Automation in Construction 170 (2025) 105960

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Hybrid-Segmentor: Hybrid approach for automated fine-grained crack segmentation in civil infrastructure

June Moh Goo [a,*], Xenios Milidonis [b], Alessandro Artusi [b], Jan Boehm [a], Carlo Ciliberto [c]

[a] Department of Civil, Environmental and Geomatic Engineering, University College London, Gower Street, London, WC1E 6BT, United Kingdom
[b] DeepCamera MRG, CYENS Centre of Excellence, Nicosia, Cyprus
[c] Department of Computer Science, University College London, Gower Street, London, WC1E 6BT, United Kingdom

ARTICLE INFO

Keywords:
Deep learning applications
Semantic segmentation
Convolutional neural networks
Transformers
Hybrid approach
Crack detection
Crack dataset
Fine-grained details

ABSTRACT

It is essential to detect and segment cracks in various infrastructures, such as roads and buildings, to ensure safety, longevity, and cost-effective maintenance. Despite deep learning advancements, precise crack detection across diverse conditions remains challenging. This paper introduces Hybrid-Segmentor, a deep learning model combining Convolutional Neural Networks-based and Transformer-based architectures to extract both fine-grained local features and global crack patterns, significantly enhancing crack detection for improved infrastructure maintenance. Hybrid-Segmentor, trained on a large custom dataset created by merging multiple open-source datasets, can accurately detect cracks on different types of surfaces, crack shapes, and sizes. The model demonstrates robustness and versatility by accurately detecting discontinuities, vague cracks, non-crack regions within crack areas, blurred images, and complex crack contours. Furthermore, when compared against other recent models for crack segmentation, the proposed model achieves state-of-the-art performance, significantly outperforming them across five key metrics: accuracy (0.971), precision (0.807), recall (0.756), F1-score (0.774), and IoU (0.631).

1. Introduction

Cracks in roads, pavements, and buildings pose a serious threat to public safety, causing accidents and damage to vehicles on roads and pavements and influencing public safety and the financial burden on buildings. Traditionally, manual inspections have been used to identify cracks in civil infrastructure, but these methods are labor-intensive, subjective, and prone to human error, resulting in inconsistent results and potential disasters [1–3]. Therefore, automated crack detection is necessary to provide an objective and highly accurate alternative. Machine learning methods, such as deep learning models, can be used to detect, segment or classify damage to civil infrastructure, which can be facilitated by the widespread deployment of surveillance cameras [4] and traffic monitoring cameras [5]. However, training accurate models for crack segmentation is challenging due to the scarcity of well-annotated and diverse datasets, which affects model robustness and generalizability. Our research aims to address this crucial data gap and develop automated crack detection to prevent dangers and reduce financial risks to communities. Progress in this direction could lead to the real-time identification of cracks in the future, ensuring a more reliable and safe utilization of critical concrete structures. The main contributions of this paper are:

- Combine and refine publicly available crack datasets to create an enhanced and extensive crack segmentation dataset.
- Introduce a data refinement methodology to combine publicly available datasets using image processing techniques.
- Introduce the Hybrid-Segmentor model to efficiently detect cracks in infrastructures, which is based on the encoder–decoder architecture that convolutional neural networks (CNNs) and transformers have efficiently used in the past.
- Emphasize the approach of the proposed model to perform effectively across a diverse range of surface types and under challenging imaging conditions, such as blurred images and areas with complex crack contours.
- The code, trained weights of the model, and the full dataset for experiments are publicly available and can be accessed here: https://github.com/junegoo94/Hybrid-Segmentor

2. Related work

2.1. Crack segmentation models

One of the earliest methods for crack detection was a CNN-based model for pixel-level crack detection using FCN [6]. This approach

* Corresponding author.
E-mail address: june.goo.21@ucl.ac.uk (J.M. Goo).

https://doi.org/10.1016/j.autcon.2024.105960
Received 4 September 2024; Received in revised form 15 December 2024; Accepted 29 December 2024
Available online 7 January 2025

(a) Blurred Image (b) Brick Image

Fig. 1. Representative failures in crack detection by traditional models: (a) shows a prediction by a Fully Convolutional Network (FCN) on a blurred image, incorrectly marked as a crack, highlighting difficulties with image clarity. (b), from a U-Net architecture, displays a brick pattern where the borders of the bricks are wrongly identified as cracks, revealing challenges in differentiating structural boundaries.

achieves end-to-end crack detection, significantly reducing training time compared to CrackNet [7], a CNN-based model that was the State-Of-The-Art (SOTA) in 2017 without using a pooling layer. While thin cracks can be accurately predicted across a variety of scenes, further enhancements are needed to capture real-time level predictions. In a similar aspect, DeepCrack [8] improves the generalization of the FCN architecture by incorporating batch normalization and side networks for faster convergence. Additionally, this research proposes the publicly available DeepCrack dataset [8], which enhances crack detection precision across diverse scenes. Cheng et al. [9] propose a full crack segmentation model based on U-Net. Subsequent research further demonstrated that the U-Net is particularly suited for crack segmentation tasks [9–13]. Some researchers pinpoint that using classical image classification structures as encoders, pre-trained with data such as ImageNet [14], strengthens feature extraction in crack segmentation networks, enhancing crack detection performance [11].

In addition, various encoder–decoder models have been introduced in the field. Among these, DeepCrack2 (not to be confused with 'DeepCrack' in Liu et al. [8] bearing the same name; we refer to this model as 'DeepCrack2' from this point onward to avoid confusion) is a deep convolutional neural network designed to facilitate automated crack detection through end-to-end training [15]. Specifically, it focuses on acquiring high-level features that effectively represent cracks. This approach involves the integration of multi-scale deep convolutional features obtained from hierarchical convolutional stages. This fusion enables the capture of intricate line structures, with finer-grained objects in larger-scale feature maps and more holistic representations in smaller-scale feature maps. DeepCrack2 adopts an encoder–decoder architecture similar to SegNet [16] and employs pairwise feature fusion between the encoder and decoder networks at the corresponding scales. DeepCrack2 is one of the most benchmarked models in the crack segmentation community.

Despite the abundance of studies that employ existing deep learning models or enhance them, these approaches may not always produce effective or efficient results in real world scenarios (Fig. 1). Recently, HrSegNet [17] was proposed as an approach to consistently maintain high resolution in the images, distinguishing itself from methods that restore high-resolution features from low-resolution ones. Furthermore, the model improves contextual information by leveraging low-resolution semantic features to guide the reconstruction of high-resolution features [17]. These features helped HrSegNet-B64 reach SOTA in accuracy and inference speed in crack segmentation.

2.2. Limitations in current approaches

Although both CNN-based models and Transformer-based models have shown prominent improvements in crack segmentation, they both convey disadvantages in recognizing cracks from images. CNN models have a limited receptive field, making it difficult to capture long-range dependencies and global context in images [18,19]. They also struggle with complex structures and require multiple layers to process global information [20]. In contrast to CNN-based models, Transformer-based models have shown limitations in capturing high-frequency components of images [21]. This can hinder their ability to effectively detect local textures and edge details [21,22]. Since cracks often have complex and varied textures, the inability to accurately capture high-frequency information may reduce the effectiveness of Transformer models in detecting fine structural details essential for segmentation tasks.

Existing deep learning-based crack segmentation studies have trained models [6,7,9,15] on crack images collected from a single surface type individually rather than from diverse sources. However, deep learning models that fuse data from multiple sources, effectively interpreting cross-domain feature correlations, have consistently succeeded in reducing false detections caused by image disturbances and noise from individual datasets, thus improving model performance [23]. Deep learning-based crack segmentation should focus on leveraging fused image data to enable more detailed and robust feature representation and extraction through advanced data fusion strategies and effective deep learning design [23,24]. Additionally, given the lack of fused image datasets revealed, further research efforts should be directed towards discovering, developing, and sharing large-scale fused image datasets [23].

Regarding datasets, while several datasets for crack-related tasks are available, there is a noticeable lack of open-source large-scale datasets. In the broader computer vision domain, large-scale datasets such as ImageNet [14] have accelerated computer vision research due to their substantial sample sizes. Similarly, in the crack segmentation domain, a single large-scale dataset for training and evaluating algorithms would be highly beneficial. Datasets such as SDNet2018 [25], with more than 50k images for classification, are steps in the right direction, but more annotated data are still needed for segmentation tasks. Furthermore, deep learning performance is often limited by the size of training datasets and the complexity of networks.

3. Dataset

We introduce a large refined dataset with the aim of creating a significantly larger and more diverse resource for crack segmentation compared to what is currently available in the literature. Since existing datasets contain a relatively small number of images compared to other well-known tasks in computer vision, large-scale deep learning models are at a high risk of overfitting in these settings. In contrast to most datasets for crack segmentation that collect data based on a single type of surface, the refined extensive dataset includes a wide range of surfaces to enhance the robustness and generalizability of trained models. Additionally, due to the characteristics of some cracks, each existing image has a small proportion of crack pixels, which could result in a form of class imbalance. To counteract this bias, we employed a data augmentation strategy to increase the number of crack pixels in our dataset.

Table 1
Sub-datasets details before data refinement.

Dataset	Size	Resolution	Surface	Crack proportion (%)
Aigle-RN	38	Various Sizes	Pavement	0.71
CFD	118	480 × 320	Pavement	1.62
CRACK500	500	2000 × 1500	Pavement	6.01
CrackLS315	315	512 × 512	Pavement	0.25
CrackTree260	260	Various Sizes	Pavement	0.46
CRKWH100	100	512 × 512	Pavement	0.36
DeepCrack	537	544 × 388	Diverse surfaces	3.5
ESAR	15	512 × 768	Pavement	0.6
GAPs384	384	640 × 540	Pavement	0.36
LCMS	5	1000 × 700	Pavement	0.67
Masonry	240	224 × 224	Bricks/Masonry walls	4.21
SDNET2018	56 092	256 × 256	Pavement	–
Stone331	331	512 × 512	Stone Surfaces	0.11
Total dataset				2.69

3.1. Sub-dataset details

We identified 13 open-source datasets that include different surfaces of pavements, walls, stone, and bricks. Table 1 shows the details of each dataset. Some datasets provide samples either collected with specific acquisition systems and under diverse background settings (e.g. Aigle-RN, ESAR, and LCMS that collectively form the AEL Dataset) [26]; or acquired with smartphone cameras (e.g. CRACK500) [27]. A number of small datasets provide road and pavements images, including CrackTree260, CRKWH100, CrackLS315 and Stone331 [28]. (e.g. CrackTree260 is a dataset of 260 visible-light road pavement images constructed based on CrackTree206 [29]) DeepCrack [8] is a large dataset created as a publicly available benchmark dataset consisting of crack images captured across various scales and scenes, specifically designed to evaluate the performance of crack detection systems. The German Asphalt Pavement Distress (GAPs) dataset, introduced in Eisenbach et al. [30], addresses the issue of comparability in pavement distress research, offering a standardized dataset with 1,969 images with high-quality gray value. It covers various distress classes, including cracks, potholes, and inlaid patches. The images have a resolution of 1,920 × 1,080 pixels with a per-pixel resolution of 1.2 mm × 1.2 mm. To enable pixel-wise crack prediction, 384 images are manually selected from GAPs and annotated, forming the GAPs384 dataset [31]. Masonry is created consisting of images captured from masonry structures, which exhibit intricate backgrounds and a diverse range of crack types and sizes [32]. CrackForest dataset (CFD), one of the most benchmarked datasets, is a labeled collection of road crack images, designed to represent the typical conditions of urban road surfaces [33,34]. Finally, SDNET2018 is a dataset comprising more than 56,000 images of cracked and non-cracked concrete bridge decks, walls, and pavements, with crack widths ranging from 0.06 to 25 mm. Since the dataset does not contain ground truth masks, we use this dataset only for the collection of non-cracked image data [25].

3.2. Data refinement

Ground truth masks in existing datasets were generated using different methods, leading to varying resolutions, distortions, and discontinuity. To address this inconsistency, masks were manually inspected and refined using basic image processing where deemed necessary to ensure that no irregularities were present, based on a process described previously [35]. Due to the inconsistency of AEL datasets with the rest of the datasets (inverted and not binary), dedicated processing steps were performed. First, the values in the masks were inverted. The pixels were then converted to either black or white based on a threshold of 255/2. All images included in our dataset were then cropped to 256 × 256 resolution without overlapping. Finally, due to the reduced number of images with cracks, we augmented our dataset with a significant portion of cracks to address the class imbalance. Specifically, images with masks containing over 5000 crack pixels were selected for augmentation, where Gaussian noise is added, and a random rotation of 90°, 180°, or 270° is applied. Non-crack data from the SDNet2018 dataset [25] were also added.

Fig. 2 shows how the original ground truth improved after the refinement process. Irregularities such as small holes, discontinuity, and thinness were corrected. Furthermore, adding the augmented dataset increased the proportion of crack pixels by 5.8%, aiming to mitigate class imbalance problems. As a result, we created a refined extensive dataset with a total of 12,000 images, which is the largest crack dataset to the best of our knowledge.

4. Model design

This section provides an in-depth overview of our Hybrid-Segmentor, an end-to-end crack segmentation model. As shown in Fig. 3, our model processes input images via two separate encoders: the CNN pathway utilizing ResNet-50 [36] and the Transformers route employing SegFormer [37]. Each of these encoders generates 5 multi-scale feature maps, which are then fused together at each of the 5 intermediate stages. In the last step, the fused feature maps are utilized to produce the final output (simplified decoder). The overall benefits of our Hybrid-Segmentor, through the combination of the two different deep learning architectures, are the ability to detect local details and global structural understanding, while spatial hierarchy leads to more accurate crack detection. Integrating features at different scales from both paths enables effective recognition of cracks of various sizes and shapes, leveraging the strengths of both local and global analysis. This ensures higher accuracy and robustness in detecting cracks in diverse types of surfaces. Sections 4.1 and 4.2 further describe the benefits introduced by the CNN and transformer paths of our architecture, respectively.

4.1. CNN path

The use of a CNN architecture is guided by the fact that we would like to capture local features from the input image. These features are both fine-grained local details, e.g., small cracks or textures, and high-level features, such as abstract shapes. This is achieved through the spatial hierarchy property of the ResNet-50 model used in our Hybrid-Segmentor, which allows the detection of various image features at multiple scales. Additionally, its translation-invariant property will help with extracting features regardless of the crack position within the input image. Finally, its capability to preserve high-resolution details will make our model more effective in detecting small cracks or local variations.

4.2. Transformer path

The use of a transformer in crack segmentation aims to extract global features from the input image, which are crucial to capture the overall shape and appearance of a crack. Here, we use as our base key concepts from the SegFormer model [37]. Through its self-attention mechanism, this model can recognize the continuity and structure of cracks that span distant regions, understanding how different parts of the crack relate to each other across the image (Long-range Dependency Capture). SegFormer also incorporates a spatial hierarchy, similar to CNNs, by processing features at different scales, making it able to capture both fine details and global structures. Another important property of the SegFormer is its global consistency, which, by analyzing the image entirely, provides insights into how cracks are distributed across the entire image, ensuring a coherent understanding of the crack patterns.

The transformer of our proposed model utilizes three additional key concepts, which are explained below: Overlapping Patch Embedding (Section 4.2.1), Efficient Self-Attention (Section 4.2.2), and Mix-Feed Forward Network (Section 4.2.3).

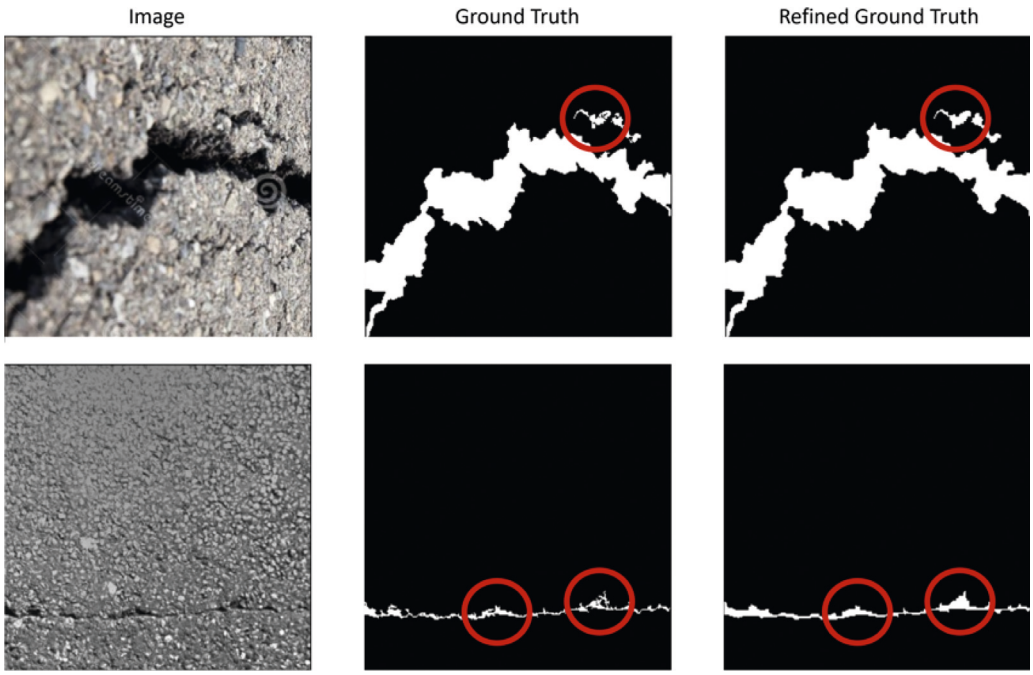

Fig. 2. Improvement in small holes, discontinuity, and thinness of ground truth after applying appropriate image processing methods.

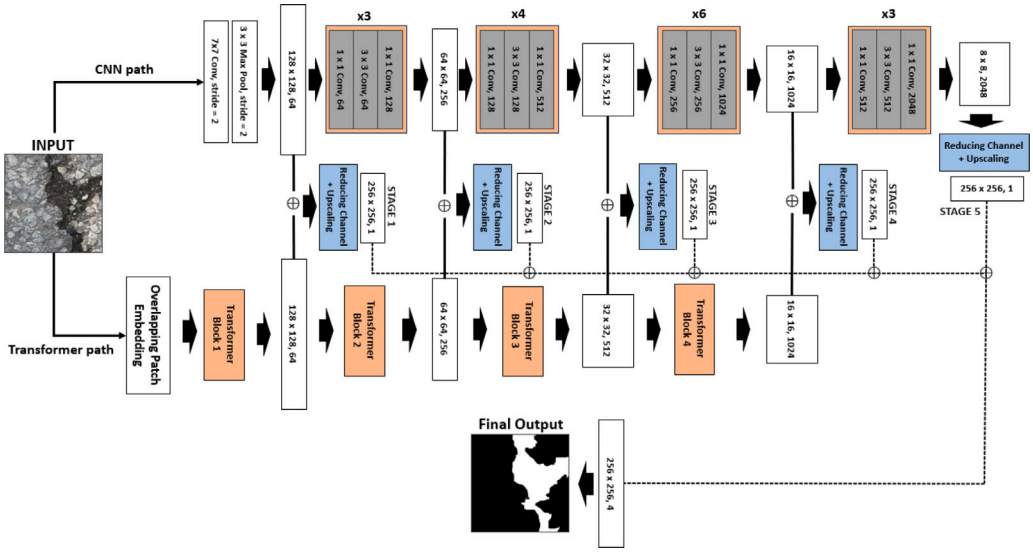

Fig. 3. Hybrid-Segmentor architecture: the upper path for CNN and the lower for Transformers. Each path generates feature maps at every layer, and the central blue boxes represent the concatenation of these feature maps.

4.2.1. Overlapping patch embedding

Local continuity is crucial for preserving fine-grained details and spatial coherence, which is important for accurate semantic segmentation. The first iterations of vision transformers used non-overlapping patch embeddings, which could lead to a loss of local continuity between patches. However, to address this, we utilized overlapping patch embedding, as introduced by SegFormer [37], which better preserves local continuity.

The Vision Transformer (ViT) [38] is an innovative approach to computer vision. It treats images as sequences of patches and processes them similarly to how transformers handle sequences of words in natural language processing. In a typical ViT architecture, an image is divided into $N \times N$ patches, which are then linearly embedded into $1 \times 1 \times C$ vectors. While this method enables the model to effectively capture global context, it can still be challenging to maintain local continuity among patches when $N \times N \times 3$ image patches are represented as $1 \times 1 \times C$ vectors.

To address this issue, SegFormer employed Overlapping Patch Embedding. Instead of simply dividing the image into non-overlapping 4×4 patches for vector embedding, Overlapping Patch Embedding takes inspiration from how CNNs use sliding windows with defined parameters such as kernel size (K), stride (S), and padding (P). It predefines these parameters to split the input image into patches of size $B \times C \times K^2 \times N$, where B represents the batch size, C is the number of channels times the stride squared, and N is the number of patches. Merging operations are then performed to transform the reshaped patches to $B \times C \times W \times H$, where W and H represent the width and height of the merged patches, respectively. As a result, the model captures both fine-grained local details and broader global features more effectively, addressing the issue of losing local continuity while still maintaining global context.

4.2.2. Efficient self-attention

Especially in models like SegFormer with smaller patch sizes like 4×4, the self-attention layer presents computational challenges. The traditional multi-head attention process involves creating matrices for query (Q), key (K), and value (V), all of which have dimensions $N(H \times W) \times C$, and performing computations using the scaled dot-product attention equation as shown in Eq. (1).

$$\text{Attention(Q, K, V)} = \text{Softmax}(\frac{QK^T}{\sqrt{d_{head}}})V \tag{1}$$

When dealing with large input images, the computational complexity of the provided Eq. (1) can lead to a significant increase in model weight. Therefore, the method that reduces the $N(H \times W)$ channels of K and V by applying a sequence reduction process based on a predefined reduction ratio is proposed [37]. It is possible to reshape the equation by dividing N by R and multiplying C by R. $C \times R$ dimensions can be reduced to C dimensions by linear operation, resulting in $\frac{N}{R} \times C$ dimensions for the Key and Value matrices. Especially useful for tasks like semantic segmentation, this method efficiently manages computational complexity while preserving representation power (Eq. (2)).

$$\begin{aligned} \hat{K} &= Reshape(\frac{N}{R}, C \cdot R)(K) \\ K &= Linear(C \cdot R, C)(\hat{K}) \end{aligned} \tag{2}$$

4.2.3. Mix-FFN

ViT [38] uses positional encoding for local information, which comes with fixed input resolution constraints and suffers performance drops as resolution changes. In order to overcome this issue, researchers replace positional encoding with a Convolutional 3×3 kernel in the FFN, asserting its non-essential role and providing flexibility without resolution restrictions.

$$\mathbf{x}_{out} = MLP(GELU(Conv_{3\times3}(MLP(\mathbf{x}_{in})))) + \mathbf{x}_{in} \tag{3}$$

In this regard, Eq. (3) simply adds a Convolutional 3×3 layer to the existing FFN within the Transformer encoder. By replacing traditional positional encoding with this adaptation, the model performance is maintained while fewer parameters are required.

4.3. Decoder

The CNN and transformer paths of our model result in substantial model complexity and size. In order to balance this, the decoder is designed to be as simple as possible. A $256 \times 256 \times 1$ feature map is generated by concatenating the outputs of both paths, as shown in Fig. 3. Feature maps from each stage are combined to create a multi-scale, multi-layer feature map, which is then used to create the final output. By integrating the strengths of both paths, this approach optimizes performance and efficiency while simplifying the decoder.

5. Experimental settings

5.1. Training setup

Models are trained and tested in a GPU cluster with 8 nodes, each with 8 NVIDIA RTX A5000 (24 GB of GPU memory onboard), running Rocky Linux 8.5 and using Python 3.10, PyTorch 2.0.1 and PyTorch Lightning 2.4.0. For all models, we use early stopping with the patience of 10 epochs to ensure convergence and avoid over-fitting.

5.2. Data

The refined dataset contains 12,000 images with and without cracks, along with the ground truth for each image. A random shuffling method is used to distribute the dataset between training, validation, and testing sets, with a ratio of 8:1:1. As a result, our dataset consists of 9600 samples for training, 1,200 samples for validation, and 1200 samples for testing. All models are trained on the training dataset, and the best model is selected based on validation dataset losses. The final model is then evaluated on the unseen test dataset.

5.3. Metrics

Our model is evaluated and compared with several metrics. We use Accuracy, Precision, Recall, Intersection over Union (IoU) and F1-Score (Dice) as standard metrics. To further validate, we use the OIS (Optimal Image Scale) and ODS (Optimal Dataset Scale) as additional evaluation metrics, which are widely used in crack segmentation tasks [39–44]. The OIS calculates the average F1 score across all images in the test dataset by averaging the maximum F1 score on a per-image basis at its own optimal threshold. On the other hand, ODS identifies a single optimal threshold for the entire dataset, then applies this threshold to every image, and averages their F1 scores. ODE and OIS are described by the following equation:

$$ODS = max(2\frac{P_t \times R_t}{P_t + R_t}) \tag{4}$$

$$OIS = \frac{1}{N}\sum_{i=1}^{N} max(2\frac{{P_t}^i \times {R_t}^i}{{P_t}^i + {R_t}^i}) \tag{5}$$

Where, $t = (0.01, 0.02, \ldots, 0.99)$ denotes the threshold, while P_t and R_t refer to the precision and recall at a given threshold t. ${P_t}^i$ and ${R_t}^i$ signify the precision and recall of the ith image. N is the total number of test images.

Using both OIS and ODS is beneficial because OIS does not overweight images with a lot of crack pixels and images with a large number of crack pixels, while ODS provides a comprehensive performance evaluation of the algorithm across the dataset.

6. Experiments

6.1. Benchmarks

To assess the performance improvement of our model over traditional segmentation models, we compare it with FCN [45] and UNet [46]. Furthermore, we include the DeepCrack2 model [15], a widely benchmarked crack detection model, as well as SegFormer [37] and HrSegNet-B64 [17], which represent SOTA models in semantic segmentation and crack segmentation, respectively. The performance of our model is assessed both quantitatively and qualitatively.

For all models, we use the Adam optimizer with a learning rate of 1.00e−04 and a batch size of 16. The learning rate of 1.00e−04 is selected based on [47], which showed that this learning rate leads to better performance improvement and minimal fluctuation among other widely used learning rates. Regarding the batch size, we chose 16, following approaches used in crack segmentation tasks performed on a Nvidia A5000 GPU [17,48,49]. This batch size is selected as a safe choice that does not exceed the memory limits of the GPU. Other hyperparameters are provided in Table 2.

Table 2
Hyperparameter settings of benchmarked models (LR indicates Learning Rate).

Model	LR schedule	Pre-trained
Hybrid-Segmentor	ReduceLROnPlateau	ResNet (IMAGENET1K)
HrSegNet	ReduceLROnPlateau	None
DeepCrack2	None	None
SegFormer	PolynomialLR	None
UNet	None	None
FCN	None	VGG19 (IMAGENET1K)

Table 3
Comparison and performance analysis of CNN and transformer paths.

Model name	Accuracy	Precision	Recall	F1 score (Dice)	IOU score
Hybrid-Segmentor (combined)	**0.970**	**0.805**	**0.732**	**0.765**	**0.622**
CNN path	0.969	0.802	0.722	0.758	0.614
Transformer path	0.965	0.717	0.772	0.741	0.592

6.2. Loss functions

Experimentally, it has been shown that class imbalance in datasets can be effectively addressed not only by using a well-designed architecture but also by using a well-designed loss function [50–52]. To improve the robustness of our model, we evaluated the performance of various loss functions: Binary Cross Entropy (BCE) [53], Dice [51], the fusion of BCE and Dice [50], and Recall Cross Entropy (RecallCE) [52].

The BCE loss has been chosen for its ability to handle skewed pixel distributions effectively. In scenarios where one class significantly outweighs others, BCE loss computes an individual loss for each pixel to ensure proportional class contribution, mitigating any dataset bias. By treating pixels equally, the model is able to focus on accurately classifying minority classes, such as crack pixels, without being biased by dominant classes. BCE loss is described by the following equation:

$$BCE(y, \hat{y}) = -\frac{1}{N}(y_i \, log(\hat{y}) + (1 - y_i) \, log(1 - \hat{y}_i)) \tag{6}$$

where N is the total number of elements (pixels in the case of segmentation). y_i is the ground truth label (0 or 1) for the ith element, and $\hat{y}_i$ is the predicted probability for the ith element.

Dice loss, which is equivalent to the F1-score, addresses the class imbalance, focusing on capturing the overlap between predicted and ground truth masks, which helps address the challenge of minority class representation. By emphasizing object boundaries and assigning non-vanishing gradients to the minority class, Dice loss ensures accurate prediction and better learning for smaller classes.

$$\text{Dice Loss} = 1 - \frac{2 \cdot Intersection}{Union} \tag{7}$$

Its ability to sensitively measure the similarity between prediction and ground truth makes it particularly useful for precise segmentation. It can be used alongside other losses, such as BCE loss, to maintain a balance between handling class imbalance and capturing fine details [50]. Here, we used a combination of BCE and Dice losses as follows:

$$\text{BCE-DICE} = \lambda \times \text{BCE loss} + (1 - \lambda) \times \text{Dice loss} \tag{8}$$

where λ represents the weight (importance) attributed to the two loss functions and takes values between 0 and 1.

Previous methods attempt to improve standard cross-entropy loss in segmentation tasks by incorporating weighted factors. However, this approach can lead to issues such as reduced precision and an increased false positive rate for minority classes. To address this problem, RecallCE loss is proposed as a hard-class mining solution. It reshapes traditional cross-entropy loss by dynamically adjusting class-specific loss weights according to a real-time recall score, offering a more effective way to handle class imbalance and improve segmentation precision [52]. We evaluate the performance of our model by comparing the RecallCE loss with the other losses previously mentioned to determine if it enhances our model's effectiveness. The equation for RecallCE loss is as follows:

$$\text{RecallCE} = -\sum_{c=1}^{C} \sum_{n:\, y_i = c} (1 - R_{c,t}) \, log(p_{n,t}) \tag{9}$$

where $R_{c,t}$ represents the recall of class c during optimization iteration t.

7. Evaluation

We carry out two prior studies to examine specific aspects of our model: (1) assessing the impact of individual encoder paths and (2) evaluating the performance of various loss functions. Initially, we aim to understand how each encoder distinctly extracts features. Then, we investigate which of the aforementioned individual loss functions and the combination of BCE and Dice losses (by assigning different weights) yields the best results in crack segmentation. Once we determine the loss function providing the optimal performance, we compare our final model with SOTA crack segmentation models.

7.1. Encoder paths

We conduct an experiment involving the training and testing the two different encoders to assess their abilities in feature extraction. Specifically, we aim to determine whether convolutional layers perform well at extracting local features while transformers are adept at capturing global features. Each path is trained as an independent network by removing the influence of the other and is compared against the fused network. An identical loss function is used for all networks for a fair comparison (BCE-DICE loss with $\lambda = 0.5$).

The results presented in Table 3 indicate that the CNN path achieves a higher precision score than the transformer path, while the latter excels in terms of recall. This suggests that the transformer tends to produce more false positives, mistakenly predicting non-crack pixels as cracks. On the other hand, the CNN path tends to produce more false negatives, possibly misclassifying crack pixels as non-cracks. These results suggest that the transformer path captures broader areas as cracks, while the CNN path captures finer details. Combining the two paths into a fused model leverages the power of both and improves the accuracy and precision of crack segmentation without significantly sacrificing recall. Fig. 4 shows example segmentations produced by each of the two encoders, further illustrating the differences in their performance.

7.2. Loss functions

We utilize various types of losses (BCE, Dice, and RecallCE) to address class imbalances and capture fine-grained details. Our experiments reveal that combining BCE and Dice losses provides a balance between recognizing dominant classes and accurately segmenting minority groups, resulting in a more effective model for imbalanced data than when using the loss functions individually (Table 4). We assess these aspects by varying the weights assigned to BCE and DICE loss functions. When the BCE and DICE loss weights are roughly equal, the model generally performs better. A BCE-DICE loss with $\lambda = 0.2$ outperforms other values in all metrics except for precision. Precision peaks at 0.817, but this trades off with recall, resulting in relatively lower performance in other metrics.

Expectedly, RecallCE results in the highest recall score, as it heavily penalizes the model for false negatives while producing well-balanced results for the other metrics. However, this loss is behind BCE-DICE in terms of accuracy and precision, indicating that it may be less effective at addressing class imbalance. In summary, the BCE-DICE loss with $\lambda = 0.2$ exhibits the best model performance, and is chosen as the loss function for our final model.

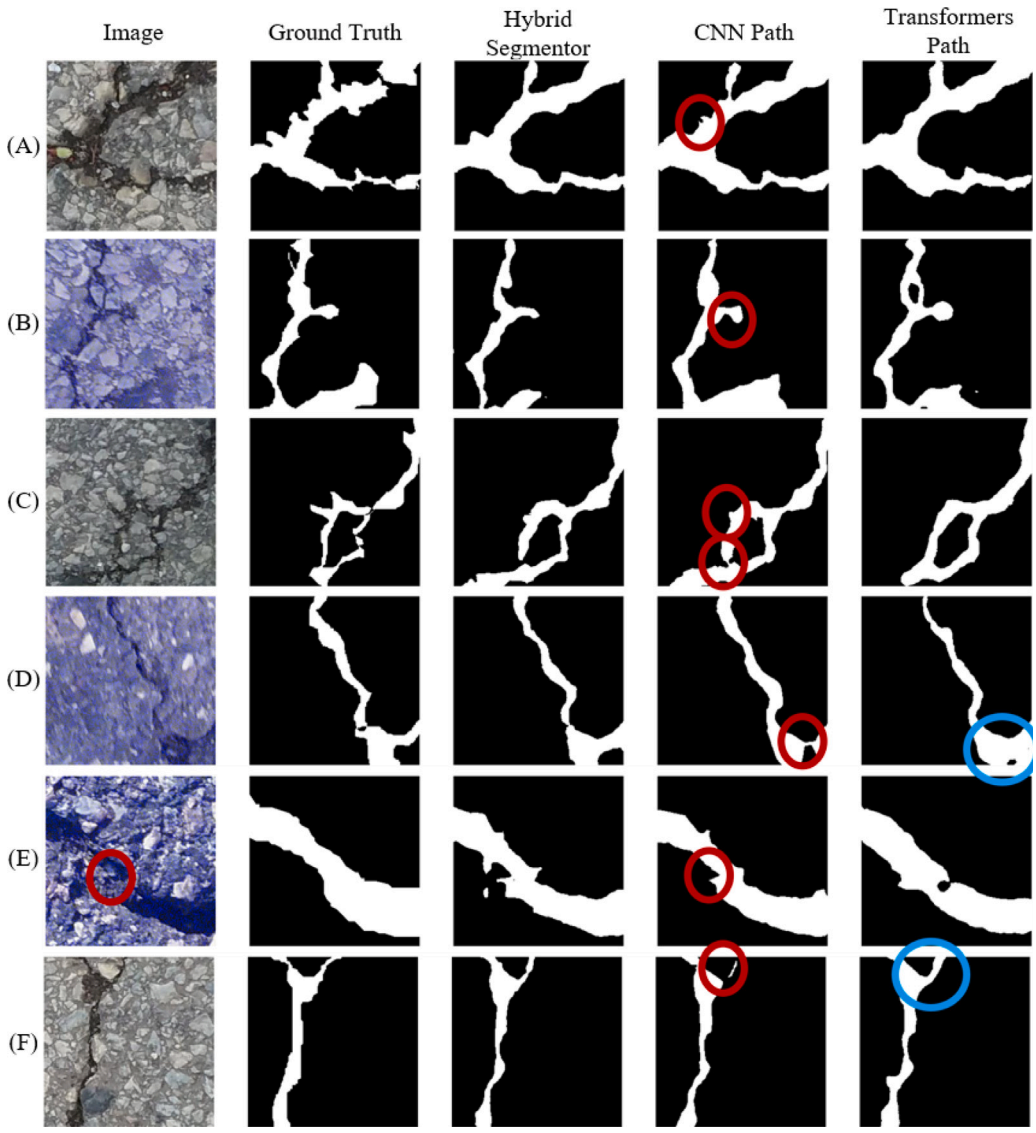

Fig. 4. Hybrid model performance compared against CNN and transformer paths. CNN path captures detailed contours (red circles), while Transformer path gives an overall structure but with thicker predictions (blue circles).

Table 4
Performance of all combinations of loss functions.

Loss functions (λ)	Accuracy	Precision	Recall	F1 score (Dice)	IOU score
DICE	0.970	0.807	0.727	0.763	0.620
BCE-DICE (0.1)	0.970	0.796	0.741	0.765	0.622
BCE-DICE (0.2)	**0.971**	0.807	0.744	**0.770**	**0.630**
BCE-DICE (0.3)	0.970	0.809	0.719	0.759	0.615
BCE-DICE (0.4)	0.970	0.809	0.720	0.760	0.616
BCE-DICE (0.5)	0.970	0.805	0.732	0.765	0.622
BCE-DICE (0.6)	0.970	0.805	0.736	0.767	0.625
BCE-DICE (0.7)	0.970	0.808	0.724	0.762	0.618
BCE-DICE (0.8)	0.969	**0.817**	0.700	0.752	0.605
BCE-DICE (0.9)	0.969	0.804	0.719	0.757	0.612
BCE	0.969	0.778	0.750	0.762	0.618
RecallCE	0.970	0.795	**0.746**	0.768	0.626

7.3. Comparison against SOTA models

We compare our best model using BCE-DICE loss ($\lambda = 0.2$) to the benchmark models in our experiment. We train 10 times for all models and compute the mean and standard deviation (std). This approach ensures a reliable comparison by capturing training variations and providing a clearer view of the performance of each model. The standard deviation further highlights the consistency of the results across multiple runs, showing the stability of each model.

As demonstrated in Table 5, our model significantly outperforms the other five models. Our model achieved an accuracy of 0.971, a precision of 0.807, a recall of 0.756, an F1-score of 0.774, and an IoU score of 0.631. These results demonstrate the model's exceptional proficiency in crack segmentation tasks. Furthermore, our model has the lowest standard deviation, indicating stable and consistent performance across multiple runs.

As in Table 6, our model demonstrates high performance in terms of both ODS and OIS metrics. In terms of ODS, our model achieves the highest score among other state-of-the-art models, which means that our model consistently achieves high performance across various crack images. This indicates that our model provides stable and consistent results even with a variety of crack patterns present in the dataset. On the other hand, for OIS, our model performs second to UNet, which suggests that UNet may achieve better performance on individual crack images when the optimal threshold is applied for each image. However, the ODS of our model highlights its strong generalization capability across the entire dataset, effectively handling a wide range of crack

Table 5
Performance (mean ± std) of our crack segmentation model against state-of-the-art models.

Model name	Accuracy	Precision	Recall	F1 score (Dice)	IoU score
FCN	0.968 ± 0.0003	0.802 ± 0.0159	0.704 ± 0.0232	0.749 ± 0.0068	0.599 ± 0.0086
UNet	0.968 ± 0.0008	0.792 ± 0.0143	0.724 ± 0.0170	0.756 ± 0.0067	0.608 ± 0.0086
DeepCrack2	0.967 ± 0.0002	0.801 ± 0.0153	0.692 ± 0.0213	0.742 ± 0.0056	0.590 ± 0.0071
SegFormer	0.965 ± 0.0003	0.784 ± 0.0185	0.674 ± 0.0236	0.724 ± 0.0057	0.568 ± 0.0071
HrSegNet	0.968 ± 0.0006	0.806 ± 0.0174	0.696 ± 0.0239	0.746 ± 0.0077	0.595 ± 0.0098
Hybrid-Segmentor	**0.971 ± 0.0002**	**0.807 ± 0.0105**	**0.756 ± 0.0156**	**0.774 ± 0.0036**	**0.631 ± 0.0036**

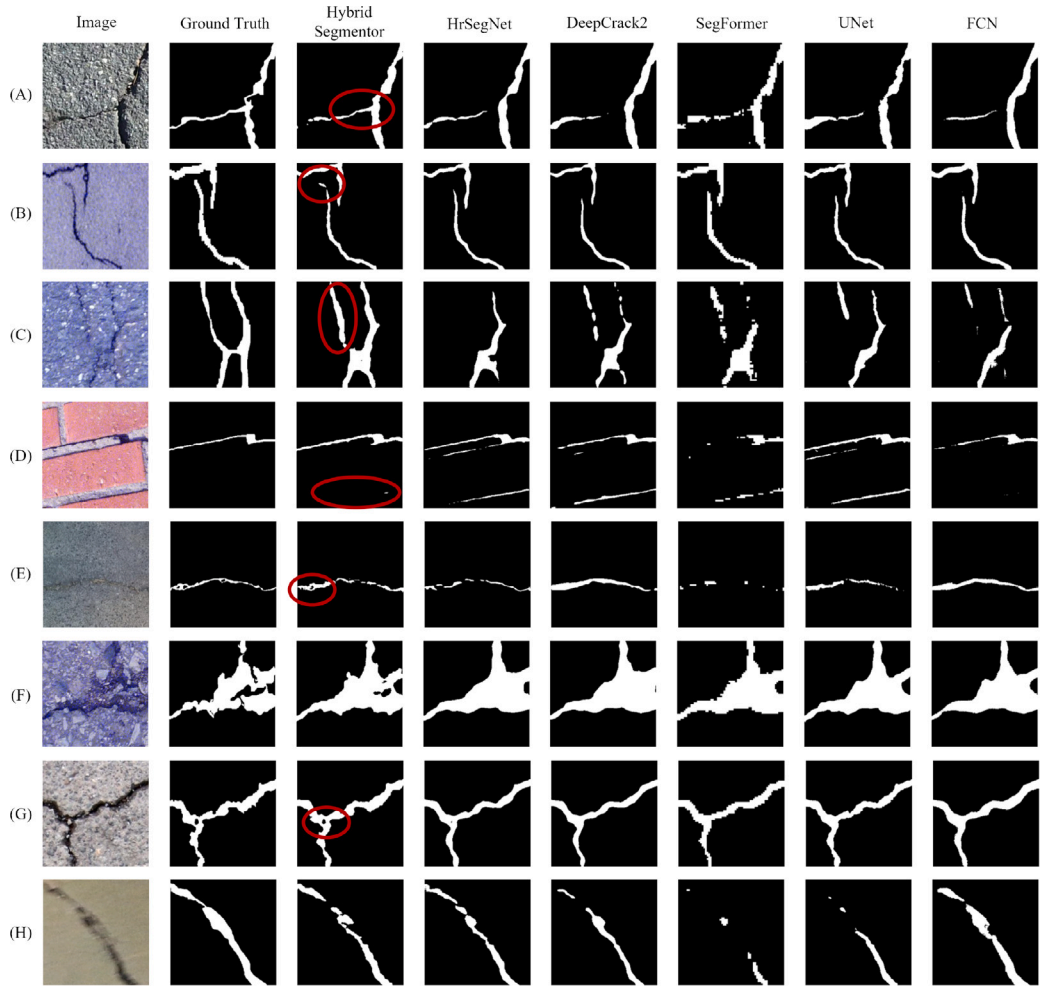

Fig. 5. Example crack images segmented by our model and benchmarked models. Red ovals highlight areas where our model outperforms other benchmarked models. In examples without red ovals, such as (F) and (H), our model demonstrates strong performance across overall structures.

Table 6
ODS and OIS of our crack segmentation model against state-of-the-art.

Metric	Hybrid segmentor	HrSegNet	DeepCrack2	SegFormer	UNet	FCN
ODS	**0.776**	0.765	0.751	0.737	0.761	0.759
OIS	0.586	0.581	0.567	0.543	**0.598**	0.586

shapes and distributions.

Qualitatively, our model shows significant improvements relative to existing models (Fig. 5). As shown in rows (A) and (C), our model handles crack discontinuity more accurately. Moreover, in (B), our model excels at identifying vague cracks that other models fail to detect. When it comes to cracks on different types of surfaces, the proposed model works effectively regardless of the surface. While crack detection on brick surfaces is challenging due to the ambiguity between cracks and brick borders and resulting shadows, as shown in (D), our model is adept at handling such scenarios. On the other hand, models such as FCN incorrectly predict brick borders as cracks. Additionally, a challenge in crack detection involves identifying non-crack areas within cracked regions, which our model effectively addresses, as evident in (E) and (G). Example (H) demonstrates that our model works relatively well on blurred images. Furthermore, (F) demonstrates the superiority of our model in detecting intricate crack contours compared to other models that have significantly more false positives.

8. Discussion and limitations

While previous research [15,17,54] on crack segmentation has made significant improvements, but certain gaps remain when it comes to model architecture and dataset reliability. Our research addresses both of these critical gaps through three focused approaches:

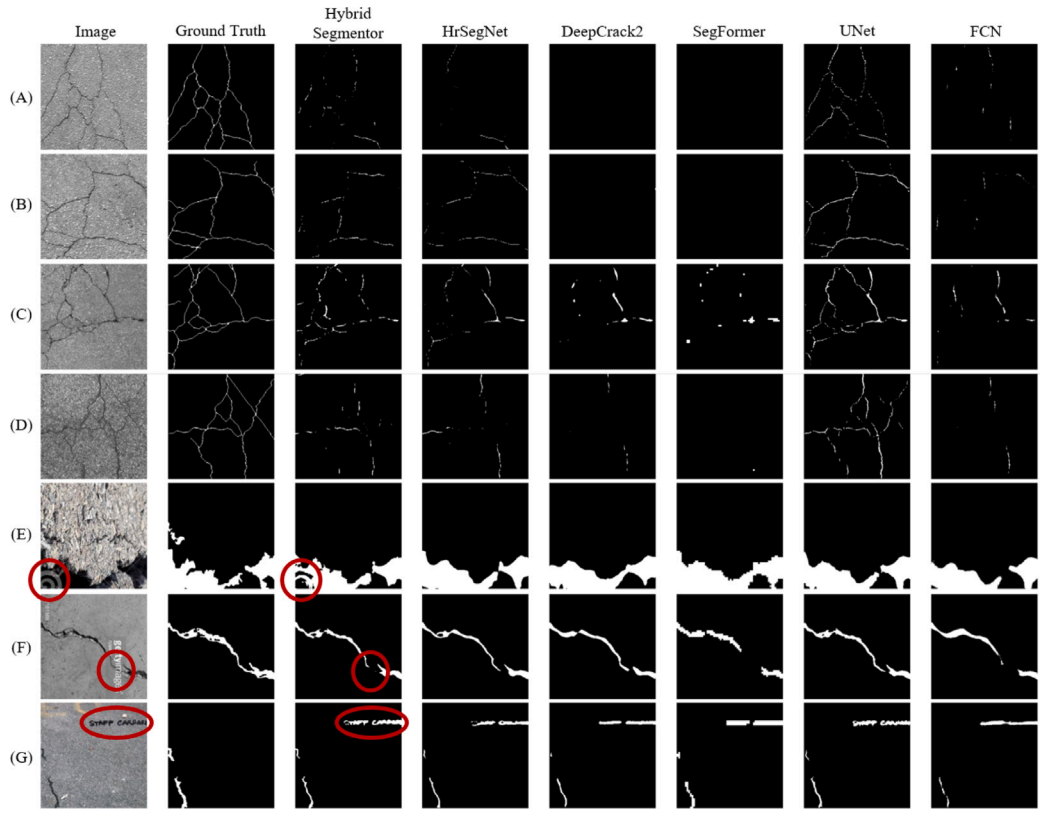

Fig. 6. Examples of Hybrid-Segmentor's limitations, including failure to detect thin or web-shaped objects and difficulties with occlusions.

- **Bridging the Gap in Feature Extraction:** There is a notable disparity in how CNN-based and transformer-based models handle feature extraction. CNNs excel at capturing local textural features, but often fail to grasp global contextual details crucial for understanding complex structures like cracks. Conversely, transformers effectively integrate global interactions but can overlook local details such as edges. This research bridged the gap by seamlessly integrating the strengths of both architectures. The resulting hybrid model ensures robust feature extraction that operates effectively across different levels.
- **Addressing Dataset Insufficiency:** A significant bottleneck in enhancing the precision of crack segmentation has been the insufficiency of datasets that accurately reflect the diversity of real world conditions. To overcome this, we combined multiple datasets with refinement methodology to create an extensive and varied dataset. Our dataset encompasses a wide range of surfaces and crack types, thereby improving training and enhancing the model's capacity to generalize across previously unseen or underrepresented conditions. This approach aims to fill existing research gaps and sets a new benchmark for future crack segmentation tasks.
- **Developing a Unified Model for Diverse Surfaces and Crack Types:** Most current models demonstrate limitations when applied to varying surfaces and crack morphologies, impacting their utility in real world applications. This research introduced a unified model designed to perform consistently well across diverse surfaces and crack types and was trained on the refined dataset of diverse surfaces and crack types.

Although our model outperforms other benchmarked models in overall performance, it still exhibits certain limitations. Two primary shortcomings of our model are identified and presented in Fig. 6.

- **Thinner Cracks Detection:** While our model excels at detecting thicker cracks within web-shaped crack patterns (outperforming other models except UNet), it faces challenges in identifying the finer branches in these patterns. As illustrated in the example images (A) to (D), although our model successfully detects the most prominent cracks, it struggles with extremely fine and delicate ones, indicating an area for improvement.
- **Sensitivity to Distortions:** Our model is sensitive to disruptions caused by distortions, such as occlusions and watermarks. In (E), a watermark located within a crack is incorrectly identified as a non-crack area. Meanwhile, in (F), even with the presence of a watermark, our model fails to predict cracks hidden by a translucent occlusion.
- **Confusion with Road Markings:** Additionally, (G) demonstrates an issue where the model fails to recognize letters on the road as part of the background. The confusion may arise from the high color contrast between the letters and the road surface. It should be noted that these challenges are common in crack detection models, but our model still demonstrates high precision and detail compared to others.

We believe that there is room for improvement in the model architecture to address these limitations. Techniques such as Generative Adversarial Networks (GAN) and meta-learning can be leveraged to generate synthetic data during the pre-processing phase, helping to address data scarcity and class imbalance. Furthermore, recognizing the growing importance of 3D crack image segmentation, creating high-quality 3D crack point cloud datasets is crucial to advance this field.

9. Conclusion

This paper described Hybrid-Segmentor, a crack segmentation model that combines a CNN path based on ResNet-50 and a Transformer path inspired by SegFormer. This dual-encoder architecture effectively captures both local and global crack features, achieving SOTA

performance in detecting fine details and cracks on diverse surfaces. Experiments demonstrate that the Hybrid-Segmentor outperforms other benchmark models, particularly in addressing discontinuities, detecting small non-cracked areas within cracks and working on blurred images.

The dataset was refined by merging 13 open-source crack datasets, standardizing ground truths, and tackling class imbalance with targeted data augmentation. This process yielded 12,000 reliable crack images with ground truth. Future work will focus on improving the detection of thin and occluded cracks, creating synthetic data using Generative Adversarial Networks (GAN), and incorporating 3D point cloud analysis to assess crack depth for better diagnostics.

CRediT authorship contribution statement

June Moh Goo: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Formal analysis, Data curation, Conceptualization. **Xenios Milidonis:** Writing – review & editing, Supervision, Funding acquisition, Data curation. **Alessandro Artusi:** Writing – review & editing, Supervision, Funding acquisition, Project administration. **Jan Boehm:** Writing – review & editing. **Carlo Ciliberto:** Writing – review & editing, Supervision.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the author(s) used Grammarly and Writefull to assist with grammar checks and refinement of sentence structure. After using this tool, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The research work of Dr. Alessandro Artusi and Dr. Xenios Milidonis has been partially funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 739578 and from the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy.

Data availability

I have shared the link on my manuscript.

References

[1] P.C. Chang, A. Flatau, S.C. Liu, Review paper: Health monitoring of civil infrastructure, Struct. Health Monit. 2 (3) (2003) 257–267, http://dx.doi.org/10.1177/1475921703036169.

[2] H. Kim, E. Ahn, S. Cho, M. Shin, S.-H. Sim, Comparative analysis of image binarization methods for crack identification in concrete structures, Cem. Concr. Res. 99 (2017) 53–61, http://dx.doi.org/10.1016/j.cemconres.2017.04.018, URL https://www.sciencedirect.com/science/article/pii/S000888461630881X.

[3] J.P. Lynch, C.R. Farrar, J.E. Michaels, Structural health monitoring: technological advances to practical implementations, Proc. IEEE 104 (8) (2016) 1508–1512, http://dx.doi.org/10.1109/JPROC.2016.2588818.

[4] A. Heidari, N. Jafari Navimipour, M. Unal, G. Zhang, Machine learning applications in internet-of-drones: Systematic review, recent deployments, and open issues, ACM Comput. Surv. 55 (12) (2023) http://dx.doi.org/10.1145/3571728.

[5] A. Razi, X. Chen, H. Li, H. Wang, B. Russo, Y. Chen, H. Yu, Deep learning serves traffic safety analysis: A forward-looking review, IET Intell. Transp. Syst. 17 (1) (2023) 22–71, http://dx.doi.org/10.1049/itr2.12257, arXiv:https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/itr2.12257, URL https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/itr2.12257.

[6] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, X. Yang, Automatic pixel-level crack detection and measurement using fully convolutional network, Comput.-Aided Civ. Infrastruct. Eng. 33 (12) (2018) 1090–1109, http://dx.doi.org/10.1111/mice.12412, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/mice.12412, URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12412.

[7] A. Zhang, K. Wang, Y. Fei, Y. Liu, C. Chen, G. Yang, J. Li, E. Yang, S. Qiu, Automated pixel-level pavement crack detection on 3D asphalt surfaces with a recurrent neural network: Automated pixel-level pavement crack detection on 3D asphalt surfaces using CrackNet-R, Comput.-Aided Civ. Infrastruct. Eng. 34 (2018) http://dx.doi.org/10.1111/mice.12409.

[8] Y. Liu, J. Yao, X. Lu, R. Xie, L. Li, DeepCrack: A deep hierarchical feature learning architecture for crack segmentation, Neurocomputing 338 (2019) 139–153, http://dx.doi.org/10.1016/j.neucom.2019.01.036.

[9] J. Cheng, W. Xiong, W. Chen, Y. Gu, Y. Li, Pixel-level crack detection using U-net, in: TENCON 2018 - 2018 IEEE Region 10 Conference, 2018, pp. 0462–0466, http://dx.doi.org/10.1109/TENCON.2018.8650059.

[10] O. Oktay, J. Schlemper, L.L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N.Y. Hammerla, B. Kainz, B. Glocker, D. Rueckert, Attention U-net: Learning where to look for the pancreas, 2018, arXiv:1804.03999.

[11] J. König, M.D. Jenkins, M. Mannion, P. Barrie, G. Morison, Optimized deep encoder-decoder methods for crack segmentation, Digit. Signal Process. 108 (2021) 102907, http://dx.doi.org/10.1016/j.dsp.2020.102907, URL https://www.sciencedirect.com/science/article/pii/S1051200420302529.

[12] D. Pal, P.B. Reddy, S. Roy, Attention UW-Net: A fully connected model for automatic segmentation and annotation of chest X-ray, Comput. Biol. Med. 150 (2022) 106083, http://dx.doi.org/10.1016/j.compbiomed.2022.106083, URL https://www.sciencedirect.com/science/article/pii/S0010482522007910.

[13] F. Panella, A. Lipani, J. Boehm, Semantic segmentation of cracks: Data challenges and architecture, Autom. Constr. 135 (2022) 104110, http://dx.doi.org/10.1016/j.autcon.2021.104110, URL https://www.sciencedirect.com/science/article/pii/S0926580521005616.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255, http://dx.doi.org/10.1109/CVPR.2009.5206848.

[15] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, S. Wang, DeepCrack: Learning hierarchical convolutional features for crack detection, IEEE Trans. Image Process. 28 (3) (2019) 1498–1512, http://dx.doi.org/10.1109/TIP.2018.2878966.

[16] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495, http://dx.doi.org/10.1109/TPAMI.2016.2644615.

[17] Y. Li, R. Ma, H. Liu, G. Cheng, Real-time high-resolution neural network with semantic guidance for crack segmentation, Autom. Constr. 156 (2023) 105112, http://dx.doi.org/10.1016/j.autcon.2023.105112, URL https://www.sciencedirect.com/science/article/pii/S0926580523003722.

[18] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, D. Tao, A survey on vision transformer, IEEE Trans. Pattern Anal. Mach. Intell. 45 (1) (2023) 87–110, http://dx.doi.org/10.1109/TPAMI.2022.3152247.

[19] C. Xiang, J. Guo, R. Cao, L. Deng, A crack-segmentation algorithm fusing transformers and convolutional neural networks for complex detection scenarios, Autom. Constr. 152 (2023) 104894, http://dx.doi.org/10.1016/j.autcon.2023.104894, URL https://www.sciencedirect.com/science/article/pii/S0926580523001541.

[20] X. Li, H. Ding, H. Yuan, W. Zhang, J. Pang, G. Cheng, K. Chen, Z. Liu, C.C. Loy, Transformer-based visual segmentation: A survey, IEEE Trans. Pattern Anal. Mach. Intell. (2024) 1–24, http://dx.doi.org/10.1109/TPAMI.2024.3434373.

[21] R. Azad, A. Kazerouni, B. Azad, E. Khodapanah Aghdam, Y. Velichko, U. Bagci, D. Merhof, Laplacian-former: Overcoming the limitations of vision transformers in local texture detection, in: Medical Image Computing and Computer Assisted Intervention – MICCAI 2023, Springer Nature Switzerland, 2023, pp. 736–746, http://dx.doi.org/10.1007/978-3-031-43898-1_70.

[22] P. Wang, W. Zheng, T. Chen, Z. Wang, Anti-oversmoothing in deep vision transformers via the Fourier domain analysis: From theory to practice, in: International Conference on Learning Representations, 2022, URL https://openreview.net/forum?id=O476oWmiNNp.

[23] N. Kheradmandi, V. Mehranfar, A critical review and comparative study on image segmentation-based techniques for pavement crack detection, Constr. Build. Mater. 321 (2022) 126162, http://dx.doi.org/10.1016/j.conbuildmat.2021.126162, URL https://www.sciencedirect.com/science/article/pii/S0950061821038940.

[24] S. Zhou, C. Canchila, W. Song, Deep learning-based crack segmentation for civil infrastructure: data types, architectures, and benchmarked performance, Autom. Constr. 146 (2023) 104678, http://dx.doi.org/10.1016/j.autcon.2022.104678, URL https://www.sciencedirect.com/science/article/pii/S0926580522005489.

[25] S. Dorafshan, R.J. Thomas, M. Maguire, SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks, Data Brief 21 (2018) 1664–1668, http://dx.doi.org/10.1016/j.dib.2018.11.015, URL https://www.sciencedirect.com/science/article/pii/S2352340918314082.

[26] R. Amhaz, S. Chambon, J. Idier, V. Baltazart, Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection, IEEE Trans. Intell. Transp. Syst. 17 (10) (2016) 2718–2729, http://dx.doi.org/10.1109/TITS.2015.2477675.

[27] L. Zhang, F. Yang, Y. Daniel Zhang, Y.J. Zhu, Road crack detection using deep convolutional neural network, in: 2016 IEEE International Conference on Image Processing, ICIP, 2016, pp. 3708–3712, http://dx.doi.org/10.1109/ICIP.2016.7533052.

[28] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, S. Wang, DeepCrack: Learning hierarchical convolutional features for crack detection, IEEE Trans. Image Process. 28 (3) (2019) 1498–1512, http://dx.doi.org/10.1109/TIP.2018.2878966.

[29] Q. Zou, Y. Cao, Q. Li, Q. Mao, S. Wang, CrackTree: Automatic crack detection from pavement images, Pattern Recognit. Lett. 33 (3) (2012) 227–238, http://dx.doi.org/10.1016/j.patrec.2011.11.004, URL https://www.sciencedirect.com/science/article/pii/S0167865511003795.

[30] M. Eisenbach, R. Stricker, D. Seichter, K. Amende, K. Debes, M. Sesselmann, D. Ebersbach, U. Stoeckert, H.-M. Gross, How to get pavement distress detection ready for deep learning? A systematic approach, in: 2017 International Joint Conference on Neural Networks, IJCNN, 2017, pp. 2039–2047, http://dx.doi.org/10.1109/IJCNN.2017.7966101.

[31] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, H. Ling, Feature pyramid and hierarchical boosting network for pavement crack detection, IEEE Trans. Intell. Transp. Syst. 21 (4) (2020) 1525–1535, http://dx.doi.org/10.1109/TITS.2019.2910595.

[32] D. Dais, I.E. Bal, E. Smyrou, V. Sarhosis, Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning, Autom. Constr. 125 (2021) 103606, http://dx.doi.org/10.1016/j.autcon.2021.103606, URL https://linkinghub.elsevier.com/retrieve/pii/S0926580521000571.

[33] Y. Shi, L. Cui, Z. Qi, F. Meng, Z. Chen, Automatic road crack detection using random structured forests, IEEE Trans. Intell. Transp. Syst. 17 (12) (2016) 3434–3445, http://dx.doi.org/10.1109/TITS.2016.2552248.

[34] L. Cui, Z. Qi, Z. Chen, F. Meng, Y. Shi, Pavement distress detection using random decision forests, in: International Conference on Data Science, Springer, 2015, pp. 95–102, http://dx.doi.org/10.1007/978-3-319-24474-7_14.

[35] S. Kulkarni, S. Singh, D. Balakrishnan, S. Sharma, S. Devunuri, S.C.R. Korlapati, CrackSeg9k: A Collection and Benchmark for Crack Segmentation Datasets and Frameworks, in: L. Karlinsky, T. Michaeli, K. Nishino (Eds.), Computer Vision – ECCV 2022 Workshops, Springer Nature Switzerland, Cham, 2023, pp. 179–195, http://dx.doi.org/10.1007/978-3-031-25082-8_12.

[36] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778, http://dx.doi.org/10.1109/CVPR.2016.90.

[37] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J.M. Alvarez, P. Luo, SegFormer: Simple and efficient design for semantic segmentation with transformers, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J.W. Vaughan (Eds.), in: Advances in Neural Information Processing Systems, vol. 34, Curran Associates, Inc., 2021, pp. 12077–12090, URL https://proceedings.neurips.cc/paper_files/paper/2021/file/64f1f27bf1b4ec22924fd0acb550c235-Paper.pdf.

[38] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16 × 16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2021, URL https://openreview.net/forum?id=YicbFdNTTy.

[39] Q. Zhou, Z. Qu, S.-Y. Wang, K.-H. Bao, A method of potentially promising network for crack detection with enhanced convolution and dynamic feature fusion, IEEE Trans. Intell. Transp. Syst. 23 (10) (2022) 18736–18745, http://dx.doi.org/10.1109/TITS.2022.3154746.

[40] H. Liu, J. Yang, X. Miao, C. Mertz, H. Kong, CrackFormer network for pavement crack segmentation, IEEE Trans. Intell. Transp. Syst. 24 (9) (2023) 9240–9252, http://dx.doi.org/10.1109/TITS.2023.3266776.

[41] V. Polovnikov, D. Alekseev, I. Vinogradov, G.V. Lashkia, DAUNet: Deep augmented neural network for pavement crack segmentation, IEEE Access 9 (2021) 125714–125723, http://dx.doi.org/10.1109/ACCESS.2021.3111223.

[42] Z. Qu, W. Chen, S.-Y. Wang, T.-M. Yi, L. Liu, A crack detection algorithm for concrete pavement based on attention mechanism and multi-features fusion, IEEE Trans. Intell. Transp. Syst. 23 (8) (2022) 11710–11719, http://dx.doi.org/10.1109/TITS.2021.3106647.

[43] H. Zhang, G. Yang, H. Li, W. Du, J. Wang, Pixel-wise detection algorithm for crack structural reconstruction based on rock CT images, Autom. Constr. 152 (2023) 104895, http://dx.doi.org/10.1016/j.autcon.2023.104895, URL https://www.sciencedirect.com/science/article/pii/S0926580523001553.

[44] M. Abdellatif, H. Peel, A.G. Cohn, R. Fuentes, Combining block-based and pixel-based approaches to improve crack detection and localisation, Autom. Constr. 122 (2021) 103492, http://dx.doi.org/10.1016/j.autcon.2020.103492, URL https://www.sciencedirect.com/science/article/pii/S0926580520310724.

[45] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE Computer Society, Los Alamitos, CA, USA, 2015, pp. 3431–3440, http://dx.doi.org/10.1109/CVPR.2015.7298965, URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2015.7298965.

[46] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer International Publishing, 2015, pp. 234–241, http://dx.doi.org/10.1007/978-3-319-24574-4_28.

[47] W. Zhao, Y. Liu, J. Zhang, Y. Shao, J. Shu, Automatic pixel-level crack detection and evaluation of concrete structures using deep learning, Struct. Control Health Monit. 29 (8) (2022) e2981, http://dx.doi.org/10.1002/stc.2981, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/stc.2981, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/stc.2981.

[48] R. Guo, J. Stubbe, Y. Zhang, C.M. Schlepütz, C.R. Gomez, M. Mehdikhani, C. Breite, Y. Swolfs, P. Villanueva-Perez, Deep-learning image enhancement and fibre segmentation from time-resolved computed tomography of fibre-reinforced composites, Compos. Sci. Technol. 244 (2023) 110278, http://dx.doi.org/10.1016/j.compscitech.2023.110278, URL https://www.sciencedirect.com/science/article/pii/S026635382300372X.

[49] P. Chen, P. Li, B. Wang, X. Ding, Y. Zhang, T. Zhang, T. Yu, GFSegNet: A multi-scale segmentation model for mining area ground fissures, Int. J. Appl. Earth Obs. Geoinf. 128 (2024) 103788, http://dx.doi.org/10.1016/j.jag.2024.103788, URL https://www.sciencedirect.com/science/article/pii/S1569843224001420.

[50] Q.D. Nguyen, H.-T. Thai, Crack segmentation of imbalanced data: The role of loss functions, Eng. Struct. 297 (2023) 116988, http://dx.doi.org/10.1016/j.engstruct.2023.116988, URL https://www.sciencedirect.com/science/article/pii/S0141029623014037.

[51] C.H. Sudre, W. Li, T. Vercauteren, S. Ourselin, M. Jorge Cardoso, Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, QuÉBec City, QC, Canada, September 14, Proceedings 3, Springer, 2017, pp. 240–248, http://dx.doi.org/10.1007/978-3-319-67558-9_28.

[52] J. Tian, N.C. Mithun, Z. Seymour, H.-P. Chiu, Z. Kira, Striking the right balance: Recall loss for semantic segmentation, in: 2022 International Conference on Robotics and Automation, ICRA, IEEE Press, 2022, pp. 5063–5069, http://dx.doi.org/10.1109/ICRA46639.2022.9811702.

[53] M. Yi-de, L. Qing, Q. Zhi-bai, Automated image segmentation using improved PCNN model based on cross-entropy, in: Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004, pp. 743–746, http://dx.doi.org/10.1109/ISIMP.2004.1434171.

[54] C. Han, T. Ma, J. Huyan, X. Huang, Y. Zhang, Crackw-net: A novel pavement crack image segmentation convolutional neural network, IEEE Trans. Intell. Transp. Syst. 23 (11) (2022) 22135–22144, http://dx.doi.org/10.1109/TITS.2021.3095507.