Towards Vision-based Concrete Crack Detection: Automatic Simulation of Real-world Cracks

Tran Hiep Dinh, *Member, IEEE,* Vu Thi Thuy Anh, TruongGiang Nguyen, Cong Hieu Le,
Nguyen Linh Trung, *Senior Member, IEEE,* Nguyen Dinh Duc, Chin-Teng Lin, *Fellow, IEEE*

***Abstract*—Vision-based concrete crack detection has recently attracted significant attention from many researchers. Although promising results have been obtained, especially for deep learning approaches, it is difficult to maintain the robustness of implemented models when tested on completely new data. A possible reason for this is that the extracted feature from the trained set might not fully characterize the crack in the test set. We propose an interdisciplinary approach to improve the effectiveness of vision-based crack detection by modeling crack propagation using fracture mechanics, simulation, and machine learning. Mathematical models of concrete cracks are obtained using machine learning on the simulation results. Experiments are conducted on various reputable crack image datasets, emphasizing the correlation between simulated and real-world cracks. The importance of propagation models is verified in a classification task, reporting a significant accuracy enhancement on results of some state-of-the-art detection and segmentation models, i.e. 1.27% on average on participating models, and 5.47% on U-Net. This novel approach is expected to have valuable points for a research area where the data quantity and quality still need to be improved.**

***Index Terms*—Keywords: Crack dataset, regression model, crack propagation, crack detection, concrete beam, Extended Finite Element Method (X-FEM), machine learning, vision-based inspection.**

I. Introduction

Concrete is one of the most popular and long-standing materials that has been used in construction for a long time. However, like other materials, concrete also has certain disadvantages such as poor tensile strength, shrinkage, that affect the durability of the structure. The formation and development of cracks in concrete can be caused by many reasons, the most common two of which are: chemical, environmental factors in the concrete setting and errors in the usage process of concrete works (such as traffic overload with bridges, houses...). When cracks occur, chloride and other corrosive substances penetrate the concrete through the cracks, leading to corrosion of the internal reinforcement and reduction of the cross-section. This deterioration can affect the flexural and shear strength of the structure, causing unfortunate incidents. In addition, widespread cracks also significantly affect the aesthetics of the structure. Therefore, early detection of cracks and prediction of their propagation is essential to help minimize potential problems, increase construction life, reduce maintenance and repair costs.

In most countries until now, crack or defect detection is still being done manually and carried out by construction inspection units. The process is costly, time-consuming, yet not accurate. Recently, with the advances in image processing, Machine Learning (ML) and Deep Learning (DL), vision-based crack detection are increasingly attracting the attention of many researchers. When inspecting large infrastructures, vision-based techniques outperform human-powered ones thanks to their outstanding advantages in terms of execution time and safety [1], [2]. Recent vision-based studies can be mainly divided into three groups:

(i) group of methods based on thresholding [3]–[5],
(ii) group of methods based on ML [6]–[8],
(iii) group of methods based on DL [9]–[18].

Although satisfactory results have been obtained from the above methods, the fundamental of which is either based on naked eye observation of the difference in the gray level, or shape between objects and the background [3]–[8], or on features extracted from deep learning models [10]–[18], the accuracy of which depends on the training dataset's quality and size [19].

Thresholding methods extract crack pixels based on the local intensity difference between the interested object and background. As discussed in [20], bad lighting conditions could severely impact the distinction between crack and noise pixels, possibly causing a mis-detection. ML-based methods use handcrafted descriptors where neighborhood information is leveraged at a single or multi-scale level to train classifiers such as k-Nearest Neighbor (KNN) or Support Vector Machine (SVM) for crack detection. As pointed out in [21], those local methods are vulnerable to complicated scenes where

Corresponding author: Tran Hiep Dinh.

Tran Hiep Dinh was funded by the Postdoctoral Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2022.STS.07. He is with the University of Engineering and Technology, Vietnam National University, Hanoi, 144 Xuan Thuy road, Cau Giay, Hanoi, Vietnam. He is also with the Joint Technology and Innovation Research Centre (JTIRC), a partnership between the University of Technology Sydney and the VNU University of Engineering and Technology. Email: tranhiep.dinh@vnu.edu.vn.

Vu Thi Thuy Anh and Nguyen Dinh Duc are with the Faculty of Construction and Transportation Engineering, University of Engineering and Technology, Vietnam National University, Hanoi, 144 Xuan Thuy road, Cau Giay, Hanoi, Vietnam. Email: {anhvutt, ducnd}@vnu.edu.vn.

TruongGiang Nguyen is with the Institute of Mechanics, Vietnam National Academy of Science and Technology, 264 Doi Can, Ba Dinh, Hanoi, Vietnam. Email: ntgiang@imech.vast.vn.

Cong Hieu Le and Nguyen Linh Trung are with the Advanced Institute of Engineering and Technology (AVITECH), University of Engineering and Technology, Vietnam National University, Hanoi, 144 Xuan Thuy road, Cau Giay, Hanoi, Vietnam Email: {18020508, linhtrung}@vnu.edu.vn.

Chin-Teng Lin is with the Australian Artificial Intelligence Institute, School of Computer Science, Faculty of Engineering and Information Technology, University of Technology Sydney, 15 Broadway, Ultimo NSW 2007, Sydney, Australia. Email: chin-teng.lin@uts.edu.au.

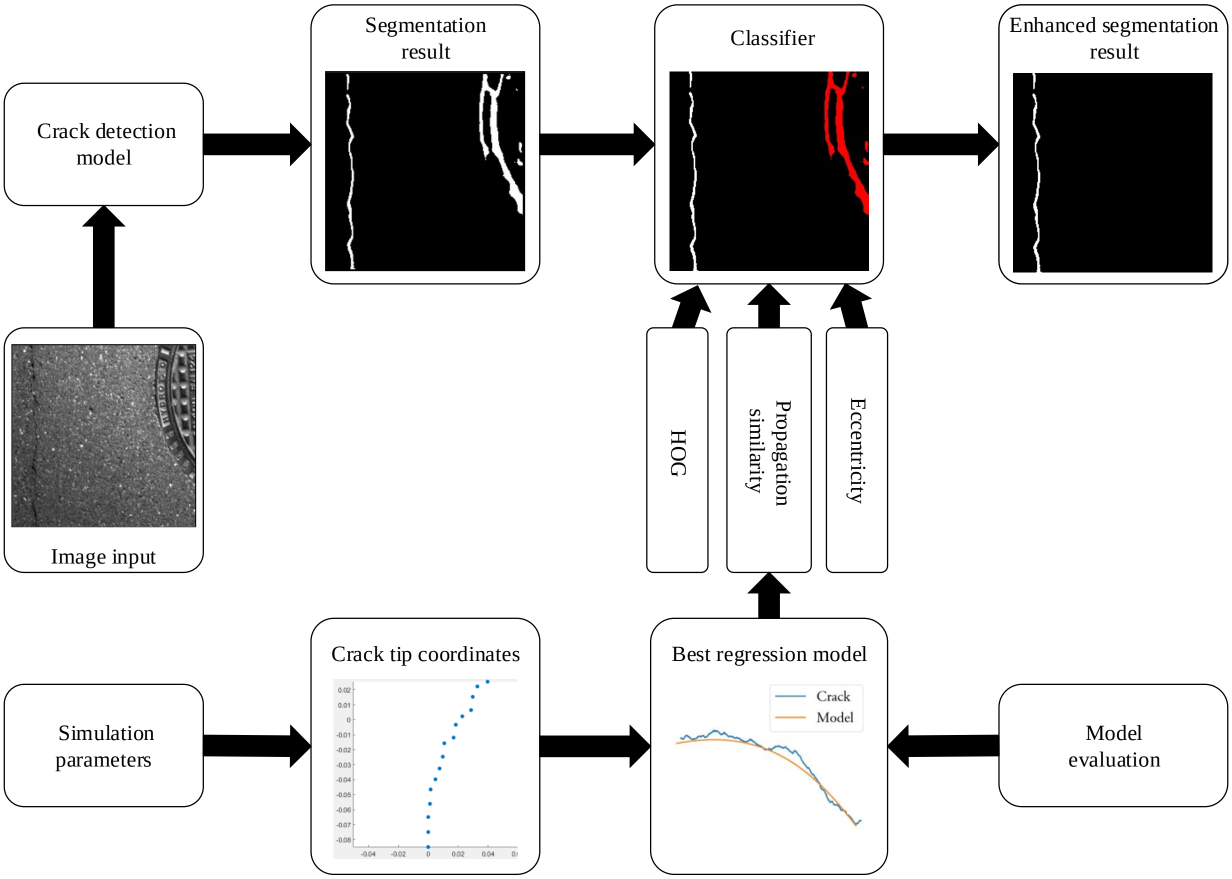


Fig. 1: Overview of the proposed post-processing.

the crack appearance in images are affected by low contrast or discontinuity. Recently, DL-based approaches for crack detection have attracted great attention and yielded promising results, thanks to their capability to extract high level features representing crack for model training. Proposed techniques are mainly focused on the optimization of network architecture to deal with the variations in crack representation, such as its continuity [21], aspect ratio [9], or multi-scale representation [12]. However, some challenges are remained, including the collection of crack image data of sufficient quality and quantity to maintain the well-performance of models on a certain target [9].

Meanwhile, the propagation of crack in materials has been well studied in fracture mechanics [22] where the materials' behavior under stresses is characterized via analytical and experimental solid mechanics. Thus, we propose in this paper an interdisciplinary approach incorporating fracture mechanics into vision-based crack detection. Our hypothesis is that, through analyzing the crack propagation from a theoretical perspective, we could come up with a model to improve the crack detection accuracy. Crack modeling has been considered in several approaches [23]–[25], where either a particular mathematical model is applied for the proposed detection method [23], [24] or a synthetic dataset is generated from the crack image taken by varying the width of each segment [25]. However, to the best of our knowledge, the formation and propagation of concrete cracks have yet to be considered in any vision-based detection method, although this is an important source of information.

Our approach in this paper is proposed as follows: First, based on fracture mechanics, and simulation by X-FEM [26], [27], a set of synthetic crack will be simulated to create a large data set for modeling mathematical representation of cracks. A three-point bending experiment will be conducted to create a crack on a real beam for verification of the proposed approach, where the best fit model for the annotated crack at different scales will be selected and analyzed. Then, the correlation between the obtained regression models representing simulated and real-world cracks is verified via some reputable crack image datasets. The best regression model is then selected as a benchmark to measure the similarity between segmentation results of some state-of-the-art crack detection/segmentation models. Other features characterizing the thinnes, linearity, and intensity distinguish of a crack are also employed to feed into a classifier to further enhance segmentation results of participated models on a challenging crack image dataset. Since it is difficult to find a training data of sufficient quality and quantity [9], the performance of the trained model can be impacted when tested on images with crack-like objects. Fig. 1 presents an overview of our proposed post-processing approach, where false positives (represented in red) of a detection model can be classified and removed effectively. The contributions of the paper can be summarized as follows:

(i) A novel idea of incorporating and implementing the knowledge of fracture mechanics in the vision-based crack detection problem.

(ii) An automatic simulation technique to create a non-limited number of simulated crack tip coordinates, the dataset of which will be made public.

(iii) Comprehensive experiments showing a high level of correlation between simulated and real-world cracks.

(iv) A post-processing approach enhancing the detection accuracy of current state-of-the-art models.

The rest of the paper is organized as follows: Section II presents the related techniques for simulation of the concrete crack. Section III describes our approach in an automatic simulation of the concrete crack. Experimental results, parameter analysis, and remarks are reported and discussed respectively in Sections IV, V and VI. Finally, Section VII draws the paper conclusion.

II. Related Techniques

Recent advances in computational engineering have eased the difficulty in the process of crack simulation, leading to more accurate and reliable solutions in the prediction of crack propagation. In the last decade, studies on cracks based on fracture mechanics can be divided into two groups:

(i) smeared crack-based approaches where the concrete strength and stiffness are adjusted or the dissipated energy is smeared to model the crack [28], [29],

(ii) discrete crack-based approaches, where a discontinuity interface is inserted into the object to describe its behavior using the discrete traction-separation law [30], [31].

Although the Finite Element Method (FEM) [28]–[31] is employed in both aforementioned groups, it has been noted in [32] that FEM is only suitable for studying fixed cracks, due to the placement of discontinuity interfaces at element boundaries in each finite element mesh. In the case of representation of crack propagation, FEM is no longer effective since re-meshing within the object is required.

Besides, the Phase-Field Method (PFM), also known as the method of establishing a variation model for crack propagation, is also very popular recently and can overcome the limitations of FEM. Unlike the X-FEM method, in view of this model, the propagation of a crack in the structure is considered as the minimization of the total potential energy of the system [33]–[35]. However, to simulate crack propagation using the PFM, the meshing is required to be very thorough to ensure the accuracy of the results.

The Strong Discontinuity Approach (SDA) [36] has become more popular in numerical simulation of the crack in recent years. Here, a macroscopic crack is described as a discontinuity in the displacement field based on the kinematics of a strong discontinuity. However, in contrast to the discrete crack model, displacement discontinuities are not restricted into the element boundary but are rather introduced within the element domain. Therefore, re-meshing in the object is not necessary. In SDA, the employment of finite element formulations can be categorized into three groups: the X-FEM-based method [26], the concept of elements with embedded discontinuities [36], or hybrid approaches combining the former two [37], [38].

Another numerical method is also widely applied in crack simulation problems, which is the Numerical Manifold Method (NMM) [39], [40]. This method can deal with existing problems when using FEM, such as continuity and discontinuity, in a unified way [41], [42]. Simultaneously, in NMM, pre-existing cracks are handled easily without applying the level set method [43] like when using the X-FEM. However, this method is somewhat complicated and mainly applied to geotechnical problems.

Significantly, the development of different methods based on fracture mechanics theory for crack propagation study has obtained remarkable results. At the same time, taking the advantages and disadvantages of the aforementioned methods into account, the X-FEM method can be considered as a reasonable solution for simulating crack propagation in concrete structures [27].

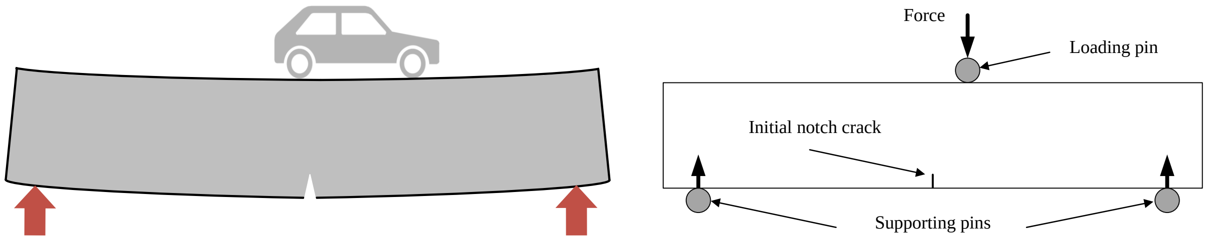
Notably, each simulation only applies to one type of model with specific boundary conditions. In crack analysis, when a single parameter change is required, such as the force position, all setting parameters in the simulation have to be re-entered in the simulation software, which is not desirable for the generation of a large number of samples. To carry out a sufficiently large number of simulations for training purposes, an automatic simulation approach is required so that multiple crack models can be generated.

An interface named Abaqus2MATLAB (A2M), connecting the well-known finite element package, Abaqus, and the popular programming platform, MATLAB, has been developed in [44]. Although A2M has provided a framework for finite element post-processing, two aspects are still untouched when dealing with crack analysis: (i) only fixed cracks are simulated, and (ii) coordinates of crack tips are not extractable.

Inspired by A2M and to overcome the mentioned challenges, we propose in this paper an automatic simulation approach where the propagation path of the simulated crack is taken into consideration and coordinates of crack tips can be extracted. Experiments on crack images obtained from a bending experiment and some reputable datasets will be conducted to verify the correlation between simulation models and real-world crack.

III. Automatic Simulation of Concrete Cracks

There are many different methods to conduct simulations or tests to determine crack propagation in concrete, each using a different sample type and test procedure corresponding to a different fracture analysis model. Experimental samples can be three or four-point bending beam, sample double cantilever, double helix, compact tensile, plate, or block [45]. As pointed out in [46], a three-point bending beam is the simplest and most effective model, especially suitable for flexural structures such as beams or slabs. This is also considered appropriate for simulating the impact of vehicles on bridges or roads in reality (Fig. 2(a)), where crack propagation occurs and can be analyzed via a three-point flexural beam with an initial notch crack (Fig. 2(b)) as noted in [47].



(a)

(b)

Fig. 2: Demonstration of the impact of a vehicle on a beam (a) and its corresponding force diagram (b).

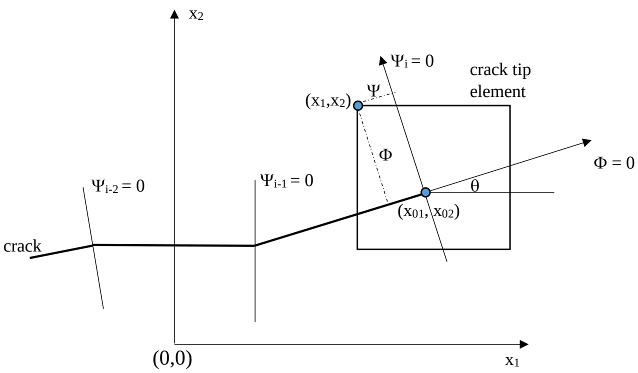


Fig. 3: Relationship between level set and crack tip coordinates.

For an automatic simulation approach of concrete crack, two problems are addressed in this section:

(i) First, concrete beams under the impact of load are simulated using X-FEM in Abaqus, in which the Level Set-based Method (LSM) for extraction of crack tip coordinates are employed to describe and track the crack growth. Changes in the simulation can be made via adjusting the input file, the result of which will be output to MATLAB for further processing.

(ii) The simulation process is then automated in MATLAB where numerous simulations along with the crack propagation paths can be generated. The result of this process is a data set of crack vertices for further processing using ML in the next section.

For the first problem, LSM has been widely used for tracking moving surfaces [48], [49], and they are naturally coupled in the framework of X-FEM. When using the LSM in X-FEM to represent a crack, two level sets are required: one to describe the crack surface, $\Phi(\mathbf{x}, t)$, and another one, $\Psi_i(\mathbf{x}, t)$, is constructed so that the intersection of two-level sets gives the crack front.

How these functions are formulated is well explained in [49], in which the crack is first defined as a set

$$\{\mathbf{x} : \Phi(\mathbf{x}, t) = 0 \text{ and } \Psi_i(\mathbf{x}, t) \leq 0\}, \tag{1}$$

where

$$\Phi(\mathbf{x}, t) = \sum \Phi_j(t) N_j(\mathbf{x}), \tag{2}$$

$$\Psi_i(\mathbf{x}, t) = \sum \Psi_{ij}(t) N_j(\mathbf{x}), \tag{3}$$

and the level set functions that represent the crack tip are initially defined by

$$\Psi_i(\mathbf{x}, 0) = (\mathbf{x} - \mathbf{x}_i)\mathbf{t}. \tag{4}$$

In the above equations, i is the number of tips on a given crack; N_j is the shape function associated with node j; t is the time; Φ_j, Ψ_{ij} are the nodal data of level set; $\mathbf{t}$ is a unit vector tangent to the crack at the tip, and $\mathbf{x}_i$ is the location of the i-th crack tip. Given the construction described in Eq. (4), the planar function Ψ_i has a zero level set which is orthogonal to Φ at the crack tip.

By manipulating the input file after simulation in Abaqus, the Φ_j, Ψ_{ij} values can be extracted and stored in the output file. In this paper, the output is further expanded to generate crack tip coordinates for automatic simulation of numerous samples via a connection between Abaqus and MATLAB. First, the following command is added to the MATLAB script file to call Abaqus within MATLAB: `system(abaqus job =` *`job_name`* `input =` *`input_file_name.inp`* `interactive)`. After that, the values Φ_j, Ψ_{ij} can be obtained in the output file `.dat` of Abaqus.

Once the values of Φ_j, Ψ_{ij} have been exported to the output file of Abaqus, crack tip coordinates can be calculated as follows:

(i) If the level set $\Phi = 0$, $\Psi_i = 0$ is considered as the local coordinate system at the crack tip, then every node can be represented in 2 coordinates, the global $\mathbf{x} = (x_1, x_2)$ and the level sets coordinate (Φ, Ψ_i).

(ii) The relationship between these two coordinate systems is represented in Fig. 3 where the coordinates of the crack tip are expressed as:

$$x_{01} = x_1 - \Psi_i \cos(\theta) - \Phi \sin(\theta), \tag{5}$$

$$x_{02} = x_2 + \Psi_i \sin(\theta) - \Phi \cos(\theta). \tag{6}$$

To create the interface that connects to MATLAB for automating the process of exporting $\Phi(\mathbf{x}, t)$, $\Psi_i(\mathbf{x}, t)$, and crack tip coordinates, a MATLAB script containing a `for` loop is executed where a predefined number of Abaqus analysis n is required. In each iteration of the loop, the following steps are conducted. First, the Abaqus input file of the k-th analysis is generated and launched and resulted in two level sets

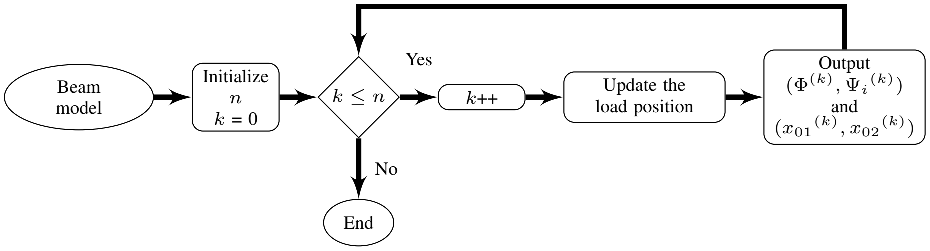


Fig. 4: Pipeline of the automatic simulation process.

$(\Phi^{(k)}, {\Psi_i}^{(k)})$. Then, the crack tip coordinates $({x_{01}}^{(k)}, {x_{02}}^{(k)})$ at increments corresponding to the force position are calculated according to Eqs. (5, 6). The load position is then updated for the next iteration. The number of increments in each crack model and that of crack models are respectively dependent on the number of the crack tips in the crack propagation path as well as the number of changes in the force position. The automatic simulation process is demonstrated in Fig. 4. An example of the simulation on a specific beam is presented in Section IV-B.

In the next section, the correlation of the generated simulation models and real crack images will be evaluated, including experimental results from a beam image and some reputable pavement crack image datasets.

IV. Results

In this section, cracks of a concrete beam are first simulated and then approximated by regression models in Python. The obtained model families, denoted as CRKMDL, is then utilized to fit to crack generated from a bending experiment and that from some reputable image datasets. In the bending experiment, a concrete beam is casted and bended to create crack using similar parameters as per the simulation. In the correlation analysis between CRKMDL and real-world cracks, the following reputable image datasets are employed: CFD [7], Crack500 [11], CRKWH100, CrackTree260, Stone331, and LS315 [12]. The similarity to the simulated propagation model is then employed in a post-processing classifier for further enhancement of detection results on the GAPs384 dataset [50] obtained from some recent effective crack detection models. Details of the employed datasets are summarized as follows:

- CFD consisting of 118 road crack images of size 480 x 320 taken by a cell phone,
- Crack500 consisting of 500 pavement crack images of size 2000 x 1500 taken by cell phones,
- CRKWH100 consisting of 100 road pavement images of size 512 x 512 taken by an area-array camera,
- CrackTree260 consisting of 260 road pavement images of size 960 x 720 taken by a line-aray camera,
- Stone331 consisting of 331 images of stone surface of size 512 x 512 taken by a line-array camera,
- LS315 consisting 315 of road pavement images of size 512 x 512 taken by an area-camera.
- GAPs384 consisting of 509 HD images that were cut and split from 380 HD images of size 1920 × 1080 from the German Asphalt Pavement Distress dataset with a resolution of 1.2 mm × 1.2 mm per pixel.

TABLE I: SIMULATION PARAMETERS

Parameters	Value
Size (cm)	80 × 20 × 20
Initial load position	5 cm from the middle of the beam
Load magnitude (kN)	45
Number of steps	1000
Step size s (cm)	0.01
Force position range L (cm)	10
Material	concrete
Initial notch crack length d (cm)	1

Since the proposed automatic simulation addresses only the propagation of a single crack, images with interconnected cracks from the above datasets are not considered in this study.

A. Evaluation Metrics

To evaluate the performance of the models, the following metrics are employed: the Root Mean Squared Error (*RMSE*), the Mean Absolute Error (*MAE*), and the Coefficient of Determination R^2. Let x_{02}, $\hat{x}_{02}$ respectively be the actual and estimated crack tip coordinate, N the number of tips in each crack, the aforementioned metrics can be calculated as follows:

$$\mathit{RMSE} = \sqrt{\frac{1}{N}\sum_{s=1}^{N}({x_{02}}_s - \hat{x}_{02})^2}, \tag{7}$$

$$\mathit{MAE} = \frac{1}{N}\sum_{s=1}^{N}|{x_{02}}_s - \widehat{x_{02}}|, \tag{8}$$

$$R^2 = 1 - \frac{\sum_{s=1}^{N}({x_{02}}_s - \hat{x}_{02})^2}{\sum({x_{02}}_s - \overline{x}_{02})^2}. \tag{9}$$

B. Results on Simulation Data

In this section, simulations of a beam of size 80 × 20 × 20 cm is conducted. A load of magnitude 45 kN is initially placed 5 cm from the middle of the beam and shifted n steps to the right within a range of L cm from the initial position. An initial notch crack of length d cm is made at the opposite surface of the beam where the load is applied. Although those parameters are adjustable, it is worth noting that if the the step size s is too small, the cracks might appear similar to each other and hence might not be beneficial for further processing. The parameters and the diagram of the simulation are respectively presented in Table I and Fig. 5.

Fig. 6 illustrates segmentation results when the load is

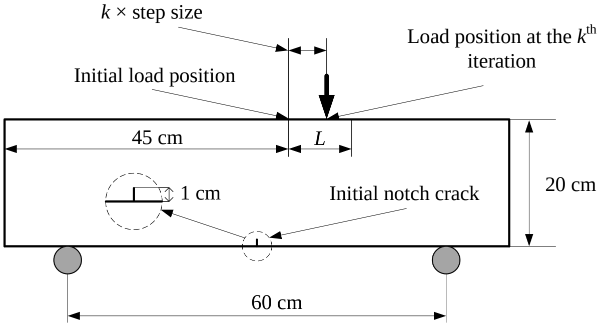
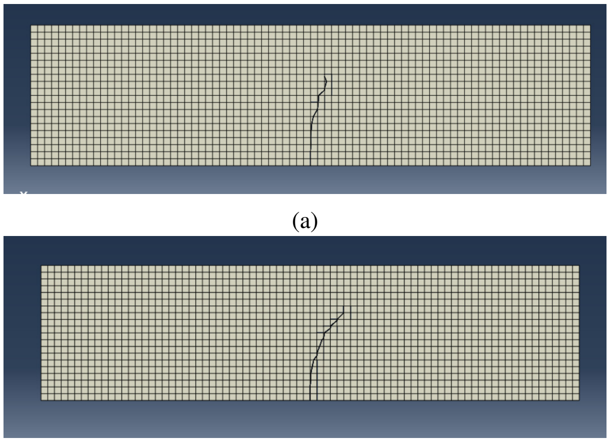


Fig. 5: Diagram of the simulation.

TABLE II: PERFORMANCE OF THE TOP REGRESSION MODEL FAMILIES ON THE SIMULATED CRACKS

Model	Variant	$\overline{RMSE}\downarrow$	$\overline{MAE}\downarrow$	$\overline{R^2}\uparrow$	$\overline{MS}\downarrow$
LM	PF of degree 1	**0.01222**	0.00954	0.95537	0.53696
	Ridge	**0.01222**	0.00955	0.95540	0.53652
	LLCV	0.01223	0.00955	**0.95579**	0.53591
	RidgeCV	0.01282	0.00977	0.95190	0.65593
	HR	0.01368	0.01080	0.94860	0.68030
	ARD	0.01225	**0.00947**	0.95501	0.56922
GPR	**DP + WK**	0.01227	0.00963	0.95551	**0.53384**
SVM	LinearSVR	0.01582	0.01264	0.93631	0.63028

Note: PF: Polynomial Features, LLCV: Cross-validated Lasso, using the LARS algorithm, Ridge: Linear least square with L2 regularization, HR: Hubert Regressor, ARD: Bayesian ARD regression, RidgeCV: Ridge regression with built-in cross-validation.



(a)

(b)

Fig. 6: Crack development at different load positions: (a) initial position, and (b) 3 cm from the initial position.

placed respectively at the initial position (5 cm from the middle of the beam) and at 3 cm from the initial position (8 cm from the middle of the beam). The crack is developed from the initial notch one, and propagated to the force position. Notably, when the load is placed further from the middle of the beam, a more significant change in propagation direction can be observed due to the appearance of the shear force. Once the automatic simulation process is completed, 1000 obtained samples are approximated using available regression models in Python. The fitness level of the models are evaluated by the evaluation metrics described in Eqs. (7)-(9). Each model is ranked according to its model score, denoted as MS, calculated as the sum of evaluation metrics, normalized in the same $[0, 1]$ range, i.e.

$$MS = |RMSE| + |MAE| + |1 - R^2|. \tag{10}$$

Table II presents the average measures of the best model families and their variants out of 25 models available in Python 3.8.8, i.e. the Linear Model (LM), the Gausian Process Regression (GRP), and the Support Vector Machine (SVM). Here, a threshold of 0.1 and a threshold of 0.7 are applied respectively to the error metrics *RMSE*, *MAE*, and the coefficient of determination R^2 to disregard models that are characterized by only one or two metrics. It can be seen that the LM PF of degree 1 and LM Ridge return the best $\overline{RMSE}$, LM ARD returns the best $\overline{MAE}$, LM LLCV returns the best $\overline{R^2}$, while GPR with Dot Product (DP) and White Kernel (WK) is the best model in terms of the combination of three evaluation metrics $\overline{MS}$. In the next sections, the correlation between the resulted CRKMDL, the dimension of which is 25 models × 1000 simulated cracks, will be analyzed to verify the possibility of approximating real-world crack with the obtained models.

C. Correlation Between CRKMDL and Crack Generated from a Bending Experiment

In this experiment, a concrete beam is made and benched at the Laboratory of Structure, University of Transport and Communications, Hanoi, Vietnam. The dimension and size of the beam is given in Table I. The beam is casted in the Laboratory with a curing time of 28 days, when the concrete strength of 51 MPa has been achieved. The experiment environment consists of a compressor and a control room as shown in Fig. 7 (a) and (b). The setup of the experiment is presented in Fig. 7(c), similar to the simulation presented in Section IV-B. The load is applied to the beam until the beam is destroyed, where the load value of 898 kN is noted in the control room.

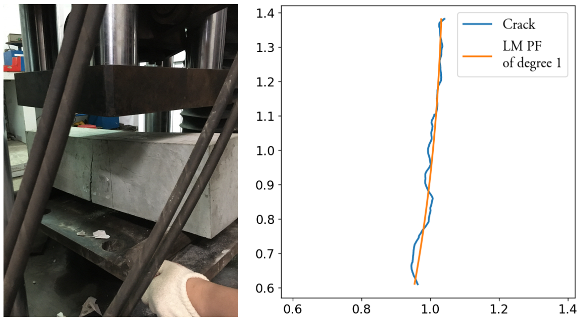
The result of the bending experiment is presented in Fig. 8. Once the beam has been destroyed, the image of the crack propagation is captured and the corresponding groundtruth is annotated as shown in Fig. 8 (a) and (b). The correlation between the annotated crack and the simulated ones are analyzed via calculating the measure of differences between the annotated crack and CRKMDL. Table III shows the performance of the best fit models for the annotated crack. It can be seen that SVM with LinearSVR returns the best R^2 while GPR DP+WK returns the least *MAE*. The best fit model in terms of the sum of the evaluation metrics is LM PF of degree 1 while the *RMSE* of LM RidgeCV is the least. The annotated crack and the fitted model is illustrated in Fig. 8(b).

D. Correlation between CRKMDL and Crack from other Image Datasets

The results reported in Table III have verified that the models generated from the simulations can well present a crack, the bending process of which is similar to the simulation. In

(a) (b) (c)

Fig. 7: Experimental environment:
(a) Compressor, (b) Control room, and (c) Experiment setup.



(a) (b)

Fig. 8: Results of the bending experiment:
(a) crack propagation path and
(b) corresponding groundtruth and best fit model.

TABLE III: TOP MODELS FOR THE CRACK GENERATED FROM THE BENDING EXPERIMENT

Model	Variant	*RMSE*↓	*MAE*↓	R^2↑	*MS*↓
LM	**PF of degree 1**	0.01041	0.00780	0.98069	**0.04085**
	Ridge	0.01041	0.00781	0.98070	0.04111
	LLCV	0.01041	0.00781	0.98070	0.04111
	RidgeCV	**0.01010**	0.00776	0.97225	0.06192
	HR	0.01063	0.00795	0.98518	0.05447
	ARD	0.01044	0.00777	0.98059	0.05463
GPR	DP + WK	0.01028	**0.00772**	0.98018	0.10564
SVM	LinearSVR	0.01070	0.00810	**0.98753**	0.06167

the next section, the correlation between those models and annotated crack from some reputable datasets is analyzed to evaluate whether the similarity is still maintained.

In this experiment, each annotated crack from the analyzed datasets is approximated by all models from the CRKMDL, where only results of the best fit model is taken into account. For each crack image dataset, the average measures are calculated as the mean of the fitting metrics obtained from the best fitting models for each image. The quantitative results of the experiment are reported in Table IV. Among six concerned crack image datasets, CRKMDL performs its best on cracks of Stone331, verified by the lowest $\overline{RMSE}$, $\overline{MAE}$, the highest $\overline{R^2}$, and the lowest $\overline{MS}$. Fig. 9 presents the best, median, and worst samples for each image dataset, sorted by the ranking metric expressed in Eq. 10. Notably, the crack shapes in CFD, CRKWH100, and CrackTree260 datasets are the most complicated, with numerous direction change points in the propagation path, leading to a difficulty in the approximation using CRKMDL. On the other hand, those in Crack500, Stone331, LS315 are more monotonous, resulting in better fitting results by models of CRKMDL.

Fig. 10 illustrates the distribution of the metric *MS* on six analyzed crack image datasets. Notably, the median of Stone331 is the lowest, following by that of LS315, Crack500, and the remaining datasets. The same order applied for the data dispersion, where Stone331, LS315, and Crack500 are respectively ranked at the first, the second, and the third position in terms of dense distribution. Meanwhile, CFD, CRKWH100, and CrackTree260 are more scattered, especially CrackTree260 with the widest range of *MS*, from 0 to 3. On the other hand, Crack500 and Stone331 are datasets with the most outliers, i.e. 11 for the former, and 7 for the latter.

E. An Application in Vision-based Crack Detection

To investigate the feasibility of the proposed approach in vision-based crack detection, a post-processing task is conducted on results of some recent effective detection and segmentation models on the GAPs384 [50] dataset. GAPs384 includes also artefacts such as black spots, road-line, material edge that lead to false positives returned by participated detection models. Pixels that are detected as crack by comparative models are classified as false positive if they are not aligned with the groundtruth of the GAPs384 dataset. Those resulting regions are labeled as crack and non-crack, taking into account some specific features such as the Histograms of Oriented Gradients (HOG) [51], the eccentricity, and the crack propagation similarity obtained from the simulated crack CRKMDL. The state-of-the-art deep learning-based crack detection and segmentation models selected for this task are as follows:

- DeepCrack [12]: The end-to-end trainable deep convolutional neural network for automatic crack detection predicts crack pixels based on the Deeply-Supervised Nets (DSN), Fully Convolutional Networks (FCN) and Deep Hierarchical Convolutional Neural Network (CNN).
- HED [13]: The Holistically Nested Edge detection is a learning-based end-to-end edge detection system that uses a trimmed VGG-like convolutional neural network for an image to image prediction task.

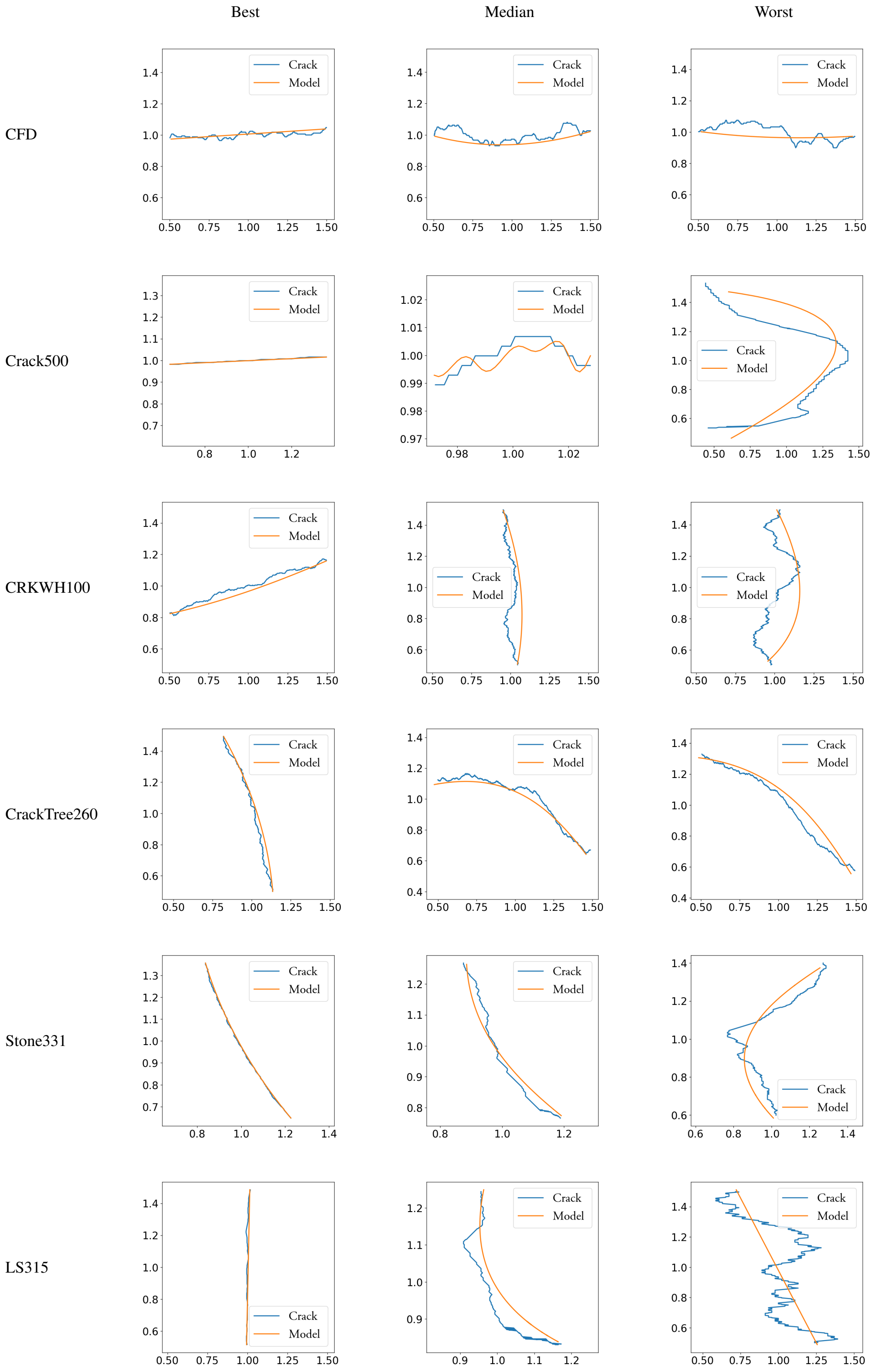


Fig. 9: Fitting results on crack image datasets.

- FCN [14]: The deep Fully Convolutional Network-based method proposed a hourglass shaped architecture as autoencoders for predicted per-pixel class labels to enable semantic segmentation.
- U-Net [15] The U-shaped encoder-decoder network architecture, which was an improved method of FCN. U-Net was initially applied in biomedical image processing then adopted to crack detection.

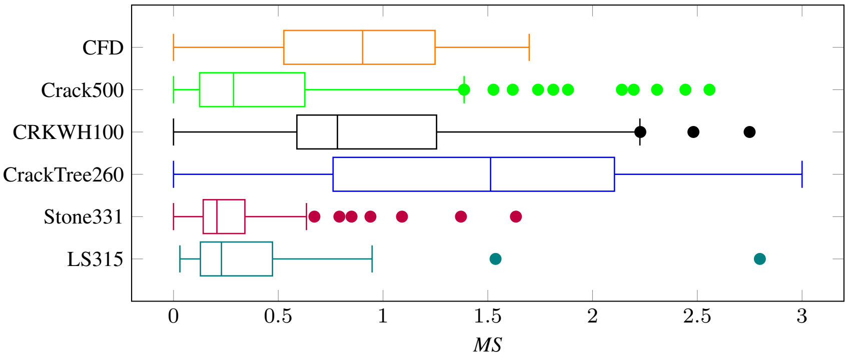


Fig. 10: Performance analysis on crack image datasets.

TABLE IV: RESULTS ON CRACK IMAGE DATASETS

Dataset	$\overline{RMSE}$↓	$\overline{MAE}$↓	$\overline{R^2}$↑	$\overline{MS}$↓
CFD	0.03021	0.02431	0.84739	0.92241
Crack500	0.03892	0.03001	0.81535	0.45577
CRKWH100	0.03175	0.02512	0.83853	0.97366
CrackTree260	0.01860	0.01445	0.92547	1.46761
Stone331	**0.01147**	**0.00914**	**0.95379**	**0.26963**
LS315	0.02209	0.01749	0.87534	0.39497

- RUC-Net [16] A ResNet- and U-Net-based semantic segmentation network for pavement crack image segmentation.
- DS-TransUNet [17]: A hierarchical swin transformer incorporated U-shaped architecture for image segmentation.
- U^{2}CrackNet [18]: A two-level nested U-structure network capturing multi-scale crack information.

The comparative models are trained and validated on an independent dataset, CFD, then tested on the GAPs384 dataset. Training and validation images are augmented from 118 original images of CFD using image rotation (0 to 90 degrees) and flipping (horizontal and vertical), cropped randomly to patches of size 256×256 and then split with a 9:1 ratio. The models are trained on NVIDIA Tesla T4 GPUs 16Gb. The Normal initialization [52] and Adaptive Moment Estimation [53] are employed for the generation of initial trainable parameters and optimization of the training, the learning rate of which is set at $1e-5$. The performance of the comparative models are evaluated using the F_β score where β^2 is selected as 0.3 as suggested in [54] for emphasizing the importance of precision in defect detection. The formula of F_β is given as

$$F_\beta = (1+\beta^2)\frac{Pr \times Rc}{\beta^2 \times Pr + Rc}, \tag{11}$$

where Pr and Rc are respectively the precision and recall of the detection.

Out of 509 images of GAPs384, 421 images consisting of single cracks are selected for the experiment. The performance of the participating models on 421 images is reported in Table V. It is significant to see that FCN provides the most accurate segmentations with balanced precision and recall scores. Hence, the results of FCN are selected as training and testing samples for the post-processing task, the ratio of which are 80:20, i.e. 336 images for training and 85 images for testing. In addition, the segmentation results of the remaining models on 85 test images are included in the test set. The corresponding extracted regions employed in the training and testing are 3238 and 7006, respectively.

TABLE V: PERFORMANCE OF PARTICIPATING MODELS ON 421 IMAGES OF SINGLE CRACK

Method	DeepCrack	FCN	HED	U-Net	RUC-Net	DS-TranUNet	U^{2}CrackNet
F_β	70.38%	70.34%	60.91%	62.64%	61.11%	67.03%	68.99%
Pr	75.32%	69.67%	64.12%	60.64%	56.18%	65.93%	68.24%
Rc	59.90%	74.52%	53.71%	71.43%	87.53%	72.28%	72.80%

In the post-processing task, each foreground region returned by a participating model is characterized by HOG (H), eccentricity (E) and similarity (S) to the simulated propagation model. The high-contrast assumption between the crack and background are generally used as a starting point in the development of crack detection algorithms [55], [56]. Hence, the first feature is selected to deal with the lighting variations and intensity transition between the crack and background pixels. The latter evaluates the stretching of the segmented crack region and can be calculated via the semi-major and the semi-minor of its fitted ellipse. Since crack is observed to have a very small width and a continuous line structure [21], E can be a good feature representing its thinness and linearity. Let N be the number of foreground regions returned by a crack detection model, the HOG H_i and the eccentricity E_i features representing each ith region are defined as

$$H_i = \left[\frac{\sum_{j=1}^{L}(h_j^{(i)} - \overline{h^{(i)}})}{L}\right], \tag{12}$$

$$E_i = \frac{\sqrt{a_i^2 - b_i^2}}{a_i}, \tag{13}$$

where $h^{(i)}$ is the extracted vector of HOG features of length L, a_i and b_i are respectively the length of the major and minor axis of the ellipse sharing the same second central moment with the interested foreground region. The similarity S_i of the ith foreground region and the simulated crack models is calculated via the *RMSE* of the skeletonized region and

TABLE VI: IMPACT OF THE POST-PROCESSING ON $F_\beta(\%)$ OF COMPARATIVE MODELS ON THE CLASSIFICATION TEST

Method	DeepCrack	FCN	HED	U-Net	RUC-Crack	Ds-TranUNet	U^{2}CrackNet
N/A	69.76	69.85	61.14	62.551	61.19	67.14	69.39
with E	69.42	70.33	60.80	67.29	60.61	67.62	70.50
with H	70.24	66.08	59.83	61.63	56.91	63.82	65.39
with S	69.88	70.14	60.83	67.11	61.23	67.26	69.55
with EH	69.03	69.25	61.04	61.72	60.73	66.26	69.42
with HS	**70.72**	**70.94**	**61.37**	**68.02**	**61.84**	67.59	69.43
with ES	69.44	68.97	59.77	65.90	60.05	66.24	**70.62**
with EHS	69.73	70.28	60.87	67.59	61.68	**67.69**	69.83

TABLE VII: IMPACT OF THE POST-PROCESSING ON $Pr(\%)$ OF COMPARATIVE MODELS ON THE CLASSIFICATION TEST

Method	DeepCrack	FCN	HED	U-Net	RUC-Crack	Ds-TranUNet	U^{2}CrackNet
N/A	74.38	69.26	64.58	60.61	56.35	66.07	68.91
with E	74.53	70.42	64.52	67.57	57.31	**68.65**	70.67
with H	75.53	67.27	63.43	61.48	54.59	65.13	66.01
with S	74.69	69.68	64.23	66.47	56.41	66.31	69.14
with EH	73.65	69.41	64.88	60.53	57.08	67.30	69.62
with HS	**76.28**	**71.03**	**65.51**	**68.37**	**57.95**	68.10	69.72
with ES	74.33	69.03	62.95	65.57	56.06	66.53	**70.98**
with EHS	74.89	70.29	64.61	67.35	57.37	67.76	69.68

TABLE VIII: IMPACT OF THE POST-PROCESSING ON $Rc(\%)$ OF COMPARATIVE MODELS ON THE CLASSIFICATION TEST

Method	DeepCrack	FCN	HED	U-Net	RUC-Crack	Ds-TranUNet	U^{2}CrackNet
N/A	59.97	73.79	53.61	71.06	86.64	72.35	72.37
with E	58.56	72.09	52.91	68.20	77.79	65.70	71.32
with H	58.81	64.59	52.43	64.10	68.83	61.56	64.67
with S	**59.69**	**73.51**	**53.44**	**70.82**	**86.57**	**71.88**	**72.30**
with EH	59.18	70.50	52.91	67.39	79.39	64.29	70.13
with HS	59.02	72.51	52.80	68.77	82.10	67.45	70.15
with ES	59.15	70.67	52.88	68.52	80.74	66.56	71.22
with EHS	58.79	72.08	52.93	69.88	84.17	69.11	71.79

the best regression model family reported in Table II, GPR DP+WK, as

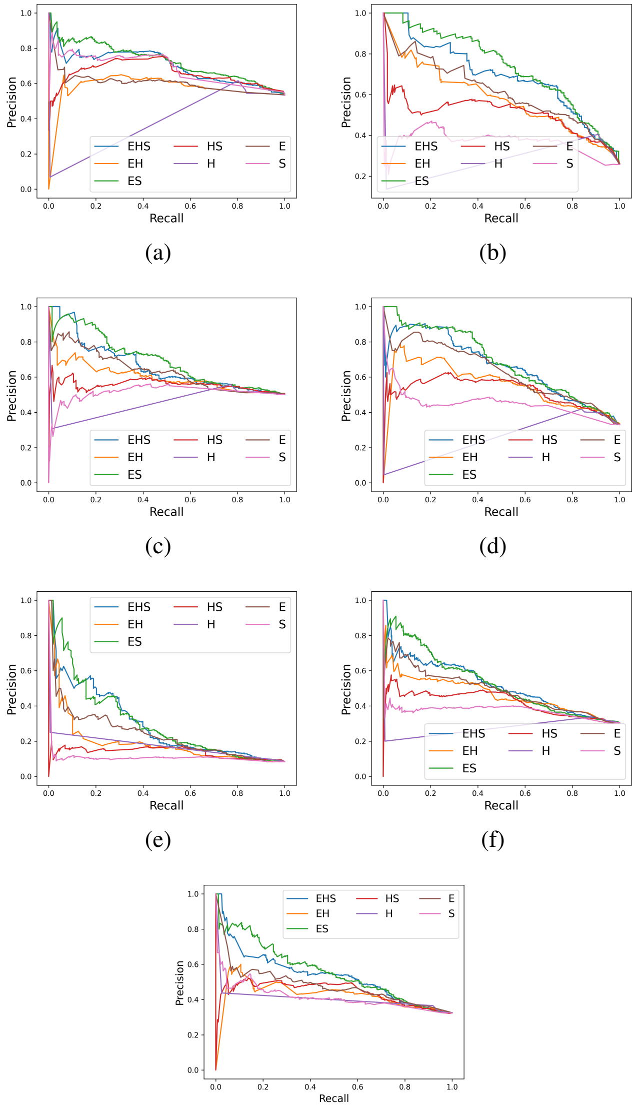
$$S_i = RMSE_i = \sqrt{\frac{(\sum_{k=1}^{A_i}(y_k - \hat{y}_k)^2}{A_i}}, \qquad (14)$$

where y_k and $\hat{y}_k$ are respectively the actual and predicted crack pixel, and A_i the number of pixels in the skeletonized ith region.

Out of 32 existing classifiers and their variants in Python 3.8.5, the Gradient Boosting Tree is employed for the classification task due to the best obtained accuracy from the 5-fold cross validation test. The optimized parameters of the model are identified via Grid Search, i.e. the learning rate, the maximum depth, and the estimator number are respectively 0.13, 3, and 11.

Let us denote the classifiers using two or three crack-shape features as EH, HS, ES, and EHS, where E, H, S, respectively, are the eccentricity, HOG, and similarity to the propagation model. Tables VI, VII, VIII report the performance of comparative models on selected images of the GAPs384 dataset without and with the involvement of the post-processing. For each participating model, the best results in F_β, Pr, and Rc when using a classifier is presented in bold. Notably, the proposed post-processing has increased the accuracy of the participating models where EHS and HS are classifiers with the most and the second most significant boost in terms of F_β (Table VI). The average increase in F_β when using HS and EHS are respectively 1.27% and 0.95% where the highest boost is achieved when using HS on segmentation results of U-Net, i.e. 5.47%. The performance of the single-feature classifiers is also reported. Out of 7 participating models, the improvement in F_β when using S is observed on 6 models, why the numbers for E and H are only 4 and 1, respectively. A further analysis in Table VII shows that using S or E can increase Pr on 6 out of 7 participating models where the number when using H is only 2. It is also noted that the gain with E is always higher than that with S. However, among 7 classifiers, S returns the best Rc in all participating models' segmentation results as reported in Table VIII. Since the post-processing task is performed on regions returned by participating models, it is expected that the precision increases and the recall decreases as a trade-off. Here, S is the only classifier delivering post-processing results with a minor deviation of Rc to that of the initial segmentations. As recall is the ratio between the number of detected and that of the actual crack pixels, the performance of S in Table VIII reemphasizes not only the robustness of the S feature but also its similarity to the crack shape from the test dataset.

The precision-recall curves of classifiers on participating



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Fig. 11: Precision-recall curves of classifiers on segmentation results of participating models: (a) DeepCrack, (b) FCN, (c) HED, (d) U-Net, (e) RUC-Net, (f) DS-TransUNet, (g) U^{2}CrackNet.

models are illustrated in Fig. 11. The curves of ES and EHS on segmentation results of all participating models are closer to the upper-right corner, verifying their exceptional performance against their counterparts. The reason for this performance is due to the significant improvement of ES, HS and EHS in terms of Pr (Table VII) while the decrease in terms of Rc is less, compared with EH and the remaining single-feature classifiers (Table VIII). The quantitative results have verified that the propagation model (S) is a strong feature that can enhance the accuracy of crack detection and segmentation techniques, especially when combining with the intensity transition feature HOG or the eccentricity E.

Fig. 12 demonstrates some false positives returned by participating models, such as pixels belong to a non-crack object (Images 2 and 3) or mis-identified pixels due to bad lighting conditions (Images 1-5), which can be efficiently removed using the EHS classifier. Notably, true positives of those models on the test images are preserved. This has confirmed the importance of the similarity to the simulated propagation model when handling impediments, which have yet to be considered in current crack detection approaches.

V. Parameter Analysis

In this section, the impact of the simulation parameters s and L presented in Table I on fitting results of six interested crack image datasets will be analyzed. Each time a parameter is adjusted, a simulated dataset is then generated, the fitting models of which are identified as described in Section IV-B. The correlation between these model sets and crack from six image datasets is then analyzed as per Section IV-D.

In the first analysis, s is adjusted while the force shifting range L is kept unchanged at 10 cm to verify the impact of the step size. Three extra simulated crack datasets are generated, the size of which is respectively 250, 500, 2000, corresponding to the step size at 0.04, 0.02, 0.005 cm. With this setting, simulated crack of the smaller dataset is a subset of that of the larger one. Fig. 13(a) illustrates the fitting error *MS* of those model sets on six crack image datasets. In general, a decreasing trend of *MS* on $[250, 1000]$ (Crack500, CRKWH100, CrackTree260, LS315) and an increasing trend after 1000 (CFD, Crack500, CRKWH100, Stone331) are observed on four out of six datasets. When analyzing the data directions on the whole range $[250, 2000]$, only two datasets (CFD and Stone331) showing an increasing trend. In fact, *MS* on CFD and Stone331 are optimal at 250 samples.

In the second analysis, the shifting range L of the force position is set at 2.5, 5, and 20 cm, generating three extra datasets of size 250, 500, and 2000 samples, respectively. The fitting error *MS* of the obtained models on six crack image datasets is illustrated in Fig. 13(b). In general, a decreasing trend is observed on four out of six datasets on the whole range [250, 2000]. Notably, *MS* of CrackTree260 seems to increase after 5 cm, while that of CRKWH100 and LS315 continues to decrease after 10 cm. On the other hand, *MS* on six participated datasets is optimal at $L = 10$ cm.

When adjusting a simulation parameter, the number of simulated cracks is adjusted accordingly. With more cracks generated, a better correlation of the obtained models to the real-world cracks is demonstrated. However, a too large number will lead to over-fitting and generated models no longer well approximate real-world cracks. This study verifies a simulated crack dataset of size 1000 to deliver the best-performed model family on all interested crack image datasets.

VI. Discussion

Experimental results have pointed out a high correlation between simulated and real-world cracks. As described in Section IV-C, models from the CRKMDL dataset can well predict the propagation path of the crack from the bending test. The propagation similarity in the simulation and the bending experiment can explain this performance. Regarding

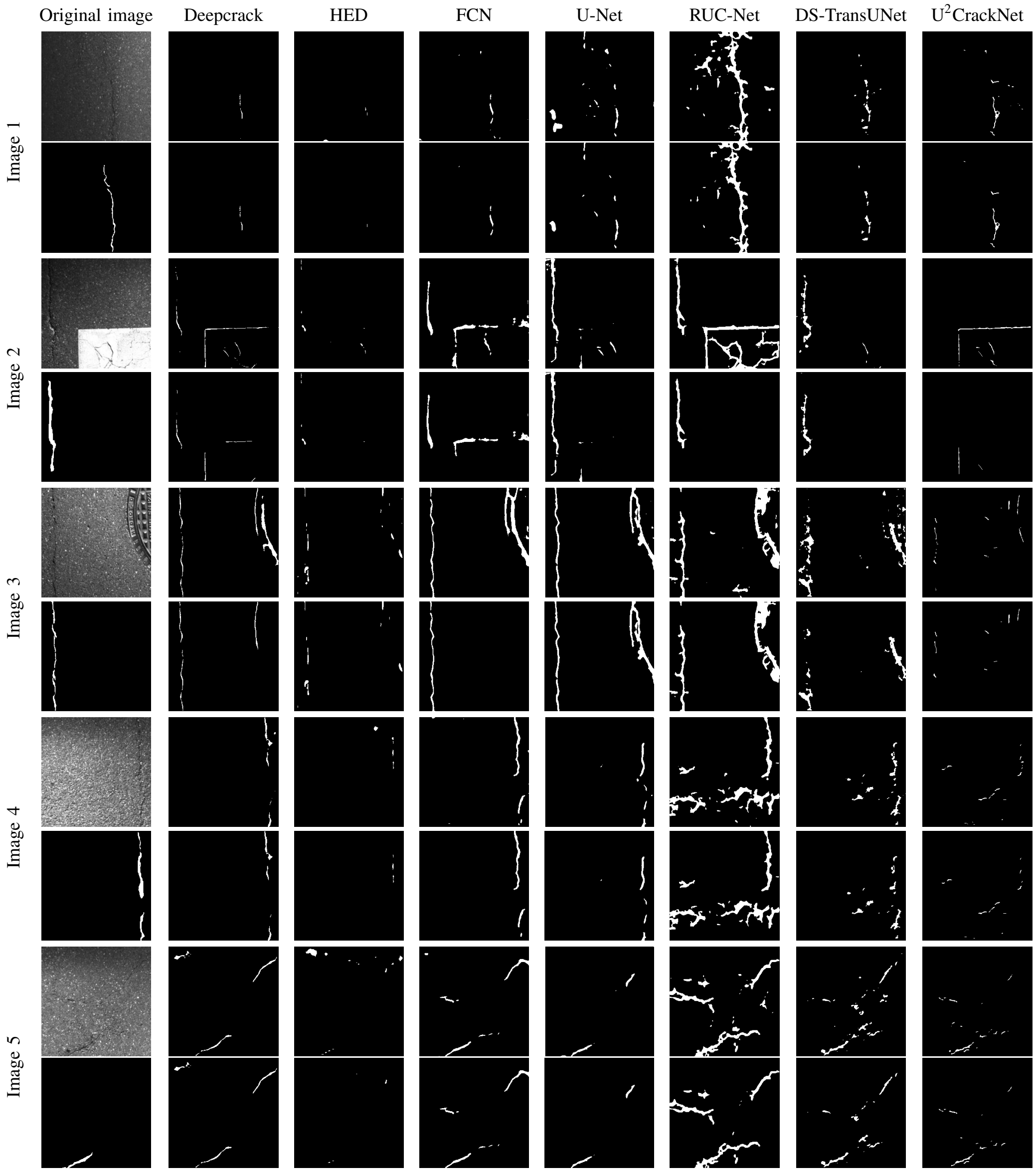
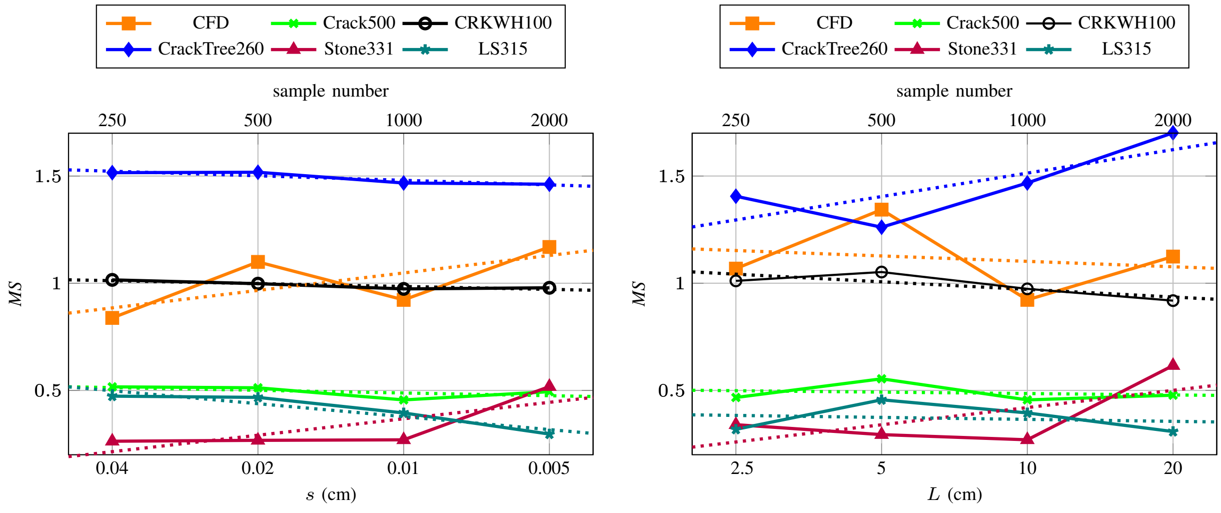


Fig. 12: Performance of EHS on participating models' results. For each image, the first row represents the original image and detection results of participating models, the second row represents the corresponding ground truth and post-processing results with EHS.

the performance of CRKMDL on crack image datasets, the fitting results are promising, especially on Crack500, Stone331, and LS315. Although an increase in fitting errors is observed when compared to the crack generated from the bending test, the results are positive. Considering the average result of CRKMDL on the analyzed datasets, the lowest $\overline{R^2}$ belong to Crack500 while the highest is from the Stone331, with 81.5% and 95.3% of the crack pixels predictable by the obtained regression models, respectively.

The result difference between the crack image from the bending test and that from other datasets can be interpreted as follows: First, due to high variability in the crack size and shape from the participated datasets, it is difficult to predict the propagation path using the current version of CRKMDL. Besides, the resolution difference between simulated and images of real-world cracks could impact the fitting results. While the number of crack tips generated from the simulation is limited in this study due to the size of the analyzed beam, that from six image datasets can be thousands in some cases. Simulating cracks on different types and sizes of beams can resolve these above matters, which will be studied in our future work.

Eventually, promising results have been obtained when employing the obtained propagation models for vision-based crack detection. Indeed, classifiers using the found feature have reported significant improvements on F_β of state-of-the-art crack detection and segmentation models to up to 5.47%. The great potential of the proposed work to enhance the accuracy of vision-based crack detection has been verified, where shape features obtained from the propagation models can be utilized to develop crack detection algorithms or build generative AI-



(a)

(b)

Fig. 13: Effect of simulation parameters on fitting errors of six crack image datasets: (a) s, (b) L.

based crack datasets.

VII. Conclusion

In this paper, an automatic approach simulating the propagation path of concrete cracks has been presented. A specific beam model has been employed as a case study for the generation of simulation samples and corresponding regression models. Experiment results on cracks generated from a bending test and that obtained from some reputable image datasets have shown a high correlation between the simulated models and the real-world crack. The method can be extended to create hundreds of thousands of simulation samples, taking into account the variability in terms of the type and size of concrete beams. While current DL-based approaches are limited by the size and the quality of the training dataset [19], the results of this study are expected to contribute to the research domain as follows: (i) providing an extra source of information in terms of crack shape based on the obtained regression models, and (ii) generating a reliable training data using augmentation and/or generative adversarial networks. Besides, current state-of-the-art methods are based on the similarity between crack and edge detection [10]–[12] in terms of shape and structure. We hope that the obtained approximation models from our work shed some light on the shape information of the object to be detected. On another note, real-world cracks are not limited to a single type, as discussed in this study. In our forthcoming research, cracks with complicated structures, such as interconnected ones, will be addressed.

References

[1] R. Ali, J. H. Chuah, M. S. A. Talip, N. Mokhtar, and M. A. Shoaib, “Structural crack detection using deep convolutional neural networks,” *Autom. Constr.*, vol. 133, p. 103989, 2022.

[2] D. Ai, G. Jiang, S.-K. Lam, P. He, and C. Li, “Computer vision framework for crack detection of civil infrastructure—a review,” *Eng. Appl. Artif. Intell.*, vol. 117, p. 105478, 2023.

[3] H. Li, D. Song, Y. Liu, and B. Li, “Automatic pavement crack detection by multi-scale image fusion,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2025–2036, 2018.

[4] T. H. Dinh, M. D. Phung, and Q. P. Ha, “Summit Navigator: A novel approach for local maxima extraction,” *IEEE Trans. Image Process.*, vol. 29, pp. 551–564, 2019.

[5] J. Gan, J. Wang, H. Yu, Q. Li, and Z. Shi, “Online rail surface inspection utilizing spatial consistency and continuity,” *IEEE Trans. Syst. Man Cybern. -Syst.*, vol. 50, no. 7, pp. 2741–2751, 2020.

[6] D. Ai, G. Jiang, L. S. Kei, and C. Li, “Automatic pixel-level pavement crack detection using information of multi-scale neighborhoods,” *IEEE Access*, vol. 6, pp. 24 452–24 463, 2018.

[7] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, “Automatic road crack detection using random structured forests,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3434–3445, 2016.

[8] L. Zhang, G. Zhou, Y. Han, H. Lin, and Y. Wu, “Application of internet of things technology and convolutional neural network model in bridge crack detection,” *IEEE Access*, vol. 6, pp. 39 442–39 451, 2018.

[9] H. Chen and H. Lin, “An effective hybrid atrous convolutional network for pixel-level crack detection,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.

[10] E. Gu, G. Xiao, F. Lian, T. Mu, J. Hong, and J. Liu, “Segmentation and evaluation of crack image from aircraft fuel tank via atrous spatial pyramid fusion and hybrid attention network,” *IEEE Trans. Instrum. Meas.*, 2023.

[11] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, “Feature pyramid and hierarchical boosting network for pavement crack detection,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1525–1535, 2019.

[12] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, “Deepcrack: Learning hierarchical convolutional features for crack detection,” *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1498–1512, 2018.

[13] S. Xie and Z. Tu, “Holistically-nested edge detection,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1395–1403.

[14] C. V. Dung *et al.*, “Autonomous concrete crack detection using deep fully convolutional neural network,” *Autom. Constr.*, vol. 99, pp. 52–58, 2019.

[15] Z. Liu, Y. Cao, Y. Wang, and W. Wang, “Computer vision-based concrete crack detection using u-net fully convolutional networks,” *Autom. Constr.*, vol. 104, pp. 129–139, 2019.

[16] G. Yu, J. Dong, Y. Wang, and X. Zhou, “RUC-Net: A residual-unet-based convolutional neural network for pixel-level pavement crack segmentation,” *Sensors*, vol. 23, no. 1, 2023.

[17] A. Lin, B. Chen, J. Xu, Z. Zhang, G. Lu, and D. Zhang, “DS-TransUNet: Dual swin transformer u-net for medical image segmentation,” *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–15, 2022.

[18] P. Shi, F. Zhu, Y. Xin, and S. Shao, “U2cracknet: a deeper architecture

with two-level nested u-structure for pavement crack detection," *Struct. Health Monit.*, p. 14759217221140976, 2022.

[19] S. K. Baduge, S. Thilakarathna, J. S. Perera, M. Arashpour, P. Sharafi, B. Teodosio, A. Shringi, and P. Mendis, "Artificial intelligence and smart vision for building and construction 4.0: Machine and deep learning methods and applications," *Autom. Constr.*, vol. 141, p. 104440, 2022.

[20] Y. Zhang, J. Zhong, Z. Liu, and Z. Han, "ECF-STPM: A robust crack detection method for railway catenary components," *IEEE Trans. Instrum. Meas.*, pp. 1–1, 2023.

[21] J. Liao, Y. Yue, D. Zhang, W. Tu, R. Cao, Q. Zou, and Q. Li, "Automatic tunnel crack inspection using an efficient mobile imaging module and a lightweight cnn," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15 190–15 203, 2022.

[22] T. L. Anderson, *Fracture mechanics: fundamentals and applications*. CRC press, 2017.

[23] T. H. Dinh, Q. P. Ha, and H. M. La, "Computer vision-based method for concrete crack detection," in *Proc. Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, 2016, pp. 1–6.

[24] Q. Chen, Y. Huang, H. Sun, and W. Huang, "Pavement crack detection using hessian structure propagation," *Adv. Eng. Inform.*, vol. 49, p. 101303, 2021.

[25] X. Weng, Y. Huang, and W. Wang, "Segment-based pavement crack quantification," *Autom. Constr.*, vol. 105, p. 102819, 2019.

[26] H. Deng, B. Yan, and T. Okabe, "Fatigue crack propagation simulation method using xfem with variable-node element," *Eng. Fract. Mech.*, vol. 269, p. 108533, 2022.

[27] K. C. Nwanoro, H. Lu, C. Yin, and C. Bailey, "Advantages of the extended finite element method for the analysis of crack propagation in power modules," *Power Electron. Devices Compon.*, vol. 4, p. 100027, 2023.

[28] L. R. S. Pereira and S. S. Penna, "Nonlinear analysis method of concrete structures under cyclic loading based on the generalized secant modulus," *Rev. IBRACON Estrut. Mater.*, vol. 15, 2022.

[29] D. Dias-da Costa, V. Cervenka, and R. Graça-e Costa, "Model uncertainty in discrete and smeared crack prediction in rc beams under flexural loads," *Eng. Fract. Mech.*, vol. 199, pp. 532–543, 2018.

[30] P. M. Phuc, D. T. Manh, and N. D. Duc, "Free vibration of cracked fgm plates with variable thickness resting on elastic foundations," *Thin-Walled Struct.*, vol. 161, p. 107425, 2021.

[31] J. Gálvez, J. Červenka, D. Cendón, and V. Saouma, "A discrete crack approach to normal/shear cracking of concrete," *Cem. Concr. Res.*, vol. 32, no. 10, pp. 1567–1585, 2002.

[32] A. R. Ingraffea and V. Saouma, "Numerical modeling of discrete crack propagation in reinforced and plain concrete," in *Fracture Mechanics of Concrete: Structural Application and Numerical Calculation*. Springer, 1985, pp. 171–225.

[33] K. A. Nair and S. Ghosh, "Crack tip enhanced phase-field model for crack evolution in crystalline ti6al from concurrent crystal plasticity fe-molecular dynamics simulations," *Eur. J. Mech. A-Solids*, vol. 100, p. 104983, 2023.

[34] E. Azinpour, S. Rzepa, D. Melzer, A. Reis, J. Džugan, and J. C. de Sa, "Phase-field ductile fracture analysis of multi-materials and functionally graded composites through numerical and experimental methods," *Theor. Appl. Fract. Mech.*, p. 103906, 2023.

[35] T. Van Do, D. Hong Doan, N. Chi Tho, and N. Dinh Duc, "Thermal buckling analysis of cracked functionally graded plates," *Int. J. Struct. Stab. Dyn.*, vol. 22, no. 08, p. 2250089, 2022.

[36] A. P. Chaves, R. G. Peixoto, and R. P. da Silva, "Analysis of 3d quasi-brittle solids failures by crack growth using the strong discontinuity approach with the boundary element method," *Int. J. Solids Struct.*, p. 112269, 2023.

[37] M. Jirásek and T. Zimmermann, "Embedded crack model. part ii: Combination with smeared cracks," *Int. J. Numer. Methods Eng.*, vol. 50, no. 6, pp. 1291–1305, 2001.

[38] Y. Theiner and G. Hofstetter, "Numerical prediction of crack propagation and crack widths in concrete structures," *Eng. Struct.*, vol. 31, no. 8, pp. 1832–1840, 2009.

[39] G.-H. Shi, "Manifold method of material analysis," Army Research Office Research Triangle Park NC, Tech. Rep., 1992.

[40] W. Wei, Q. Zhao, Q. Jiang, and G. Grasselli, "A new contact formulation for large frictional sliding and its implement in the explicit numerical manifold method," *Rock Mech. Rock Eng.*, vol. 53, no. 1, pp. 435–451, 2020.

[41] G. Li, K. Wang, X. Qian *et al.*, "An nmm-based fluid-solid coupling model for simulating rock hydraulic fracturing process," *Eng. Fract. Mech.*, vol. 235, p. 107193, 2020.

[42] X. Zhou, L. Fan, and Z. Wu, "Effects of microfracture on wave propagation through rock mass," *Int. J. Geomech.*, vol. 17, no. 9, p. 04017072, 2017.

[43] L. Fan, X. Yi, and G. Ma, "Numerical manifold method (nmm) simulation of stress wave propagation through fractured rock mass," *Int. J. Appl. Mech.*, vol. 5, no. 02, p. 1350022, 2013.

[44] G. Papazafeiropoulos, M. Muñiz-Calvente, and E. Martínez-Pañeda, "Abaqus2matlab: A suitable tool for finite element post-processing," *Adv. Eng. Softw.*, vol. 105, pp. 9–16, 2017.

[45] B. L. Karihaloo, "Fracture mechanics & structural concrete," *Longman Scientific and Technical*, 1995.

[46] Z. P. Bažant, Q. Yu, and G. Zi, "Choice of standard fracture test for concrete and its statistical evaluation," *Int. J. Fract.*, vol. 118, no. 4, pp. 303–337, 2002.

[47] S. P. Shah, "Size-effect method for determining fracture energy and process zone size of concrete," *Mater. Struct.*, vol. 23, no. 6, pp. 461–465, 1990.

[48] S. Groß and A. Reusken, "An extended pressure finite element space for two-phase incompressible flows with surface tension," *J. Comput. Phys.*, vol. 224, no. 1, pp. 40–58, 2007.

[49] T. Belytschko, N. Moës, S. Usui, and C. Parimi, "Arbitrary discontinuities in finite elements," *Int. J. Numer. Methods Eng.*, vol. 50, no. 4, pp. 993–1013, 2001.

[50] M. Eisenbach, R. Stricker, D. Seichter, K. Amende, K. Debes, M. Sesselmann, D. Ebersbach, U. Stoeckert, and H.-M. Gross, "How to get pavement distress detection ready for deep learning? a systematic approach," in *Proc. Int. Jt. Conf. Neural Netw. (IJCNN)*. IEEE, 2017, pp. 2039–2047.

[51] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, 2005, pp. 886–893 vol. 1.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1026–1034.

[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–8.

[54] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 815–828, 2019.

[55] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "Cracktree: Automatic crack detection from pavement images," *Pattern Recognit. Lett.*, vol. 33, no. 3, pp. 227–238, 2012.

[56] Q. Li, Q. Zou, D. Zhang, and Q. Mao, "Fosa: F* seed-growing approach for crack-line detection from pavement images," *Image Vision Comput.*, vol. 29, no. 12, pp. 861–872, 2011.