshitha
rakshitha.r@jssstuniv.in

¹ JSS Science and Technology University, Mysuru, India

² Freelance Researcher, Bangalore, India



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Recent studies underscore the efficacy of transfer learning in the domain of crack detection, marking it as a pivotal subset of machine learning that has garnered substantial traction, especially with the emergence of large-scale pre-trained foundation models in artificial intelligence. These foundation models, such as large language models (LLMs), are characterized by extensive training on vast and diverse datasets, enabling exceptional generalization and adaptability across a wide range of applications, from content generation to conversational AI. However, in the recent past, the advent of foundation models has driven a paradigm shift towards transformers for visual recognition tasks. These models, equipped with pre-trained transformer networks and lightweight decoders optimized for edge computing, have made multi-modal zero-shot inference in both natural language and images a reality. Notable examples include Generative Pre-trained Transformers (GPTx), the Language Model for Dialogue Applications (LaMDA), Vision Transformer Detectron (ViTDet), and the Segment Anything Model (SAM). This progression has catalyzed new opportunities for advanced pixel-wise detection and segmentation, driving our research to focus on cutting-edge foundation segmentation models, notably Meta AI's SAM and Detectron2, an open-source framework engineered by Facebook AI Research (FAIR). Detectron2, built on the PyTorch platform, offers robust implementations of state-of-the-art models such as Faster R-CNN, Mask R-CNN, and RetinaNet, establishing it as a powerful tool for object detection and segmentation.

In our investigation, we emphasize SAM as a foundation model, which, due to its extensive pre-training, demonstrates unparalleled generalization capabilities, particularly in pixel-wise segmentation tasks. Detectron2 complements this approach by providing a flexible and sophisticated framework for precise crack pixel wise detection. Traditional methodologies often struggle with the morphological variability of cracks and face challenges in generalizing effectively across diverse datasets. In contrast, SAM and Detectron2 mitigate these limitations by leveraging the robust architectures of transfer learning and foundation models, thereby significantly enhancing model robustness and adaptability across varied operational scenarios. This comprehensive analysis, based on experiments conducted on two benchmark datasets, rigorously evaluates the practical applicability of these models in crack pixel detection, validating their potential to deliver accurate, efficient, and scalable solutions in this domain.

In conclusion, the transformative potential of SAM and Detectron2 in crack detection is rooted in their ability to leverage large-scale pre-trained datasets for exceptional generalization and precise pixel-level segmentation. Our

empirical analysis, conducted using two benchmark datasets, validates the efficacy of these models in delivering accurate, efficient, and scalable crack pixel detection solutions.

The notable contributions of this study are delineated as follows:

1. Fine-tuning two state-of-the-art models characterized by distinct architectural configurations adapting them to the task of crack segmentation namely, Detectron2 framework on four baselines and SAM model trained using three loss functions.
2. we integrate Detectron2 with SAM (Combine object detection with Segmentation), train the Detectron2 model using images and masks to generate approximate boundary boxes around the objects of interest is given as input prompt for SAM models to generate segmentation masks.

The subsequent sections of this manuscript are organized as follows: **Related works** section elucidates the methodological framework employed in this study. **Methodology** section, encapsulates the results obtained and offers an in-depth analysis thereof. Finally, **Results and discussion** section encapsulates the conclusion drawn from the findings elucidated in the preceding sections.

Related works

In the past, traditional image processing methods, such as histogram-based thresholding [6], local analysis, and filtering techniques like adaptive filtering and Gabor filters combined with morphological operations [7, 8], have been extensively utilized for crack detection. While these techniques are computationally efficient and straightforward, they often yield high rates of false positives and incomplete detections. Early efforts in machine learning, employing algorithms such as Support Vector Machines (SVM) [9, 10], Markov Models [11], and Random Forest [12], showed promise in crack detection tasks. However, these methods often struggle to generalize in complex environments with background noise, such as shadows or stains, leading to false positives and incomplete detections. To address these shortcomings, deep learning (DL) models have emerged as a more effective alternative, automatically learning hierarchical features from large datasets. Unlike traditional approaches, DL models such as Convolutional Neural Networks (CNNs) can directly learn complex patterns from raw data, significantly improving detection accuracy in challenging scenarios. In environments with shadows, noise, or diffusion points, DL-based approaches [13–15] greatly outperform traditional machine learning techniques, both in terms of accuracy and computational efficiency. Unlike machine

learning models, which depend on handcrafted features, DL models can autonomously learn multiple levels of abstraction, enabling the detection of intricate patterns directly from large datasets.

Transfer learning has been effectively leveraged in crack classification and detection, with models such as EfficientNetB0 [16] and InceptionV3 [17] exhibiting superior performance. For instance, Cao et al. [18] integrated object detection frameworks like Faster R-CNN and SSD with deep convolutional neural networks, attaining the highest mean Average Precision (mAP) of 53.06% when using Faster R-CNN in conjunction with Inception V2. This approach capitalizes on knowledge transfer from analogous domains, enabling the fine-tuning of extensive datasets to optimize models like VGG and MobileNet for crack classification, and employing YOLO [16], Faster R-CNN, and SSD for crack detection, as well as FCN, UNet, and SegNet [18] for crack segmentation. The accuracy of these systems is further enhanced through the application of advanced attention mechanisms, such as transformer-based multi-scale fusion models [19], SegCrack [20], and transformer encoder-decoder architectures [21]. Additionally, segmentation models proposed by NHT Nguyen et al. [19], along with transformer-based fusion models, have substantially improved pixel-level crack detection accuracy. Zou et al. [21] introduced DeepCrack, a model that proficiently captures the hierarchical features of cracks, achieving a pixel-level detection F-measure of 0.89.

The YOLO framework has been employed for the identification of cracks, where crack dimensions are determined by leveraging the precise positions of laser beams projected onto structural surfaces. Huyen et al. [20] introduced the CrackU-Net model, which achieved an exceptional precision of 0.986 in the detection of pavement cracks. Similarly, Kim et al. [21] proposed a crack detection methodology using a shallow Convolutional Neural Network (CNN) architecture, wherein the hyperparameters of the LeNet-5 model were optimized to achieve a peak accuracy of 99.8%, while minimizing the parameter count for improved computational efficiency. Despite the strong performance of these models in feature extraction and classification across various applications, significant challenges remain in enhancing detection accuracy, particularly in environments with complex backgrounds. Although these models have substantially advanced crack detection techniques, the challenge of selecting the most optimal model architecture for specific datasets remains unsolved. The need for further experimentation is paramount, especially within the domain of autonomous systems and advanced computer vision techniques, to continually refine detection accuracy and broaden the applicability of these models. This

paper explores the efficacy of transfer learning in deep feature extraction for crack detection, highlighting a significant improvement in performance.

While architectures such as U-Net, SegNet, and Fully Convolutional Networks (FCN) have demonstrated commendable success in pixel-level crack segmentation, they often face limitations in detecting smaller, finer cracks or require a considerable volume of labeled data to maintain performance. Detectron2, an advanced object detection framework, addresses these limitations by precisely localizing crack regions through bounding box generation, which can then be used as inputs for more accurate segmentation tasks. The Segment Anything Model (SAM), a foundational model, further enhances segmentation by harnessing its generalized learning across diverse datasets, outperforming other models in environments with complex or noisy backgrounds. Notably, prior research has not explored crack pixel detection through the combined application of object detection, segmentation models, and foundational models. This study pioneers the integration of state-of-the-art computer vision technologies, specifically Detectron2 and the Segment Anything Model (SAM), for crack pixel segmentation and detection. To the best of our knowledge, this is the first study to apply the integration of Detectron2 with SAM explicitly for crack detection. Although prior works have utilized YOLO for object detection in conjunction with segmentation networks, the combination of Detectron2 and SAM introduces a novel paradigm. Unlike conventional approaches that focus on object detection or segmentation in isolation, our methodology leverages the advanced object detection capabilities of Detectron2 to generate bounding boxes that serve as prompts for SAM, facilitating a more refined and precise crack pixel segmentation process. This innovative integration, not previously applied to any dataset or domain, distinguishes our work from existing research, significantly improving both detection accuracy and segmentation precision in complex environmental conditions.

Methodology

Proposed method

We propose an automated crack segmentation framework that integrates Detectron2 and the Segment Anything Model (SAM). The process begins with data preparation, where the dataset is split into training, validation, and test sets. The validation set is used to fine-tune model hyperparameters, while the test set evaluates the model's performance on unseen data. Four baseline models are trained using Detectron2, and further refined with SAM by applying various loss functions to optimize the predictions. During testing, Detectron2 generates bounding box prompts, which are directly fed into SAM

for segmentation, producing the final prediction masks. This integration of Detectron2’s object detection capabilities with SAM’s segmentation enhances the accuracy of crack detection in pavements as depicted in Fig. 1. The focus on how Detectron2’s bounding boxes drive SAM’s segmentation process is a core contribution of the study, resulting in improved segmentation performance.

Data preparation

Our research utilizes two benchmark datasets for crack segmentation: the CFD dataset (the smallest) and the Crack500 dataset (the largest), ensuring a comprehensive evaluation. The data was systematically partitioned into training, validation, and testing subsets following an 80:10:10 ratio, selected after extensive experimentation to achieve optimal balance between model learning and evaluation. This partitioning, executed with a seed value of 42 to ensure reproducibility, consistently delivered superior accuracy. Pre-processing involved normalizing pixel values to the 0–1 range and resizing all images to 256×256 pixels for uniformity, facilitating robust model training and evaluation, as represented in Fig. 2.

CFD dataset

The CFD dataset [10] comprises 118 RGB road images captured in Beijing using an iPhone 5, each with a resolution of 480×320 pixels. The dataset consists of 250

training images, 50 validation images, and 200 test images. These images include a variety of noise elements, including oil marks, shadows, and water stains. The dataset focuses specifically on pavement texture and cracks, intentionally excluding irrelevant objects such as garbage or cars on the road. This diversity in noise and environmental conditions poses a significant challenge for evaluating crack detection algorithms, making it a suitable reflection of real-world urban road surface conditions.

Crack500 dataset

Crack500 [12] is a dataset consisting of 500 crack images that were captured at Temple University using cell phones. Each image has a size of 2000×1500 pixels and pixel-level annotated binary maps. The data set includes 250 training, 50 validation, and 200 test images in the dataset. After each image is split into 16 non-overlapping sections only the sections with cracks longer than 1,000 pixels are preserved. This yields 348 validation images, 1,124 test images, and 1,896 training images. With pixel-by-pixel annotations, Crack500 is the largest pavement crack dataset that has been made available to the public for research purposes.

Crack segmentation methods

The Patch-level classification for crack segmentation can rapidly and accurately locate and count surface cracks on

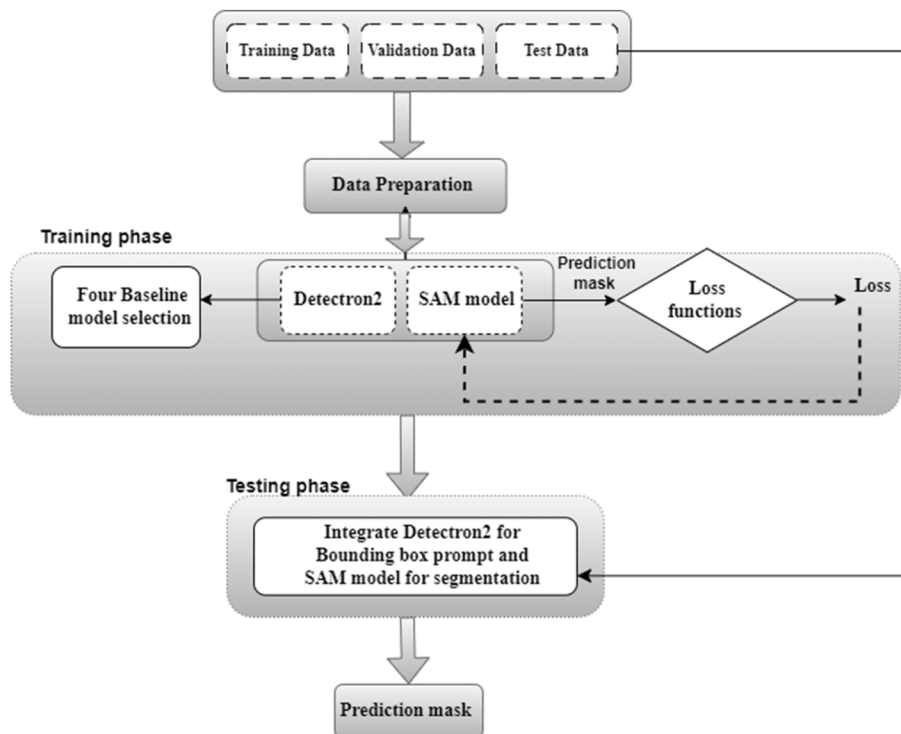


Fig. 1 Flowchart illustrating the proposed methodology

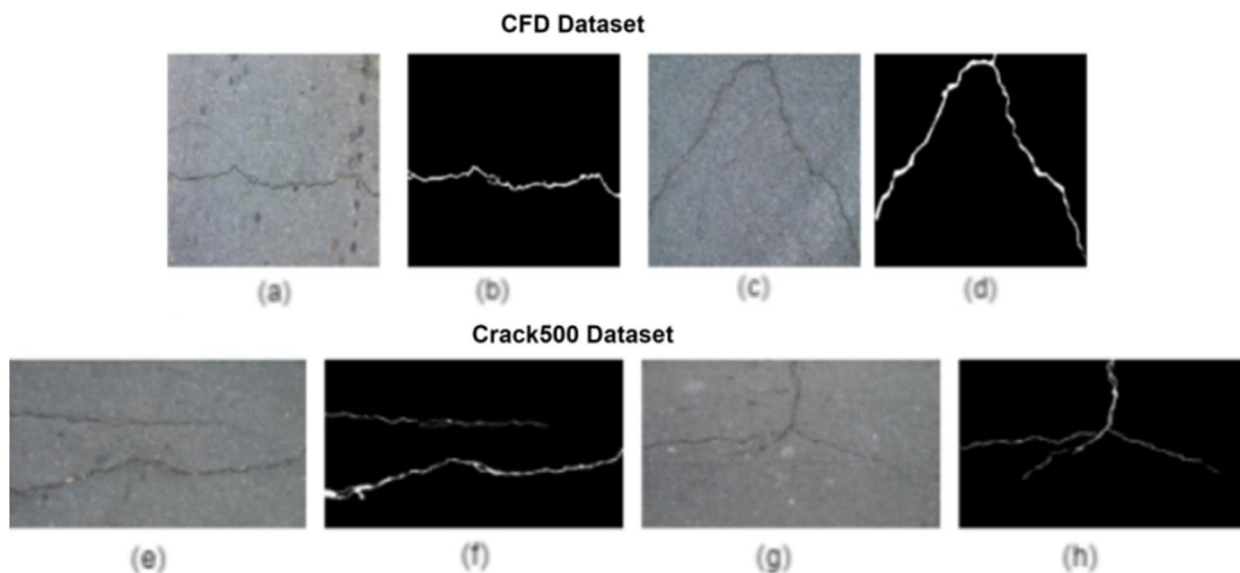


Fig. 2 Sample images and corresponding ground truths from CFD dataset and Crack500 dataset

monitored pavement sections. However, these methods have difficulty providing precise information on individual crack parameters such as length, width, and severity, which are vital for a comprehensive pavement condition assessment. Conversely, pixel-level pavement crack detection delivers detailed crack parameters necessary for thorough pavement condition evaluation, establishing it as the preferred approach for crack assessment.

a) SAM model

The Segment Anything Model (SAM) by Meta AI has gained attention for its impressive zero-shot performance and capability to produce high-quality object masks from diverse input prompts. The primary advantage of SAM compared to other state-of-the-art segmentation models lies in its ability to generalize across a wide range of tasks without task-specific fine-tuning. This adaptability makes SAM a versatile tool, especially when high accuracy is needed across diverse datasets. However, the specific visual examples in the figure might not fully convey this strength, and we will consider adding more representative images to better illustrate SAM's capabilities. SAM functions as a class-agnostic segmentation model, utilizing a Vision Transformer (ViT) for image encoding and a sophisticated two-layer mask decoder. Trained on the extensive SA-1B dataset, which includes over 11 million images and 1.1 billion masks, it stands as the largest segmentation dataset to date. SAM's architecture features an image encoder with ViT to extract detailed embeddings, a prompt encoder to interpret various user inputs, and a lightweight mask decoder for precise pixel-level

segmentation decisions. This design enables SAM to effectively adapt to new segmentation tasks with minimal additional training, ensuring high accuracy.

In this work, ViT-H, an advanced variant of ViT enhanced with self-attention mechanisms, proves crucial for capturing intricate pixel relationships in images. ViT-H's higher resolution and scalability compared to other variants enable it to handle complex spatial dependencies effectively. The bounding box prompts employed in SAM-based crack segmentation is essential for accurately delineating the Region of Interest (ROI), derived from ground truth masks and adjusted randomly during training. These prompts are generated from bounding boxes from ground truth segmentation masks by identifying the smallest enclosing rectangle around the object of interest. This is achieved by locating all non-zero pixels in the mask, which correspond to the crack object, and calculating the bounding box coordinates $[x_{min}, y_{min}, x_{max}, y_{max}]$ based on the minimum and maximum x and y indices of these pixels. During training, bounding boxes are deliberately randomized in size and position to introduce variability, thereby enhancing SAM's robustness and improving its adaptability and generalization across diverse datasets. This approach ensures SAM's efficacy in performing precise segmentation tasks across a broad range of applications.

In the testing phase, the model relies on bounding boxes generated from input images, where ground truth data are unavailable. The model's training with varied bounding boxes equips it to accurately predict the Region of Interest (ROI) during testing, allowing it to effectively manage variability. This method enhances the model's

Table 1 Algorithm to generate the grids of points**Algorithm: Generating and Reshaping Grid Points for Input to a Model**

Input: A 2D image array size and grid size
Output: bounding box prompt for the prompt encoder.

1. Define the size of your array and grid:
Set `array_size` to 256.
Set `grid_size` to 10.
2. Generate grid points:
Use `np.linspace(0, array_size-1, grid_size)` to generate linearly spaced values for both 'x' and 'y' coordinates within the range of `array_size`.
3. Create a grid of coordinates:
Use `np.meshgrid(x, y)` to create a grid from the x and y coordinates.
4. Combine and Convert to Tensor:
Combine the (x, y) coordinates into a list of list of lists.
Convert the combined grid points directly to a PyTorch tensor using `torch.tensor(input_points).view(1, 1, grid_size*grid_size, 2)`.
5. Final Output: Return the reshaped `input_points` tensor, ready to be used as a bounding box prompt by the prompt encoder.

ability to generalize to novel and unseen data, aligning with our goal of achieving precise segmentation across different datasets and conditions. When segmenting a new image with our trained model, a prompt is required. Given that the object locations are unknown, bounding boxes cannot be directly utilized. Instead, we employ a grid of points to generate the bounding box, which serves as the segmentation prompt.

A systematic grid of points is initially generated across the image, with each point corresponding to a specific (x, y) coordinate. These coordinates are then converted into a PyTorch tensor for subsequent processing. The grid is meticulously constructed by partitioning the image into a predetermined number of evenly distributed coordinates, ensuring comprehensive coverage of the image's spatial domain. Significant regions within this grid are delineated by bounding boxes, which play a pivotal role in the segmentation process of the SAM model. These bounding boxes, defined by the minimum (top-left) and maximum (bottom-right) coordinates of the grid, serve as input prompts that guide the SAM model's attention to targeted regions, thereby enhancing segmentation precision. For instance, a random patch within the image can be selected by determining its grid indices, which are then used to generate bounding boxes that focus the model's attention on specific areas, thereby optimizing the segmentation performance of the SAM model, the algorithm is explained in Table 1.

In Fig. 3, the "Image Encoder" and "Prompt Encoder" represent integral components of the Segment Anything Model (SAM). The Image Encoder is tasked with processing the input image to distill essential features, while the Prompt Encoder interprets the bounding box prompts

that strategically guide the segmentation process. The "Mask Decoder" subsequently integrates these encoded representations to produce the final segmentation mask.

During training, experiment is performed using three different loss function are used such as Focal Loss, DiceCELoss, and DiceFocalLoss. Focal Loss, DiceCELoss, and DiceFocalLoss are loss functions designed to enhance semantic segmentation, particularly in the presence of class imbalance. The choice of loss/objective function depends on to minimize the difference between predicted and actual labels, thereby quantifying their discrepancy. The loss functions was chosen because of their effectiveness in handling class imbalance and improving segmentation accuracy.

Focal loss

Focal loss [22] is good for multiclass classification where some classes are easy and other difficult to classify, Focal Loss is tailored to address the challenge of class imbalance by decreasing the relative loss for well-classified examples and placing greater emphasis on difficult-to-classify instances, thus helping the model learn from difficult samples and improving performance on minority classes.

$$FL(pt) = -\alpha t.(1 - pt)^\gamma . \log(pt) \quad (1)$$

DiceCELoss

DiceCELoss [23] combines Dice Loss, which evaluates the overlap between predictions and ground truth segmentations, with Cross-Entropy Loss (CE), enhancing the model's ability to handle both boundary and classification accuracy simultaneously and return their weighted sum.

$$DiceCELoss = \alpha * DiceLoss + (1 - \alpha) * CELoss \quad (2)$$

DiceFocalLoss

DiceFocalLoss [24] merges the benefits of Dice Loss and Focal Loss, then return their weighted sum, ensuring precise boundary segmentation while emphasizing difficult cases and smaller regions, thus effectively managing class imbalance and varying object sizes. These loss functions are crucial for semantic segmentation as they improve the model's ability to accurately segment complex scenes, address class imbalances, and enhance performance on challenging segmentation tasks.

$$DiceFocalLoss = \alpha * FL(Dice(y_{true}, y_{pred})) + (1 - \alpha) * FL(y_{true}, y_{pred}) \quad (3)$$

b) Detectron2

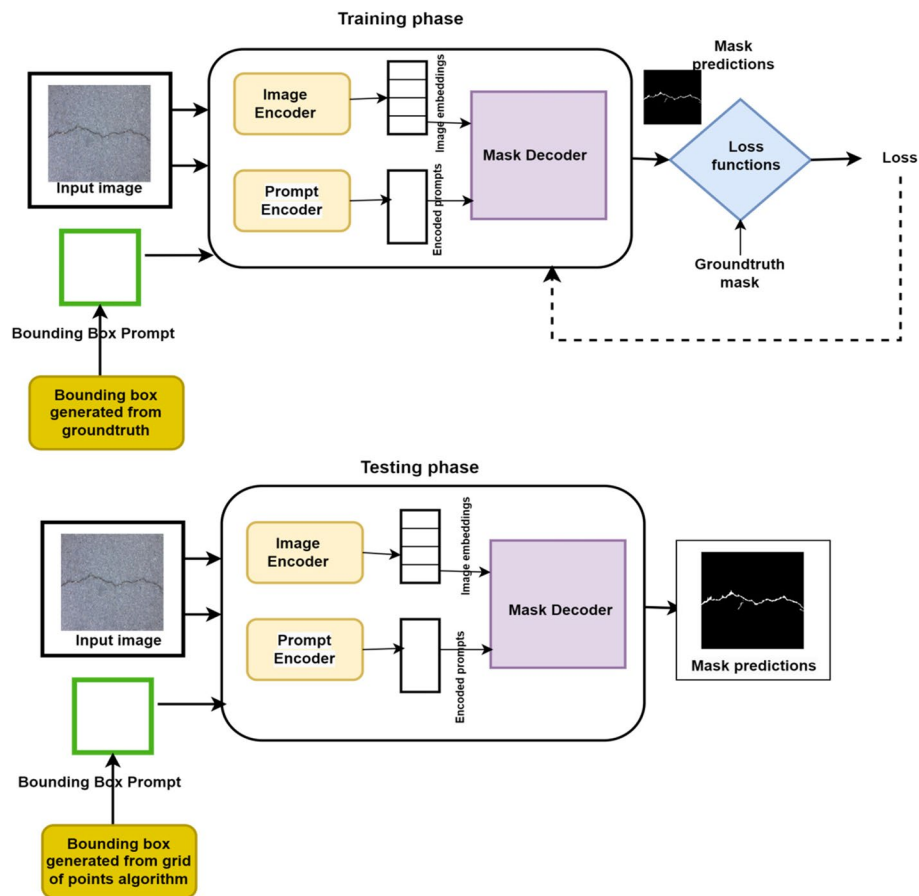


Fig. 3 Proposed network (Crack-SAM) employing the Segment Anything Model (SAM)

Introduced by Facebook AI Research in 2019, Detectron2 [25] includes various computer vision models such as Faster R-CNN, Mask R-CNN, and more, designed for tasks like object detection and segmentation. Its architecture features a backbone for feature extraction, a neck for feature pyramid construction, an RPN for region generation, and a head for detection with non-maximal suppression. We employ instance segmentation baselines such as `mask_rcnn_R_50_FPN_3x`, `mask_rcnn_R_101_FPN_3x`, and object detection baselines such as `faster_rcnn_R_50_FPN_3x`, and `faster_rcnn_R_101_FPN_3x` due to their proven performance and versatility in object detection and segmentation tasks and their ability to balance computational efficiency. Faster R-CNN and Mask R-CNN are selected for their state-of-the-art performance and high accuracy. ResNet-50 provides a balance of performance and efficiency, while ResNet-101 captures more complex features. Feature Pyramid Networks (FPN) ensure robust multi-scale detection. The “`_3x`” notation signifies that the model was trained for three times the standard number of iterations or epochs typically associated with a `1x` schedule. This extended training duration

helps ensure effective convergence, especially when dealing with complex data or model architectures. These models are trained on the COCO 2017 [26] dataset, which includes 200,000 images annotated with 80 object categories, enhancing their generalization capabilities.

To prepare crack CFD dataset annotations for Detectron2, we convert them from standard image format to COCO JSON format using a customized Python script. This process involves using OpenCV to extract contours from binary masks, which represent object boundaries. These contours are then converted into annotations, including bounding boxes, area, and segmentation information. Each annotation is associated with an image ID, category ID, and other properties required by the COCO format. Details such as image ID, height, width, file name, category ID, and bounding box coordinates are extracted from individual images and their label files, and then compiled into a single JSON file.

During training, the model configuration files are adjusted for pixel segmentation alignment. The training process uses a batch size of 4, a base learning rate of 0.00025, and runs for up to 1000 epochs. Since the

dataset includes only one class ('crack'), the number of classes is set to 1. The model's ROI head score is computed using CrossEntropyLoss, while smooth L1 loss is used for coordinate value regression.

Performance evaluation metrics

The evaluation aims to assess the effectiveness of the proposed method for detecting cracks in asphalt pavement. Various metrics are utilized, including Intersection over Union (IoU), Accuracy, Precision, Recall, and F1 score. Precision measures the accuracy of identifying true positive crack pixels, while recall evaluates the proportion of actual crack pixels detected. The F1 score, as the harmonic mean of precision and recall, provides a balanced performance assessment. Accuracy represents the ratio of correct predictions to total predictions. IoU assesses the overlap between ground truth and predicted crack pixels, while Mean Intersection over Union (mIoU) is used for semantic segmentation evaluation.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

and

$$\text{F1 Score} = \frac{2 * TP}{2 * TP + FP + FN} \quad (6)$$

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total Number of predictions}} \quad (7)$$

$$\text{IOU} = \frac{\text{area of Pa intersection Pb}}{\text{area of Pa union Pb}} \quad (8)$$

$$\text{mIoU} = \frac{\text{Intersection over Union}}{n} \quad (9)$$

Results and discussion

The experiments were carried out in an environment with Nvidia T4 GPU support, using models implemented in PyTorch (2.1), Cuda (cu121), detectron2(0.6) and TensorFlow within a Python 3.7 environment. We leveraged libraries such as Keras, SimpleITK, and scikit-learn for development. Detectron2, implemented in PyTorch and CUDA, delivers robust, high-speed, and highly accurate results. The dataset images were kept at their original size of (256, 256) for the four baselines from Detectron2 and the SAM model. The segmentation task was established

as the core objective. The Segment Anything Model (SAM), configured with multiple loss functions, was first scrutinized for segmentation accuracy. Following this, Detectron2's Mask R-CNN baseline was evaluated for crack segmentation. Despite SAM's superior segmentation performance, Detectron2 excelled in object detection, yielding high Average Precision (AP) scores. To enhance overall performance, the bounding box outputs from Detectron2 were subsequently employed as prompts for SAM, facilitating the integration of both models.

We undertook a comprehensive manual hyperparameter tuning process to refine learning rates, batch sizes, and the number of epochs for each model, guided by validation metrics, with a focus on accuracy and loss. Various hyperparameter configurations were systematically evaluated, and optimal values were identified when validation performance plateaued, indicating diminishing returns. The selection of loss functions played a pivotal role in shaping model performance. Within the Segment Anything Model (SAM), we rigorously assessed several loss functions, including Focal Loss, DiceCELoss, and DiceFocalLoss, to gauge their impact on efficacy. Focal Loss mitigates class imbalance by emphasizing hard-to-classify examples, enhancing sensitivity to minority classes. DiceCELoss integrates Cross-Entropy and Dice Loss, delivering pixel-level accuracy and improved overlap, making it well-suited for tasks requiring precise boundary delineation and robust predictions. DiceFocalLoss, a fusion of Dice and Focal Losses, excels under severe class imbalance, providing superior boundary segmentation while prioritizing underrepresented classes. These loss functions are critical for boosting SAM's generalization and performance across diverse tasks, with DiceFocalLoss proving the most versatile for complex, imbalance-sensitive scenarios.

Performance of individual models

SAM Model

The SAM model was trained using images divided into patches with a size of 64 and a step size of 64. The models were optimized using the Adam optimizer with a learning rate of 1e-5 for 100 epochs, and no weight decay. We demonstrate the effectiveness of incorporating various loss functions including Focal Loss, DiceCELoss, and DiceFocalLoss for the Segment Anything Model (SAM). The numerical performance of the methods for different loss functions as shown in the Fig. 4.

When fine-tuned with Focal Loss, DiceCELoss, and DiceFocalLoss, the SAM model shows that DiceFocalLoss achieves lower and more stable mean loss values quickly, indicating efficient learning, while Focal Loss and DiceCELoss struggles with higher, fluctuating

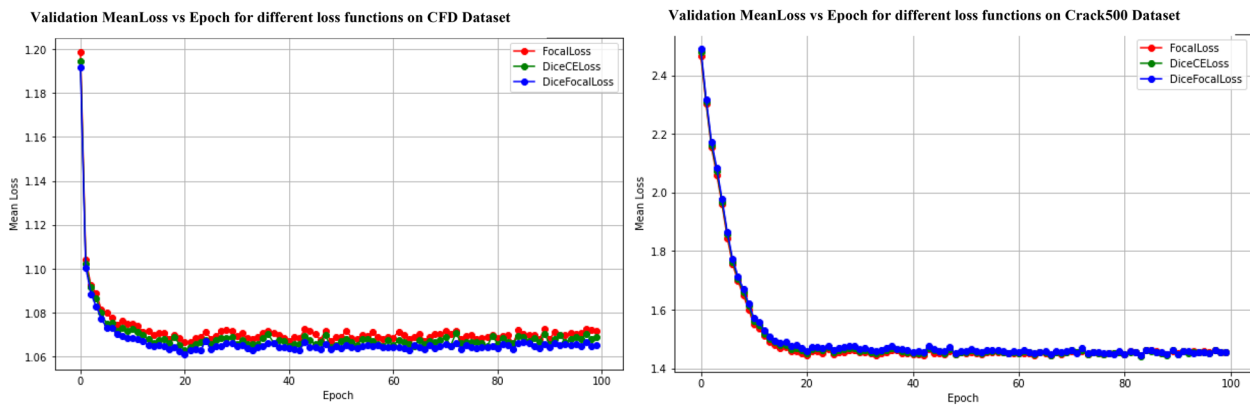


Fig. 4 The performance metrics comparison of SAM model on CFD dataset and Crack500 dataset with different loss functions

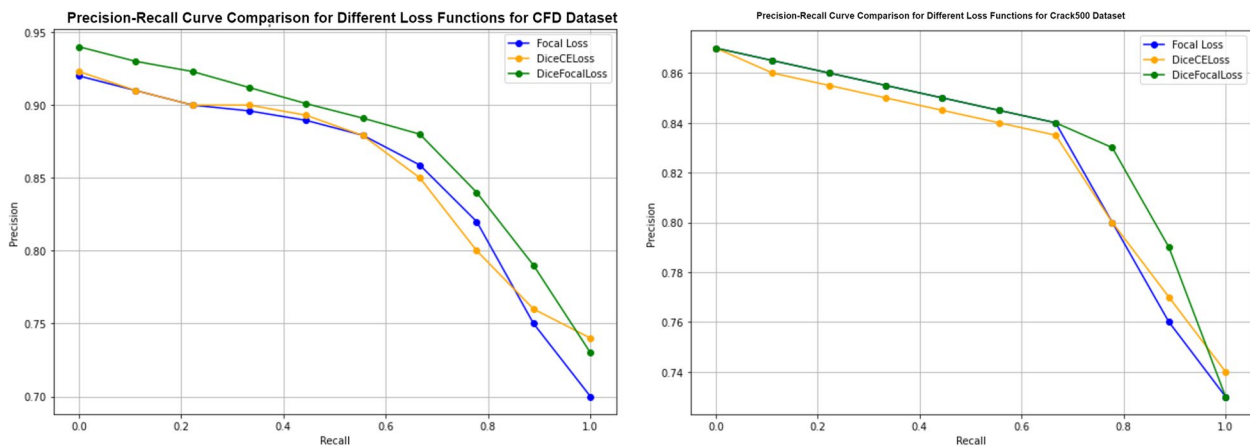


Fig. 5 Precision-Recall curve of our method on different loss functions

values, and proves ineffective for crack segmentation. As the number of epochs increases, the validation mean loss for all three loss functions decreases, suggesting ongoing learning and improvement, with DiceFocalLoss consistently performing best, followed by DiceCELoss and FocalLoss. The decreasing rate of mean loss slows after 60 epochs, indicating the models may be approaching convergence.

Figure 5 illustrates the Precision-Recall curves for three loss functions applied to CFD and Crack500 Datasets. The Focal Loss curve (blue) starts with high precision and recall but dips noticeably towards the end, indicating a drop in precision at higher recall values due to its focus on hard-to-classify examples. The DiceCELoss curve (green) follows a similar pattern but maintains a slightly better balance between precision and recall, especially in the mid-range. The DiceFocalLoss curve (orange) performs the best overall, maintaining higher precision over a broader range of recall values by combining the strengths of Dice and Focal Losses, providing the best trade-off among the three.

The SAM model’s performance is highlighted by the standard deviation Accuracy (0.004 to 0.010) and IoU scores (0.005 to 0.009) across both datasets, indicating consistent and reliable segmentation of crack regions. The higher variance in Accuracy scores suggests sensitivity to different image characteristics, highlighting potential areas for optimization. Additionally, cross-dataset validation demonstrates the model’s adaptability and potential for deployment in varied scenarios, further showcasing its strong generalization capability.

From Table 2, SAM-Focal Loss performs well on the dataset it was trained on, especially CFD dataset, but drops significantly on different datasets. SAM-DiceCELoss shows better performance across all scenarios, particularly in maintaining accuracy and IoU scores on different datasets. SAM-DiceFocalLoss consistently outperforms the other two, demonstrating the best generalization ability with the highest intra- and cross-dataset performance. Therefore, the SAM model for crack segmentation is fine-tuned with DiceFocalLoss, which effectively addresses class imbalance and difficult-to-classify

Table 2 Crack segmentation accuracy assessed by mean accuracy and IoU (mean ± standard deviation)

Method	Train Dataset	Test Dataset	Precision	Recall	F1 Score	Accuracy	IoU scores	Training Duration (hh: mm: ss)	Inference Time
SAM- Focal Loss	CFD dataset	CFD dataset	92.42±0.003	91.49±0.004	92.41±0.003	91.83±0.005	0.64±0.005	04:39:01	0.291
	CFD dataset	Crack500 dataset	70.65±0.003	75.12±0.002	72.96±0.003	73.46±0.009	0.49±0.007	04:41:05	0.339
	Crack500 dataset	Crack500 dataset	85.16±0.003	83.68±0.006	86.32±0.005	88.51±0.004	0.56±0.005	05:17:54	0.346
	Crack500 dataset	CFD dataset	77.29±0.002	75.41±0.001	76.12±0.001	76.90±0.008	0.50±0.008	05:17:09	0.328
SAM- Dice-CELoss	CFD dataset	CFD dataset	94.92±0.003	92.41±0.006	93.55±0.006	92.85±0.006	0.64±0.007	04:42:24	0.310
	CFD dataset	Crack500 dataset	78.31±0.002	76.83±0.005	77.34±0.002	77.49±0.005	0.51±0.006	04:59:10	0.346
	Crack500 dataset	Crack500 dataset	89.53±0.003	85.71±0.006	88.74±0.005	87.99±0.006	0.58±0.008	05:11:36	0.358
	Crack500 dataset	CFD dataset	79.42±0.005	76.64±0.007	77.34±0.003	78.54±0.008	0.53±0.009	05:19:21	0.340
SAM- Dice-FocalLoss	CFD dataset	CFD dataset	96.30±0.002	93.91±0.006	95.34±0.002	95.54±0.004	0.69±0.005	04:48:43	0.350
	CFD dataset	Crack500 dataset	79.90±0.005	82.42±0.005	80.43±0.003	80.90±0.008	0.55±0.008	04:57:10	0.361
	Crack500 dataset	Crack500 dataset	90.42±0.004	92.53±0.005	90.48±0.003	91.53±0.006	0.59±0.006	05:34:45	0.384
	Crack500 dataset	CFD dataset	83.72±0.003	80.18±0.001	82.53±0.002	81.88±0.004	0.56±0.005	05:56:34	0.394

examples, enhancing gradient signals during training and improving model performance. DiceFocalLoss combines Dice Loss’s strength in handling imbalanced classes and Focal Loss’s ability to manage easy negatives, making it well-suited for segmentation models, particularly with datasets where foreground objects are much smaller than the background. In Fig. 6 we demonstrate the effectiveness of incorporating DiceFocalLoss Loss into Segment

Anything Model (SAM) and output is shown for the patch image.

Detectron2 model baseline performance

The performance of the baseline models, including Mask R-CNN (mask_rcnn_R_50_FPN_3x, mask_rcnn_R_101_FPN_3x) for instance segmentation tasks and Faster R-CNN (faster_rcnn_R_50_FPN_3x, faster_rcnn_R_101_FPN_3x) for object detection tasks, was evaluated using

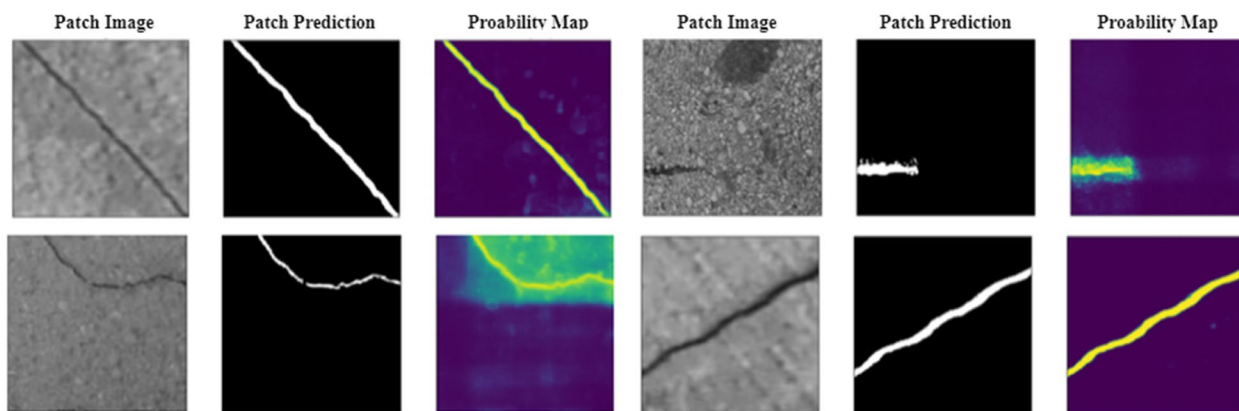


Fig. 6 Visualization depicting the model’s predictions on a patch of the input image alongside the corresponding probability map on CFD dataset

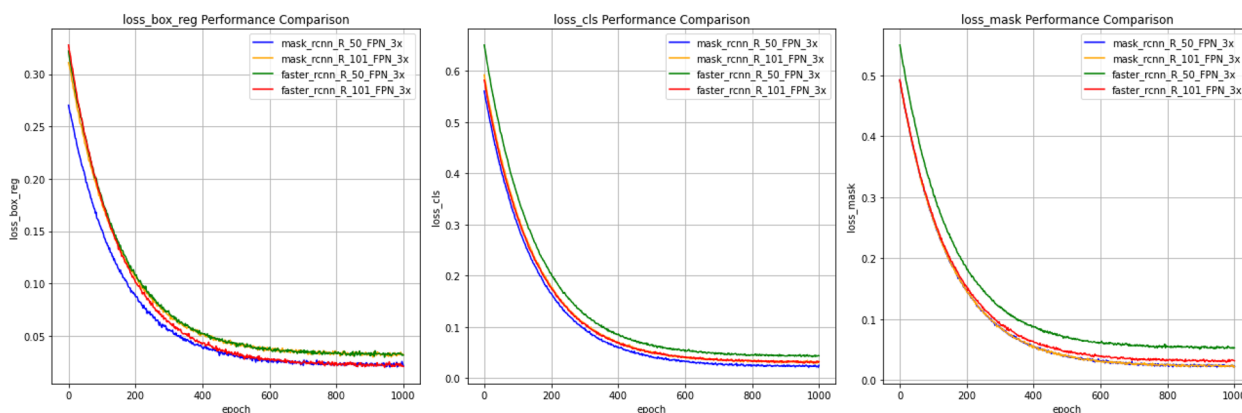


Fig. 7 Analysis of Detectron2’s training metrics as a baseline on CFD dataset

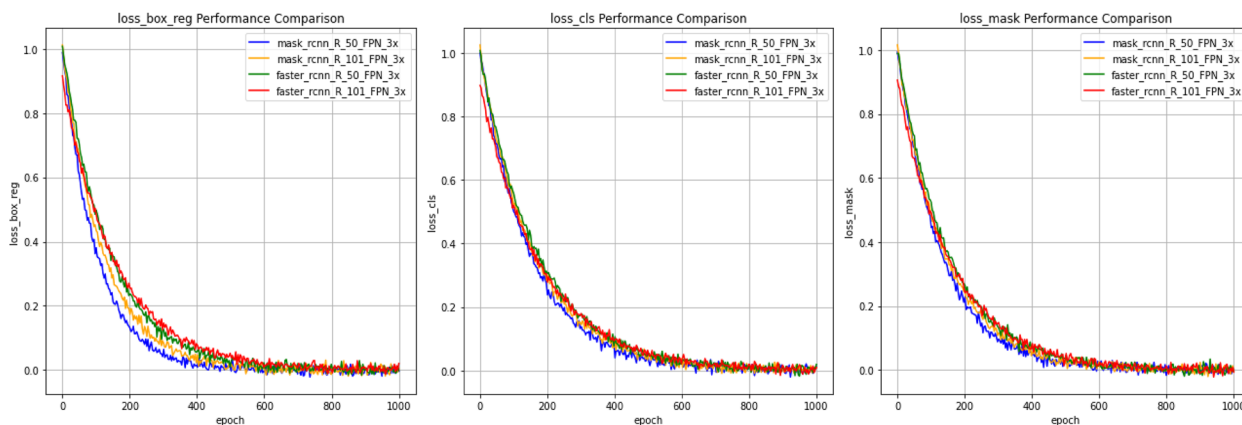


Fig. 8 Analysis of Detectron2’s training metrics as a baseline on Crack500 dataset

COCO metrics in Detectron2. Detectron2 typically uses the Mask R-CNN model for crack segmentation tasks, which extends Faster R-CNN by adding a branch that outputs binary masks for each detected crack object, enabling precise crack segmentation. All baseline models were trained using a learning rate of 0.00025 and a batch size of 8. The training was conducted over a maximum of 1,000 iterations, effectively representing the number of epochs, to ensure the models were adequately trained on the dataset. Evaluation metrics included Average Precision (AP) and Average Recall (AR) based on Intersection over Union (IoU) thresholds, assessing overlap between predicted and ground truth bounding boxes. IoU thresholds of 0.50 (AP50), 0.75 (AP75), and the range 0.50:0.95 were used to evaluate alignment accuracy, with AP [27] and AR values calculated for different object sizes (small, medium, large) and detection limits. Bounding box results report AP metrics for various IoU thresholds and object sizes, including AP, AP50, AP75, and AP for small (APs), medium (APm), and large objects (APl).

The classification of small (APs), medium (APm), and large objects (APl) was based on the pixel area of bounding boxes surrounding the detected cracks. Specifically, cracks were classified as small if their bounding box area was less than 32×32 pixels, medium if between 32×32 and 96×96 pixels, and large if exceeding 96×96 pixels. This approach ensures precise and scalable evaluation across varying object sizes. Loss components analysed include classification loss (loss_cls), bounding box regression loss (loss_box_reg), and mask loss (loss_mask), corresponding to object classification accuracy, bounding box localization accuracy, and segmentation mask accuracy, respectively. Figures 7 and 8 illustrates the degradation of these loss metrics across epochs, showing minimal learning after 200 epochs; however, training was continued for 1,000 epochs to ensure precise localization of crack damage coordinates.

The Figs. 7 and 8 depict a comparative performance evaluation of diverse model architectures—Mask R-CNN and Faster R-CNN, utilizing ResNet-50 and ResNet-101

backbones—across three key loss metrics: classification loss (loss_cls), bounding box regression loss (loss_box_reg), and mask loss (loss_mask) over 1000 training epochs. The convergence trends exhibit notable performance disparities, especially in the bounding box regression loss, where Faster R-CNN with ResNet-50 (FPN) shows a substantial decrease. Moreover, the classification and mask loss metrics underscore the enhanced convergence stability afforded by the deeper ResNet-101 backbone, indicating its superior efficacy in feature representation and model training.

Tables 3 and 4 summarize object detection performance, reported as Average Precision (AP) across different IoU thresholds, using various backbone architectures. “bbox” denotes the bounding box-based approach, while “segm” refers to the instance segmentation-based approach.

The analysis reveals that the faster_rcnn_R_101_FPN_3x exhibits superior bounding box performance on CFD dataset, achieving the highest AP (92.53), AP50 (94.21), and AP75 (96.93), underscoring its exceptional object detection capabilities. Conversely, the Mask R-CNN R-101 FPN 3x excels in segmentation with the highest segmentation AP (91.62). On Crack500 dataset, the Fast R-CNN R-50 FPN 3x demonstrates optimal

performance on small objects (APs 91.91) and the lowest inference time per image (0.3170s), while the Mask R-CNN R-50 FPN 3x achieves the shortest training duration (0:48:44), indicating superior computational efficiency. Models with the ResNet-101 backbone generally achieve higher AP values due to enhanced feature representation and learning capacity, despite increased training and inference times. Overall, the Faster R-CNN R-50 FPN 3x excels in object detection across both datasets, whereas the Mask R-CNN R-101 FPN 3x excels in segmentation, emphasizing the trade-off between model complexity and computational demands.

In conclusion, summarized results in Tables 3 and 4 indicate high average precision for both models, with the Faster R-CNN R-101 FPN 3x performing slightly better overall. Performance can be further enhanced by adjusting IoU and maximum detection settings. The models exhibit limitations in segmentation tasks but demonstrate proficiency in object detection using bounding boxes as shown in Figs. 9 and 10.

Integration of Detectron2 model with SAM model

The integration of Detectron2 with the SAM framework is a cornerstone of our methodology. Despite Detectron2’s advanced architecture tailored for object

Table 3 Evaluation of the performance of Detectron2baselines using CFD dataset

Baselines	Approach	AP	AP50	AP75	APs	APm	API	Training Duration (hh: mm: ss)	Test Duration/ image (seconds)	Max Memory/ Epoch (MB)
mask_rcnn_R_50_FPN_3x	bbox	88.46	89.32	92.84	88.76	91.32	93.43	0:48:44	0.310	8297
	Segm	90.65	91.63	93.09	91.42	93.75	95.31			
mask_rcnn_R_101_FPN_3x	bbox	89.21	91.89	93.44	90.96	92.21	94.79	1:11:44	0.325	12933
	Segm	91.62	92.87	95.12	92.42	94.75	94.93			
faster_rcnn_R_50_FPN_3x	bbox	91.82	93.04	95.93	89.50	91.52	94.21	0:57:44	0.289	9564
faster_rcnn_R_101_FPN_3x	bbox	92.53	94.21	96.93	90.28	93.42	93.04	1:03:30	0.292	13843

Table 4 Evaluation of the performance of Detectron2baselines using Crack500dataset

Baselines	Approach	AP	AP50	AP75	APs	APm	API	Training Duration (hh: mm: ss)	Test Duration/ image (seconds)	Max Memory/ Epoch (MB)
mask_rcnn_R_50_FPN_3x	bbox	83.67	86.32	87.84	82.76	85.32	86.43	1:16:31	0.3170	9341
	segm	85.21	89.89	90.44	84.96	86.21	89.79			
mask_rcnn_R_101_FPN_3x	bbox	84.62	87.87	88.12	83.42	88.75	89.21	1:58:72	0.3425	10073
	segm	86.65	88.63	90.09	88.83	89.49	90.72			
faster_rcnn_R_50_FPN_3x	bbox	88.82	90.51	90.93	88.42	89.52	90.29	1:36:31	0.3389	9761
faster_rcnn_R_101_FPN_3x	bbox	87.74	89.04	90.64	88.37	90.27	90.64	1:47:80	0.3491	11903

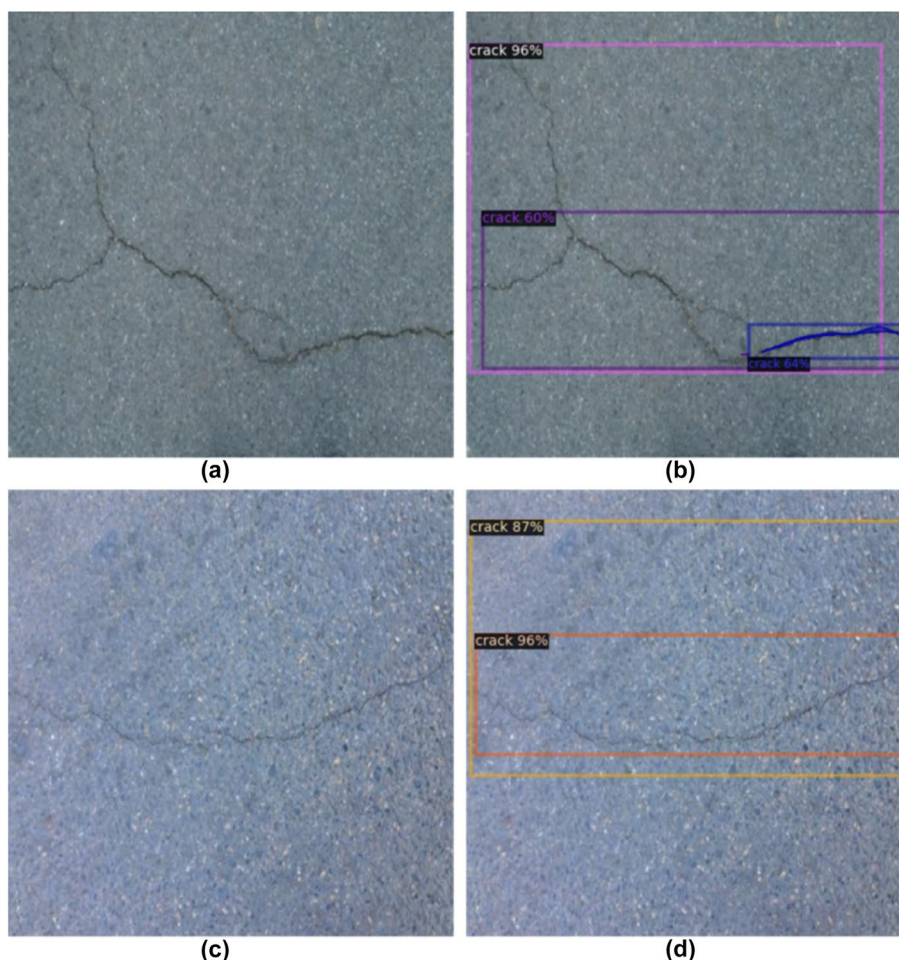


Fig. 9 Evaluation of the Detectron2 Faster R-CNN R-101 FPN 3x baseline models on CFD dataset. (a), (c) are the original images and (b), (d) are the predicted images

detection, it consistently exhibited suboptimal performance in crack segmentation tasks across various datasets, resulting in markedly lower mean IoU, Precision, Recall, and F1-scores compared to SAM models. This performance gap arises primarily due to Detectron2's design, which is optimized for object detection rather than the nuanced demands of instance segmentation, particularly when dealing with complex and heterogeneous crack patterns.

Detectron2 is trained to produce segmentation masks corresponding to the detected objects within the input images. These segmentation masks serve as a preliminary step in approximating bounding boxes that define the regions of interest. Throughout both training and inference phases, the model processes input data to generate these masks, effectively delineating the spatial extent of detected objects. For each identified object, bounding box coordinates are derived through a detailed analysis of the segmentation masks' geometric

properties. This analysis is performed using the region-props function from the skimage.measure library, which computes the properties of the labeled regions within the masks, yielding precise bounding box coordinates that encompass the detected objects.

To enhance performance, we eschewed the conventional grid-based approach for bounding box generation during the testing phase in favor of employing the faster_rcnn_R_101_FPN_3x architecture from Detectron2. This model was specifically deployed to generate bounding boxes around regions of interest, which were subsequently used as input prompts for the Segment Anything Model (SAM). SAM capitalized on these Detectron2-derived prompts to produce high-resolution segmentation masks, refining the initial boundaries with greater precision. This method not only streamlined the crack segmentation process but also yielded substantial improvements in accuracy, as corroborated by the superior performance metrics observed when

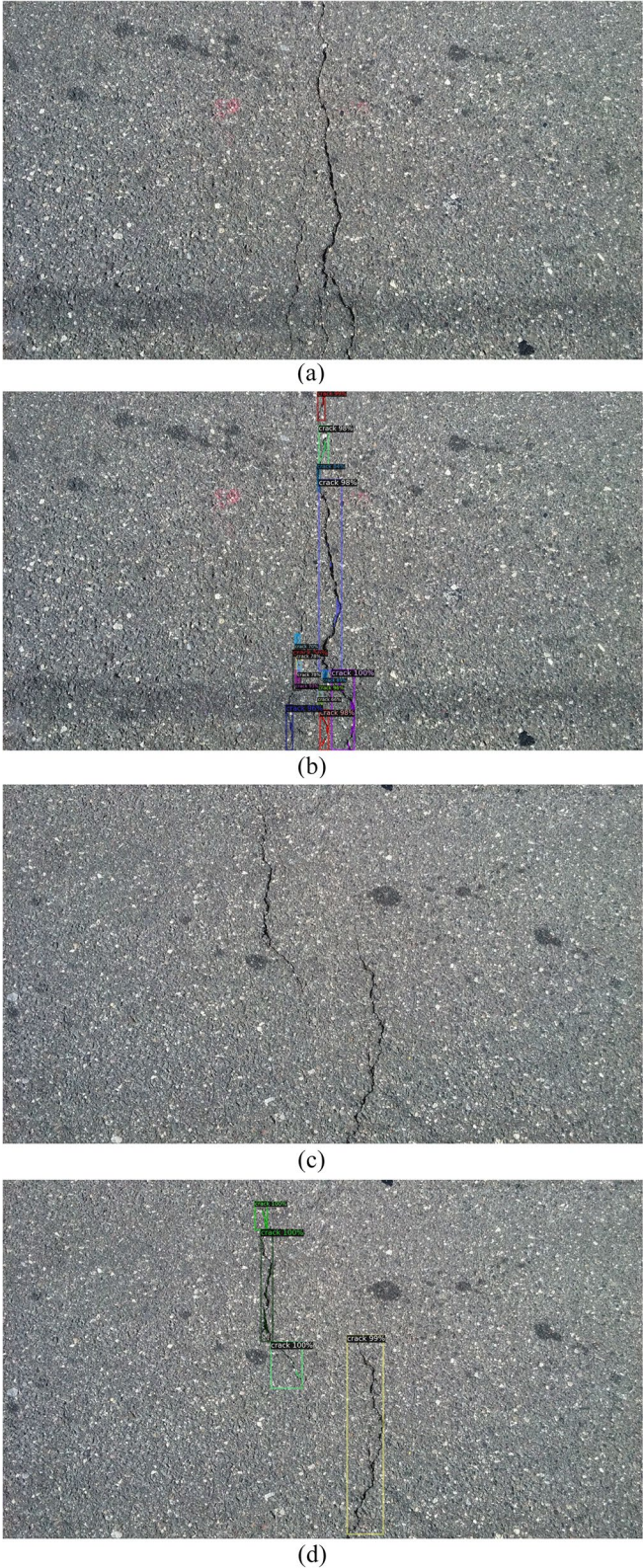


Fig. 10 Evaluation of the Detectron2 Faster R-CNN R-101 FPN 3x baseline models on Crack500 dataset. (a), (c) are the original images and (b), (d) are the predicted images

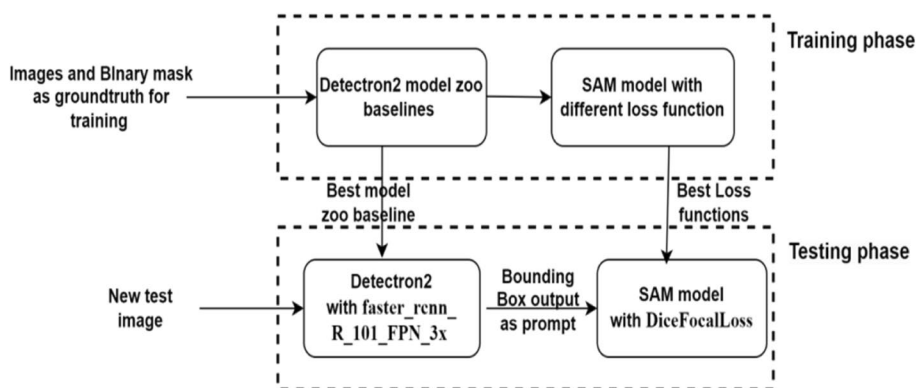


Fig. 11 Workflow of integration of Detectron2 model with SAM model

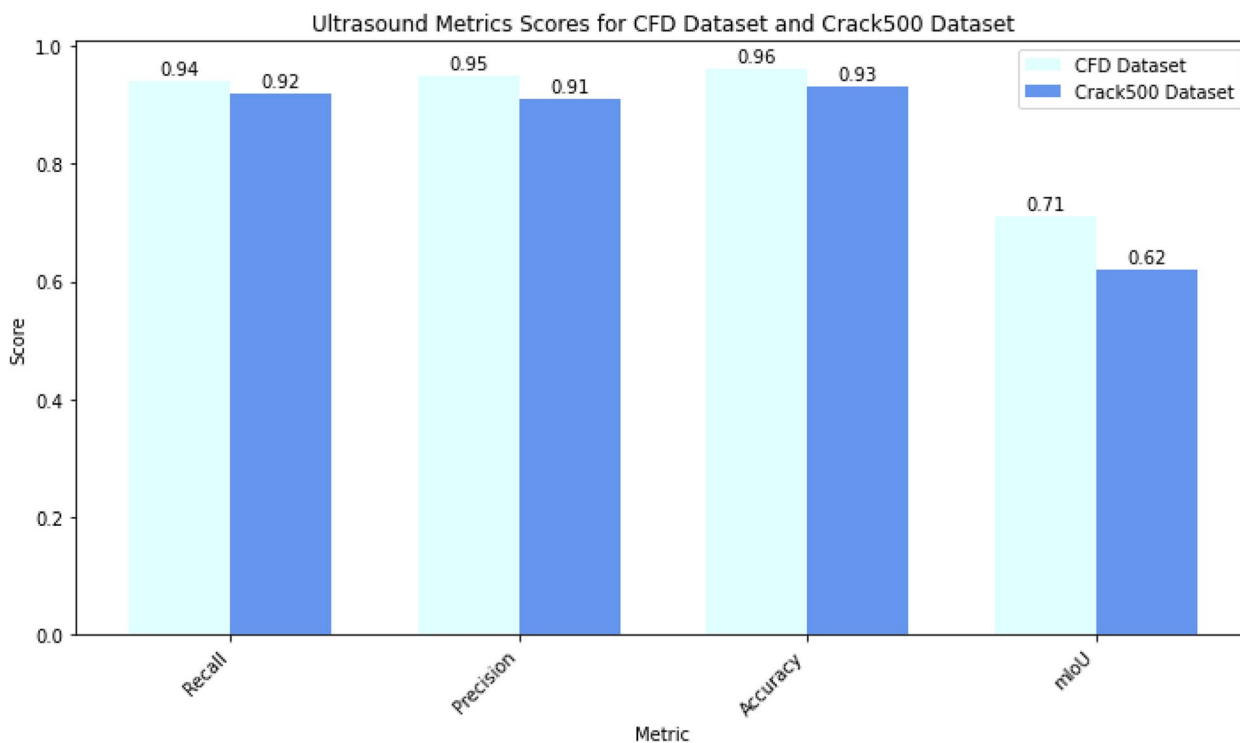


Fig. 12 Performance metrics of Recall, Precision, Accuracy, and mean Intersection over Union (mIoU) for CFD dataset and Crack500 dataset

leveraging Detectron2-generated prompts as depicted in Fig. 11.

The Detectron2 model with baseline `faster_rcnn_R_101_FPN_3x` is employed to detect the ROI, and SAM model is fine-tuned with `DiceFocalLoss`. The performance metrics of the Detectron2-SAM model are presented in Fig. 12.

The bar chart analysis illustrates that the CFD dataset, utilizing the advanced Detectron2-SAM model, consistently surpasses the Crack500 dataset across all key metrics, including Precision, Recall, Accuracy, and

mIoU. The CFD dataset demonstrates superior performance with higher metric scores and reduced variability, reflecting greater robustness and consistency. Its higher median values and more constrained ranges indicate reliable detection and segmentation capabilities, minimizing performance fluctuations. Conversely, the Crack500 dataset shows increased variability, particularly in Recall and mIoU, suggesting less stable performance. The Detectron2-SAM model’s hybrid architecture, which integrates state-of-the-art object detection and fine-tuned segmentation, significantly

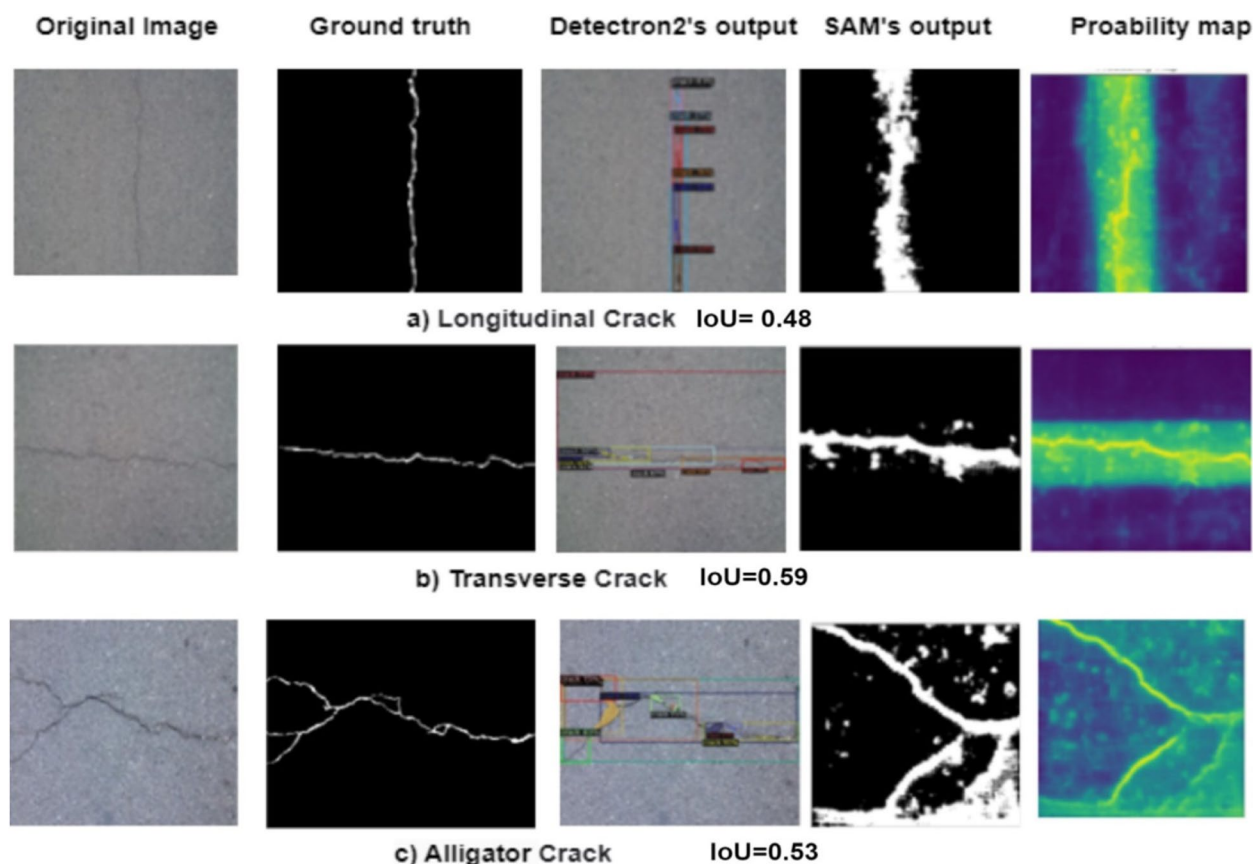


Fig. 13 Crack detection outputs for three types of cracks observed in CFD dataset. Each row displays the Original Image, Ground Truth, Detectron2's Bounding Box Output, SAM's Segmentation Output for the patch image, and its corresponding Probability Map

enhances its efficiency and precision in handling boundary-sensitive tasks, as evidenced in the results for the CFD dataset depicted in Fig. 12.

Overall, the Segment Anything Model (SAM) achieves the highest accuracy in detecting all three types of cracks in both CFD dataset and Crack500 dataset, with CFD dataset specifically containing images of small-width cracks. Despite Detectron2 having the lowest segmentation accuracy, it excels in detecting longitudinal, transverse, and alligator cracks. Accurate crack detection is crucial for proactive pavement maintenance, helping to protect property and reduce the need for labour-intensive manual inspections. However, the proposed method faces significant challenges in terms of generalization, model complexity, and sensitivity to different environmental conditions and image features. The model's performance may decline when applied to varied datasets, especially those with different object types, sizes, or imaging conditions, leading to higher error rates. Additionally, integrating SAM with Detectron2 increases computational demands, resulting in longer processing times, higher memory usage, and greater power consumption, making

it less suitable for environments with limited resources. External factors like changes in lighting, weather, and image quality can further weaken the model's reliability, potentially leading to inaccurate detections and poor segmentation. Addressing these challenges through techniques like data augmentation, transfer learning, or adaptive learning is essential for improving the method's reliability and effectiveness across different real-world situations. The process is further complicated by challenges such as complex texture backgrounds, tiny cracks, and varying lighting conditions, which make it difficult to automate and accurately detect cracks (Fig. 13).

The proposed crack detection approach, integrating Detectron2 with the Segment Anything Model (SAM), faces notable challenges when applied to road surfaces with visually noisy backgrounds, such as shadows and complex textures. As shown in Fig. 14 (a), the model struggles with low-contrast regions, resulting in inaccurate bounding box predictions and under-segmentation. In Fig. 14 (b), it fails to differentiate between cracks and noise patterns, often generating false positives due to the resemblance of noise to crack structures. The CFD

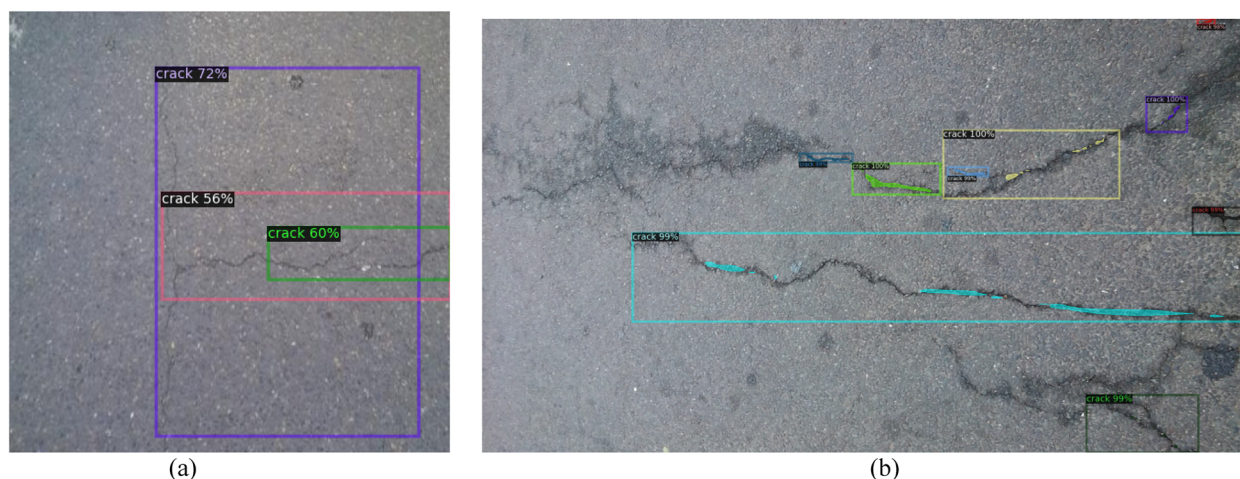


Fig. 14 (a) Bounding boxes predicted by the Detectron2 model for the CFD dataset. (b) Bounding boxes predicted by the Detectron2 model for the Crack500 dataset

dataset shows a 4% failure rate for thin cracks, while the more complex Crack500 dataset reaches a 7% misclassification rate. These challenges highlight the need for architectural enhancements, such as multi-scale feature extraction and attention mechanisms, as well as improved data augmentation to increase robustness and segmentation accuracy in noisy environments.

While the SAM model outperforms Detectron2 in segmentation accuracy, its real-world impact extends beyond mere performance metrics. SAM's enhanced precision in detecting road cracks not only accelerates the identification process but also significantly reduces the operational costs, labor, and time traditionally associated with road maintenance. By minimizing the need for manual inspections, the model enables data-driven prioritization of repair tasks, yielding considerable financial savings and preventing redundant maintenance efforts. However, its limitations become apparent when dealing with lower-quality images or complex crack patterns. In conditions involving suboptimal lighting, occlusions, or highly fragmented cracks, the model's accuracy may deteriorate. To address these challenges, future research will focus on augmenting the model's robustness against diverse environmental conditions, exploring adaptive techniques to improve its performance in more demanding and heterogeneous datasets encountered in real-world scenarios.

State of art comparison



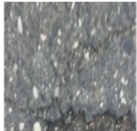





The compared models are being used for a segmentation task. Our method demonstrates significantly better performance compared to other state-of-the-art techniques in this task, showcasing strong generalization capability (Table 5).

The comparative analysis between SAM, Detectron2, and baseline models for crack detection underscores several key technical distinctions, particularly in terms of performance optimization and real-world applicability. While traditional models like U-Net and CrackDiff exhibit commendable outcomes across metrics such as F1 score and mIoU, SAM and Detectron2 surpass these benchmarks through their advanced architectures, excelling in boundary-sensitive segmentation tasks. SAM's refined segmentation precision, when coupled with Detectron2's state-of-the-art object detection framework, delivers significantly enhanced boundary delineation, especially in handling complex, irregular crack geometries. Although SAM's grid-based segmentation introduces higher computational demands, its remarkable scalability and adaptability across heterogeneous datasets far surpass the capabilities of models like DAU-Net, which are computationally intensive and less versatile. Furthermore, SAM and Detectron2 offer a highly optimized balance between computational efficiency and segmentation granularity, rendering them ideal for large-scale, complex crack detection challenges. Their inherent capacity to generalize across diverse datasets and maintain precision without sacrificing computational efficiency positions them as the superior choice for real-world deployment, where scalability, boundary accuracy, and versatility are critical.

Conclusion

This study introduces an advanced deep crack segmentation framework that leverages Detectron2 and the Segment Anything Model (SAM) for precise segmentation of crack defects in pavement images. We explored various baselines from the Detectron2 framework, including

Table 5 The results and comparative analysis of the methods

Baselines				
Method	Dataset	Metrics	Original image	Predicted masks
Encoder–decoder blocks with attention gate [28]	CFD dataset	mean Accuracy = 94.39		
Modified VGG16 model [29]	CFD dataset	F1 SCORE = 0.896		
DAUNET (Deep Augmented UNet) [30]	Crack500 dataset	mIoU = 0.565		
CrackDiff [31]	Crack500 dataset	mIoU = 0.841		
U-Net [32]	CFD dataset	F1 SCORE = 0.9494		
deeply supervised modules [33]	Crack500 dataset	ODS = 0.627		
U-Net [34]	Crack500 dataset	ODS = 0.757		

mask_rcnn_R_50_FPN_3x, mask_rcnn_R_101_FPN_3x, faster_rcnn_R_50_FPN_3x, and faster_rcnn_R_101_FPN_3x. Additionally, we compared the SAM model with different loss functions such as Focal Loss, DiceCELoss, and DiceFocalLoss for crack pixel segmentation. Experimental evaluations on two distinct crack datasets indicate that the enhanced SAM model, optimized with DiceFocalLoss, surpasses Detectron2, achieving mean IoU scores of 0.65 and 0.59 on the test sets of CFD dataset and Crack500 dataset, respectively, with minimal standard deviation. Although Detectron2 demonstrates higher average precision in crack detection, its segmentation performance is comparatively lower, with the Faster R-CNN R-101 FPN 3x baseline showing slightly better results than other baselines. The proposed integrated approach utilizes bounding box outputs from Detectron2 as input prompts for the SAM model, facilitating accurate pixel-level crack detection in asphalt pavements. This methodology achieved mean IoU scores of 0.83 and 0.75 on the test sets of CFD dataset and Crack500 dataset, respectively, underscoring its efficacy in automatic crack detection and quantification for maintenance applications. These advancements have the potential to significantly impact the field of pavement maintenance by enabling more reliable and precise assessments of pavement condition. This, in turn, can

lead to more timely and targeted maintenance interventions, ultimately extending the lifespan of infrastructure and reducing long-term maintenance costs. The integration of these models into automated inspection systems could streamline the process of crack detection, making it more efficient and consistent across different environments. However, we also recognize the need for future research to further validate these findings across diverse datasets and environmental conditions, ensuring that the proposed methods are robust and generalizable for widespread use in real-world applications.

Authors’ contributions

Rakshitha R Methodology, S. Srinath:Writing – original draft, N Vinay Kumar - reviewing the draft , Rashmi S - reviewing the draft , Poornima BV - reviewing the draft.

Funding

The authors they did not receive any financial support, grants, or other assistance while preparing this manuscript.

Availability of data and materials

The datasets used during the current study are accessible at: <https://github.com/cuilimeng/CrackForest-dataset> and <https://github.com/fyangneil/pavement-crack-detection>. The code used in the analysis is available from the corresponding author on reasonable request.

Data availability

No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 22 July 2024 Revised: 13 September 2024 Accepted: 17 September 2024

Published online: 02 October 2024

References

- Koch C, Georgieva K, Kasireddy V, Akinci B, Fieguth P (2015) A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Adv Eng Inf* 29(2):196–210. <https://doi.org/10.1016/j.aei.2015.01.008>
- Weng X, Huang Y, Wang W (2019) Segment-based pavement crack quantification. *Autom Constr* 105:102819. <https://doi.org/10.1016/j.autcon.2019.04.014>
- Munawar HS, Hammad AWA, Haddad A, Soares CAP, Waller ST (2021) Image-based crack detection methods: a review. *Infrastructures* 6(8):115. <https://doi.org/10.3390/infrastructures6080115>
- Shen W, Wang X, Bai X, Zhang Z (2015) DeepContour: a deep convolutional feature learned by positive-sharing loss for contour detection. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit* 07–12–June–2015(October 2016):3982–3991. <https://doi.org/10.1109/CVPR.2015.7299024>
- Ranjbar S, Nejad FM, Zakeri H (2021) An image-based system for pavement crack evaluation using transfer learning and wavelet transform. *Int J Pavement Res Technol* 14(4):437–449. <https://doi.org/10.1007/s42947-020-0098-9>
- Elbehri H, Hefnawy A, Elewa M (2005) Surface defects detection for ceramic tiles using image processing and morphological techniques. *Proc - WEC'05 3rd World Enformatika Conf* 5:158–162
- Chambon S (2011) Moliard J-M (2011) Automatic Road Pavement Assessment with Image Processing: Review and Comparison. *Int J Geophys*. 2011(1):989354. <https://doi.org/10.1155/2011/989354>
- Subirats P et al (2006) Automation of pavement surface crack detection using the. *Image Proc* 1(1):3037–3040
- Chen C, Seo H, Jun CH, Zhao Y (2022) Pavement crack detection and classification based on fusion feature of LBP and PCA with SVM. *Int J Pavement Eng* 23(9):3274–3283. <https://doi.org/10.1080/10298436.2021.1888092>
- Hoang ND, Huynh TC, Tran XL (2022) Tran VD (2022) A Novel Approach for Detection of Pavement Crack and Sealed Crack using image Processing and Salp Swarm Algorithm Optimized Machine Learning. *Adv Civ Eng* 2022(1):9193511. <https://doi.org/10.1155/2022/9193511>
- Chambon S, Gourraud C, Moliard JM, Nicolle P (2010) Road crack extraction with adapted filtering and Markov model-based segmentation: introduction and validation. *VISAPP 2010 - Proc Int Conf Comput Vis Theory Appl* 2(no May 2010):pp81–90. <https://doi.org/10.5220/0002848800810090>
- Shi Y, Cui L, Qi Z, Meng F, Chen Z (2016) Automatic road crack detection using random structured forests. *IEEE Trans Intell Transp Syst* 17(12):3434–3445. <https://doi.org/10.1109/TITS.2016.2552248>
- Li H, Zong J, Nie J, Wu Z, Han H (2021) Pavement crack detection algorithm based on densely connected and deeply supervised network. *IEEE Access* 9:11835–11842. <https://doi.org/10.1109/ACCESS.2021.3050401>
- Zhang L, Yang F, Daniel Zhang Y, Zhu YJ (2016) Road crack detection using deep convolutional neural network. *Int Conf Image Process. ICIP*. pp 3708–3712. <https://doi.org/10.1109/ICIP.2016.7533052>
- Meng X (2021) Concrete crack detection algorithm based on deep residual neural networks. *Sci Program* 2021:1. <https://doi.org/10.1155/2021/3137083>
- Su C, Wang W (2020) Concrete cracks detection using convolutional neural network based on transfer learning. *Math Probl. Eng.* 2020:1. <https://doi.org/10.1155/2020/7240129>
- Ye XW, Jin T, Chen PY (2019) Structural crack detection using deep learning-based fully convolutional networks. *Adv Struct Eng* 22(16):3412–3419. <https://doi.org/10.1177/1369433219836292>
- Cao MT, Tran QV, Nguyen NM, Chang KT (2020) Survey on performance of deep learning models for detecting road damages using multiple dash-cam image resources. *Adv Eng Inf* 46:101182. <https://doi.org/10.1016/j.aei.2020.101182>
- Park SE, Eem SH, Jeon H (2020) Concrete crack detection and quantification using deep learning and structured light. *Constr Build Mater* 252:119096. <https://doi.org/10.1016/j.conbuildmat.2020.119096>
- JOUR et al (2020) CrackU-net: a novel deep convolutional neural network for pixelwise pavement crack detection. *Struct Control Heal Monit* 27(8):1545–2255
- Kim B, Yuvaraj N, Ramasamy S, Rathinakumar A (2021) Surface crack detection using deep learning with shallow CNN architecture for enhanced computation. *Neural Comput Appl* 33. <https://doi.org/10.1007/s00521-021-05690-8>
- Lin TY, Goyal P, Girshick R, He K, Dollar P (2020) Focal loss for dense object detection. *IEEE Trans Pattern Anal Mach Intell* 42(2):318–327. <https://doi.org/10.1109/TPAMI.2018.2858826>
- 'No Title'. Available: https://docs.monai.io/en/stable/losses.html#dice_loss
- 'No Title'. Available: <https://docs.monai.io/en/stable/losses.html#dicefocalloss>
- Kirillov A, Wu Y, He K, Girshick R (2020) Pointrend: image segmentation as rendering. *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit* 9796–9805. <https://doi.org/10.1109/CVPR42600.2020.00982>
- Lin TY et al (2014) Microsoft COCO: common objects in context. *Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)* 8693:740–755. https://doi.org/10.1007/978-3-319-10602-1_48
- He K, Gkioxari G, Dollár P, Girshick R (2018) "Mask R-CNN," arXiv preprint arXiv:1703.06870. [Online]. Available: <https://arxiv.org/abs/1703.06870>
- Raza Ali MAS, Chuah JH (2022) Mohamad Sofian Abu Talip, Norrima Mokhtar, 'Crack Segmentation Network using additive attention Gate—CSN-II'. *Eng Appl Artif Intell*. 114
- Qu Z, Mei J, Liu L, Zhou DY (2020) Crack detection of concrete pavement with cross-entropy loss function and improved VGG16 network model. *IEEE Access* 8:54564–54573. <https://doi.org/10.1109/ACCESS.2020.2981561>
- Polovnikov V, Alekseev D, Vinogradov I, Lashkia GV (2021) DAUNet: deep augmented neural network for pavement crack segmentation. *IEEE Access* 9:125714–125723. <https://doi.org/10.1109/ACCESS.2021.3111223>
- Zhang H, Chen N, Li M, Mao S (2024) The Crack Diffusion Model: an innovative diffusion-based Method for Pavement Crack Detection. *Remote Sens* 16(6):986. <https://doi.org/10.3390/rs16060986>
- Song W, Jia G, Zhu H, Jia D, Gao L (2020) Automated pavement crack damage detection using deep multiscale convolutional features. *J Adv Trans* 1:6412562
- Li H, Zong J, Nie J, Wu Z, Han H (2021) Pavement crack detection algorithm based on densely connected and deeply supervised network. *vol XX*. <https://doi.org/10.1109/ACCESS.2021.3050401>
- Zhao F, Chao Y, Li L (2023) A Crack Segmentation Model Combining Morphological Network and Multiple Loss Mechanism. *Sensors* 23(3):1127. <https://doi.org/10.3390/s23031127>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.