Synergizing Low Rank Representation and Deep Learning for Automatic Pavement Crack Detection

Zhi Gao, Xuhui Zhao, *Graduate Student Member, IEEE*, Min Cao, Ziyao Li, Kangcheng Liu, *Member, IEEE*, and Ben M. Chen, *Fellow, IEEE*

***Abstract*— Due to the critical role of pavement crack detection for road maintenance and eventually ensuring safety, remarkable efforts have been devoted to this research area, and such a trend is further intensified for the coming unmanned vehicle era. However, such crack detection task still remains unexpectedly challenging in practice since the appearance of both cracks and the background are diverse and complex in real scenarios. In this work, we propose an automatic pavement crack detection method via synergizing low rank representation (LRR) and deep learning techniques. First, leveraging LRR which facilitates anomaly detection without making any specific assumption, we can easily discriminate most of the frames with cracks from the long sequence with a consistent pavement base, followed by a straightforward algorithm to localize the cracks. In order to achieve the intelligence of detecting cracks with different pavement basis under unconstrained imaging conditions, we resort to deep learning techniques and propose a deep convolutional neural network for crack detection leveraging on multi-level features and atrous spatial pyramid pooling (ASPP). We train this network based on the training data obtained in the previous stage in an end-to-end manner. Extensive experiments on a wide range of pavements demonstrate the high performance in terms of both accuracy and automaticity. Moreover, the dataset generated by us is much more extensive and challenging than public ones. We put it online at https://gaozhinuswhu.com to benefit the community.**

***Index Terms*— Pavement crack detection, low rank representation, deep learning.**

I. Introduction

ROADS play an extremely important role in human activities and the governments have made tremendous efforts to build high-quality road networks. Consequently, as an essential operation to evaluate the pavement condition for maintenance purpose and eventually ensuring safety, pavement crack detection has been attracting increasingly remarkable efforts. In particular, such research has been fueled by the booming unmanned vehicle techniques from two aspects: first, the demand has been further heightened since the unmanned vehicles always expect the road in good condition; second, the prospect has been changed as the unmanned platforms can eventually help to realize full automation of the tedious crack detection tasks without any human intervention. However, such crack detection task still remains unexpectedly challenging in practice since the appearance of both cracks and the background are diverse and complex in real scenarios, as shown in Fig. 1. Therefore, effective and robust crack detection with less manual operation has been attracting increased attention from both academic and industrial communities.

Fig. 1. The diverse appearances of roads with and without cracks. Row 1: Different types of cracks. Row 2: Various road backgrounds. Both cracks and background vary in appearance significantly, and may with much noise, dark shadow, and inconsistent brightness, etc.

Generally, the available pavement crack detection methods can be classified into two categories: 3D data (such as depth map and point cloud) based and 2D image based (henceforth 3D-based and 2D-based respectively). The 3D-based methods essentially perform crack detection relying on the 3D information which can be obtained using various sensors, such as LiDAR sensors [1], [2], [3], structured light emitters [4], [5], [6], holographic detectors [7], and stereo cameras [8], [9], [10]. In contrast, 2D-based methods mainly depend on visual information (pixel intensity or features) that are captured with linear or array cameras [11], [12]. Intuitively, the 3D-based methods are more robust than the 2D-based ones, attributing to the 3D information which is generally more stable than 2D images, but at the expense of much higher cost and complexity of the sensor setup. In Table I, we briefly summarize the pros and cons of both 3D-based and 2D-based methods.

Manuscript received 19 May 2022; revised 31 December 2022 and 26 March 2023; accepted 8 May 2023. Date of publication 24 May 2023; date of current version 4 October 2023. This work was supported in part by the National Natural Science Foundation of China Major Program under Grant 42192580 and Grant 42192583, in part by the Hubei Province Natural Science Foundation under Grant 2021CFA088 and Grant 2020CFA003, and in part by the Science and Technology Major Project under Grant 2021AAA010 and Grant 2021AAA010-3. The Associate Editor for this article was M. Guo. *(Corresponding author: Zhi Gao.)*

Zhi Gao and Xuhui Zhao are with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China, and also with the Hubei Luojia Laboratory, Wuhan 430079, China (e-mail: gaozhinus@gmail.com; zhaoxuhui@whu.edu.cn).

Min Cao is with Wuhan Guanggu Zoyon Science and Technology Company Ltd., Wuhan 430223, China (e-mail: market@zoyon.com.cn).

Ziyao Li is with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China (e-mail: liziyao@whu.edu.cn).

Kangcheng Liu is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: kangcheng.liu@ntu.edu.sg).

Ben M. Chen is with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: bmchen@cuhk.edu.hk).

Digital Object Identifier 10.1109/TITS.2023.3275570

TABLE I
DATA AND SENSING TECHNIQUES FOR CRACK DETECTION

Source	Sensing Tech.	Pros	Cons
2D image	Linear/array imaging	Dense sensing with rich visual information	Sensitive to external interference objects
3D data	LiDAR	High resolution with satisfying efficiency	High cost with complex sensor setup
	Structured light	Simple sensor setup with high sensing accuracy	Sample with single line and complex calibration
	Laser holography	Imaging with high sensitivity and response	Sophisticated sensor setup with high cost
	Stereovision	Easy to setup with off-the-shelf sensors	Sensitive to unstable illumination and features

In this work, intending to press maximal benefit from the visual information, we focus on 2D image data and propose an automatic pavement crack detection method via synergizing low rank representation (LRR) and deep learning techniques. At first glance, such a crack detection task seems to be simple and can be accomplished using the extensions of popular machine vision algorithms. However, it is unexpectedly challenging in practice due to the diversity and complexity of both cracks and the background in terms of appearance. Fig. 1 shows some typical road surfaces with and without cracks, and we further summarize general difficulties below:

- Both cracks and pavements vary significantly in appearance, leading to challenges in a unified description.
- Cracks share similarities with pavements in color and texture, rendering it difficult to be distinguished.
- The presence of various interference factors, such as ruts, stains, and shadows makes the problem harder.

In past decades, researchers have extensively exploited visual features to accomplish automatic pavement crack detection. Early efforts usually focused on low-level intensity relevant features [13], [14], followed by thresholding algorithms ranging from the classical one [15] to more complex ones [16]. With the development of image processing techniques, more sophisticated methods which investigated the higher level information or structural constraint had been proposed for such tasks, including morphological operations [17], hand-crafted filters [18], [19], local features [12], [20], domain transformations [21], [22], texture and saliency analysis [23], [24], [25]. Moreover, some work formulated the crack detection problem from different theoretical frameworks and provided effective solutions, such as minimal path optimization [26], [27], [28], [29], [30], geometric analysis [31], [32], [33], contour tracking [34], classification [35], [36], [37], [38], [39], and superpixel segmentation [40], [41]. Despite claimed competitive performances on specific datasets, these methods, however, usually achieve unsatisfying results even fail in complex situations due to their overly restrictive assumptions about the crack or background, or both.

In recent years, encouraged by the stunning performance in many vision tasks, deep convolutional neural networks (DCNN, generally also termed as deep learning) have been applied for such crack detection task and reported fairly promising results on the testing datasets. The task is generally tackled from different views with various backbones, such as the classification [42], [43], [44], [45], [46], the object detection [47], [48], [49], [50], the segmentation [51], [52], [53], [54], [55]. However, the data-driven nature of deep learning technologies hindered these DCNN-based methods from more popular applications in practice for two major reasons: first, it is difficult, tedious, and expensive to prepare enough high-quality samples (both positive and negative) for training to obtain expected performance, despite some efforts with weakly-supervised learning [56]; second, as the generalization is always a weakness and concern of DCNN-based methods, existing methods cannot perform consistently well in practical scenarios which are diverse, complex, and may have never been seen in the training stage.

To overcome the aforementioned challenges and limitations of both traditional methods and DCNN-based methods, we propose an automatic pavement crack sample generation and detection method that works effectively under a wide variety of scenes encountered in practice via synergizing low rank representation and deep learning techniques. LRR, well known for self-expressing, can effectively detect anomaly from a batch of data with few assumptions of the foreground or background no matter how complex the pavement looks. We can obtain both high-quality positive samples (pixel-level labelled cracks with complex background) and negative samples (complex background without crack) with little requirement of parameters tuning. With enough training samples and orchestrating organization, we formulate pavement crack detection as semantic segmentation with DCNN. In particular, we follow the encoder-decoder structure with dual feature extractors (focusing on low-level feature and high-level semantics) for robust description and atrous spatial pyramid pooling (ASPP) for a wider receptive field. Extensive experiments on a wide range of pavements demonstrate that our method outperforms many state-of-the-art approaches in terms of both accuracy and automaticity. Moreover, the generated dataset is more extensive and challenging than the public ones, we make it available at https://gaozhinuswhu.com to benefit the community.

The remainder of this paper is organized as follows. We first introduce the related works in Section II. Section III is devoted to our pavement crack detection method via synergizing LRR and deep learning techniques, and Section IV presents our experiments and analysis. Finally, we conclude our work and discuss valuable future directions in Section V.

II. RELATED WORKS

We here discuss 3D-based and 2D-based pavement crack detection works according to the aforementioned taxonomy, where the latter will be elaborated in detail.

A. 3D-Based Methods

LiDAR and structured light techniques are popularly adopted to obtain the 3D information of the pavement. 3D-based methods generally exploit the difference between cracks and pavements in the depth direction for crack detection. In some novel applications, the LiDAR sensor was

mounted on mobile robots [3] and UAVs [57] for road and bridge damage detection, respectively. Generally, 3D-based methods can be divided into traditional methods and learning methods. Traditional methods mainly leverage geometric elevation and reflection intensity for crack detection with various techniques, including spatial filtering [1], Gaussian filtering [58] and iterative voting [59]. But these methods usually require much computation and achieve poor accuracy in some scenarios with a low signal-to-noise ratio (SNR). While learning methods most adopt dimensionality reduction, converting 3D point clouds to 2D images to reduce the processing difficulty, such as CrackNet [60] and its variant [61]. This strategy discards elevation information and reduces the resolution of point clouds by projecting multiple points into one pixel. Typically, the performance heavily relies on well-annotated training data, which is time-consuming, high-cost, and labor-intensive. Therefore, some researchers try to detect cracks with graph convolutional networks (GCN) and train the network in a semi-supervised manner [62]. However, the graph network heavily relies on the pattern of target data. For the best expression of features, a new graph structure needs to be redesigned for different cracks, resulting in declined transfer ability and cannot be applied to various scenarios with simple re-training. Moreover, 3D-based methods cannot detect cracks that are subtle in 3D point clouds but salient in visual images due to the limited resolution of discrete sampling.

As a summary, compared with vision cameras, the laser scanners or radar, not only significantly increase the cost of the system, also render it too complex to be operated and maintained by non-professionals. Compared with 2D optical images, 3D data is robust to some external interference, such as illumination variation, resulting in a reduced difficulty for crack detection in some scenes. But 3D data generally lack visual texture that plays an important role in crack detection. Therefore, we focus on 2D approaches in this paper.

B. 2D-Based Methods

For the rich information it contains, 2D optical image has been extensively exploited for pavement crack detection. The available 2D-based methods essentially exploited the intensity and a wide range of techniques and their combinations have been investigated for robust performance in complex scenes, bringing difficulties in methodology taxonomy. We here briefly breakdown these methods into the following categories according to the main features and techniques adopted.

1) Pixel Intensity-Based Methods: Maybe, the simplest strategy of crack detection is to perform thresholding on the intensity, and researchers have proposed a variety of methods for adaptive threshold estimation [13], [15], [36], [63]. Also intuitively, the grayscale gradient has been exploited for crack detection via edge detection [37] and contour tracking [34]. To further exploit the local information among neighbouring pixels, a variety of methods based on morphological operators [17], filters [18], [19], and local binary patterns (LBPs) [12], [20] have been proposed for crack detection. Despite numerous adopted methods, it is still hard to detect cracks accurately with only intensity information in our complex world due to intensity-similar interference, such as dark ruts.

2) Feature-Based Methods: In addition, to further exploit the visual information in a larger scale or a higher level, methods based on texture and saliency analysis have been proposed. In [23] and [24], a unified framework was proposed including estimating textural patterns of underlying backgrounds, obtaining the local variation, and isolating crack pixels. In [25] and [64], the saliency estimation and local statistical characteristics analysis were cooperated for crack detection. Besides, based on the results of key point detection, some researchers formulate crack detection as an optimization problem of minimal path selection in a graph. In particular, different metrics such as predefined length [26], depth [30] and thickness [27] have been proposed within such frameworks. Some works are further developed with geometric topological analysis of cracks [31], [32], [33]. Also naturally, the hand-crafted feature descriptors are fed into various classifiers for crack detection [38], [39], following traditional machine learning frameworks. Meanwhile, some methods divide images into irregular non-overlapping regions with superpixel segmentation [40], [41].

Moreover, based on the basic observation that the high frequency components of the image usually correspond to the cracks, methods based on frequency analysis have been proposed. Typically, wavelet transform has been well studied to decompose the original image into different frequency bands, followed by noise suppression and morphological operations on the high frequency content for crack detection [21], [22]. With the various appearance of cracks in different scenes, it is hard to find a general description and distinguish all types of cracks using limited data. Therefore, many methods only work in target scenarios with satisfying accuracy.

3) Deep Learning Methods: Similar to the trends in various vision tasks, many DCNN-based methods have been proposed for crack detection. In [42], [43], and [45], the crack detection was formulated as a classification task, therein sophisticated techniques including multi-scale feature extraction [46] and naive Bayes fusion scheme [44] have been investigated to improve the resulting performance. Despite various techniques adopted, the core idea is to express cracks with distinguishable features and then classify images or patches. Generally, it works in some simple scenarios, but cannot achieve good performance in complex scenes with confusing objects. Also, some methods detect cracks with patch-level rather than pixel-level, leading to coarse results. From the view of semantic segmentation, a hierarchical segmentation network [52] has been proposed based on SegNet [51]. Compared with classification, these methods exploit more latent and semantic information in the image and usually achieves better performance (pixel-level mask). However, with more complex network structures, more labelled samples are required for training, which is a predicament in crack detection due to limited public datasets and various scenarios. Despite some works are also proposed to reduce the dependence on training data with weakly-supervised [56], semi-supervised [65], even unsupervised strategy [66], they can be regarded as compromise to this dilemma when no enough data available, since

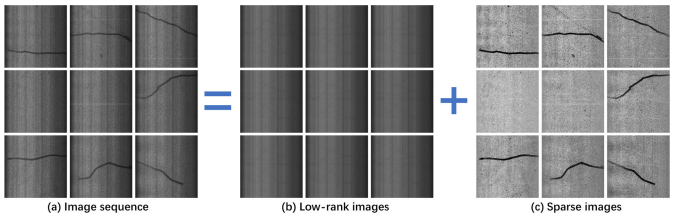

Fig. 2. LRR decomposes the (a) pavement image sequence into (b) low rank images and (c) sparse images. Cracks are highlighted in (c) sparse images and can be easily processed for pixel-wise masks.

fully-supervised learning is usually better than aforementioned strategies. Furthermore, some methods [47], [48], [49], [50] detect cracks with bounding box results from the perspective of object detection, following the famous one-stage object detection network YoLo series work. Cracks are regarded as objects and detected with some classical object detection networks or their variants. Compared with the aforementioned methods, these methods usually lead to the accuracy decline with a bounding box rather than pixel-wise mask. Typically, all these data-driven methods rely on large training samples for robust performance. However, few public datasets cover a wide range of scenes, leading to a limited generalization. Besides the pavement crack detection, other networks are proposed for structural health and damage monitoring [53], [67], [68], [69], [70], where STRNet [55] especially achieves state-of-the-art performance in that field.

From traditional methods to deep learning methods, many researchers have made great efforts and achieved some compelling results. However, due to the aforementioned natures of cracks and backgrounds, it is still a challenge to achieve robust and accurate crack detection. Many existing methods are only suitable for special scenarios or limited applications. We think the root of this problem lies in strong priori and hypotheses about data and inadequate exploitation of valuable latent information in an image. This problem can be potentially solved by low rank method [71], which decomposes a matrix into low rank and sparse parts with a theoretical convergence guarantee. Therefore, we introduce LRR to crack detection for the self-representation of image data. Then, we design a DCNN from the view of semantic segmentation and train this network with the generated dataset from LRR. To our best knowledge, there is no similar work at present.

III. Our Algorithm

We propose a novel method for crack detection leveraging LRR and DCNN, the whole scheme is shown in Fig. 3. This section is dedicated to the details of our algorithm, including the subsections of the LRR formulation for crack detection, post-processing and sample preparation, DCNN for crack detection, and implementation details.

A. LRR Formulation for Crack Detection

Rank is one of the most fundamental characteristics of a matrix and can be leveraged for exploiting essential information. Many researchers achieve amazing results in different fields, such as moving object detection [72] and traffic analysis [73]. In the inspection of pavement, we usually get many image sequences collected by cameras mounted on a vehicle, which contain rich tempo-spatial information. As *Aristotle* says in *Metaphysics*: The whole is greater than the sum of the parts. We take the whole sequence as input and explore the latent information hidden in the sequence, which is of great value for crack detection. Images with cracks usually account for part of the sequence, meanwhile, cracks generally occupy a small area in the image. It naturally occurs to us to introduce LRR into crack detection as shown in Fig. 2. We regard cracks in images as sparse anomalies and formulate the detection task as an LRR problem with few hypotheses. Suppose we have n images $\boldsymbol{I}_1, \boldsymbol{I}_2, \cdots, \boldsymbol{I}_n$ in a sequence, w and h are the width and height of images respectively. For each image, we concatenate all columns to generate a one-dimensional vector (here we note this operation as $vec()$) then stack all vectors to compose a hyper-dimensional matrix $\boldsymbol{D}$ ($w \times h$ for rows, n for columns). With the aforementioned properties of cracks in inspection image sequences, $\boldsymbol{D}$ should have a potential low rank property, which we formulate as the superposition of a low rank matrix $\boldsymbol{L}$ and a sparse matrix $\boldsymbol{S}$, as Eq. 1.

$$\boldsymbol{D} = [vec(\boldsymbol{I}_1)|vec(\boldsymbol{I}_2)|\cdots|vec(\boldsymbol{I}_n)] = \boldsymbol{L} + \boldsymbol{S} \tag{1}$$

If cracks are crisply split from the background, the rank of $\boldsymbol{L}$ can be optimized with objective function Eq. 2.

$$\min_{L,S} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 \quad \text{subject to } \boldsymbol{D} = \boldsymbol{L} + \boldsymbol{S}, \tag{2}$$

where the nuclear norm is applied to enforce the low rank constraint, the ℓ_1-norm is used to approximate non-zero items in the sparse part. Here, λ is a value to balance the decomposition. The main challenge in efficiently solving Eq. 2 is coping with the constraint $\boldsymbol{D} = \boldsymbol{L} + \boldsymbol{S}$. We adopt the inexact augmented Lagrange multiplier (ALM) method [74], where the augmented Lagrangian function is defined in Eq. 3:

$$\begin{aligned}\mathcal{L}_\mu(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) \doteq \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{\Lambda}, \boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\rangle \\ + \frac{\mu}{2}\|\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}\|_F^2,\end{aligned} \tag{3}$$

where $\boldsymbol{\Lambda}$ is the Lagrange multiplier, μ is a positive scalar, $\|\cdot\|_F$ is the Frobenius norm. We solve the problem by repeatedly setting and updating $\boldsymbol{\Lambda}$, as written in Eq. 4:

$$(\boldsymbol{L}_{k+1}, \boldsymbol{S}_{k+1}) = \arg\min_{\boldsymbol{L},\boldsymbol{S}} \mathcal{L}_\mu(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}_k) \tag{4}$$

For convenience, we let $\mathcal{S}_\tau[x] : \mathbb{R} \to \mathbb{R}$ denote the shrinkage operator for variant x with a constant τ, as written in Eq. 5.

$$\mathcal{S}_\tau[x] = \text{sgn}(x)\max(|x| - \tau, 0) \tag{5}$$

Then we extend it to matrices and denote the singular value thresholding operator $\mathcal{D}_\tau(\boldsymbol{D}) = \boldsymbol{U}\mathcal{S}_\tau(\boldsymbol{\Sigma})\boldsymbol{V}^*$, where $\boldsymbol{D} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ is any singular value decomposition, which is typically the major computational burden of LRR for the hyper-dimensional matrix and can be optimized in both algorithm and implementation. Thus, we get a practical strategy as written in Eq. 6. We first minimize $\mathcal{L}_\mu$ with respect to $\boldsymbol{L}$ (fixing $\boldsymbol{S}$), then minimize $\mathcal{L}_\mu$ with respect to $\boldsymbol{S}$ (fixing $\boldsymbol{L}$), and

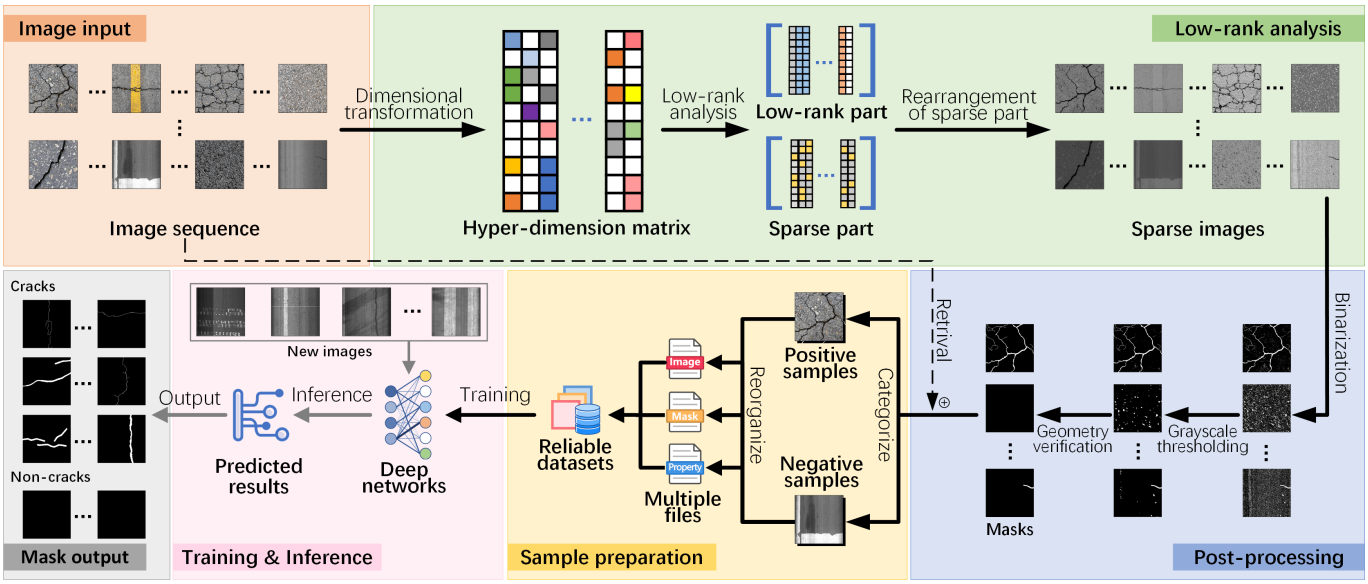

Fig. 3. Diagram of our framework for pavement crack detection, which mainly involves low rank representation and deep learning. We first process input images (orange part) and conduct low rank analysis (green part) to obtain sparse images, following the simple post-processing with grayscale and geometry clues for binary masks (blue part). Then, we train our network on automatically generated samples from LRR (yellow and pink parts). Finally, the trained network process new images and output masks as shown in gray arrows (pink and gray parts).

Algorithm 1 LRR for Crack Detection via Inexact ALM

Initialize: $\boldsymbol{S_0} = \boldsymbol{\Lambda_0} = 0, \mu > 0$.
While not converged **do**
  Compute $\boldsymbol{L}_{k+1} = \mathcal{D}_{1/\mu}(\boldsymbol{D} - \boldsymbol{S}_k - \mu^{-1}\boldsymbol{\Lambda}_k)$;
  Compute $\boldsymbol{S}_{k+1} = \mathcal{S}_{\lambda/\mu}(\boldsymbol{D} - \boldsymbol{L}_{k+1} - \mu^{-1}\boldsymbol{\Lambda}_k)$;
  Compute $\boldsymbol{\Lambda}_{k+1} = \boldsymbol{\Lambda}_k + \mu(\boldsymbol{L}_{k+1} + \boldsymbol{S}_{k+1} - \boldsymbol{D})$;
end while
Output: $\boldsymbol{L}_k$; $\boldsymbol{S}_k$.

finally update the Lagrange multiplier matrix $\boldsymbol{\Lambda}$ based on the residual $\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{D}$. We summarize it in Algorithm 1.

$$\begin{cases} \arg\min_{\boldsymbol{S}} \mathcal{L}_\mu(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \mathcal{S}_{\lambda/\mu}(\boldsymbol{D} - \boldsymbol{L} - \mu^{-1}\boldsymbol{\Lambda}) \\ \arg\min_{\boldsymbol{L}} \mathcal{L}_\mu(\boldsymbol{L}, \boldsymbol{S}, \boldsymbol{\Lambda}) = \mathcal{D}_{1/\mu}(\boldsymbol{D} - \boldsymbol{S} - \mu^{-1}\boldsymbol{\Lambda}) \end{cases} \tag{6}$$

By iterative optimization, we can effectively eliminate complex backgrounds in the pavement (such as uneven lighting) and detect cracks efficiently, where the object function converged to the global optimal and does not change significantly if you continue conduct LRR on output sparse images. Note that for LRR, simple post-processing is required for sparse images to produce the final binary mask, as many entries in $\boldsymbol{S}$ may contain vanishingly small values. For a more comprehensive introduction to low rank theories, readers can refer to the book [75] and our previous work [73].

B. Post-Processing and Sample Preparation

We obtain binary pixel-wise masks by post-processing on sparse images from LRR with two stages: grayscale thresholding and geometry verification as shown in Fig. 4. The former generates initial binary masks and the latter further verifies crack candidates with geometric properties. Then, we categorize masks and generate our crack dataset.

1) Grayscale Thresholding: We design a four-step procedure including grayscale reassignment, grayscale stretching, adaptive binarization, and noise removal as shown in Fig. 4. In grayscale reassignment, we first calculate the histogram of the sparse image and find the mode grayscale G_m. Then we reassign the grayscale of each pixel by comparing its original grayscale G_{ori} and mode grayscale G_m with Eq. 7.

$$G_{re} = \begin{cases} G_{ori} & G_{ori} < G_m \\ G_m & G_{ori} \geqslant G_m \end{cases} \tag{7}$$

With reassignment, we suppress noisy pixels brighter than G_m and conduct stretching for higher contrast. We adopt linear percent stretching due to its robustness toward extreme values. Specifically, we trim extreme values from both ends of the histogram with two percentage and stretch pixels inside the range, retaining the idea that most information can be carried in the range of 2σ according to Gaussian distribution.

After grayscale reassignment and stretching, we get enhanced sparse images that are very similar to binary masks. Then we conduct binarization accompanied by noise filtering. For strip cracks, we adopt simple thresholding and median filtering. For string cracks, we propose a sophisticated strategy with dual binarization and bit-wise operation due to fragile visual features. With loose and strict thresholding, we get masks with and without noises, respectively. Then we conduct bit-wise intersection on strict masks and loose masks, where details are preserved and noises are filtered effectively.

2) Geometry Verification: From the view of geometry, strip and string cracks share similarities in appearance, therefore we propose a hierarchical strategy for judgment based on weighted voting of multiple features. We first search contours and extract skeletons on masks as shown in the black line and white dotted line in Fig. 5, respectively. Then we sample n points (P_1, P_2, ..., P_n, blue points in Fig. 5) on the skeleton

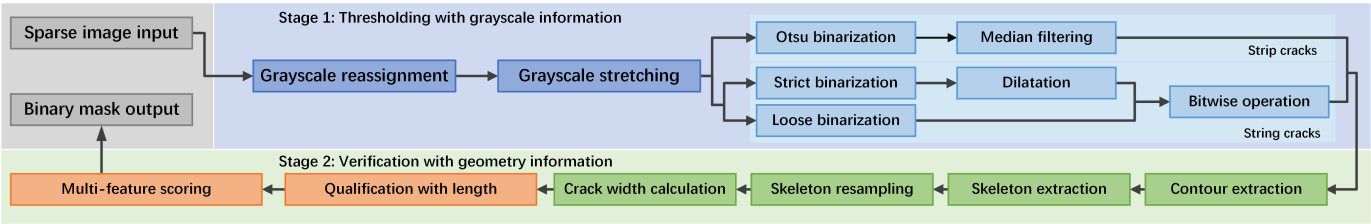

Fig. 4. Flowchart of simple post-processing for LRR, including stage 1 and stage 2, where we get binary masks from sparse images (gray blocks). We first conduct grayscale adjustment (dark blue blocks) and initial thresholding for different cracks (pale blue blocks). Then, we calculate the selected geometry indicators (green blocks) and judge crack candidates with the length and weighted scores (orange blocks).

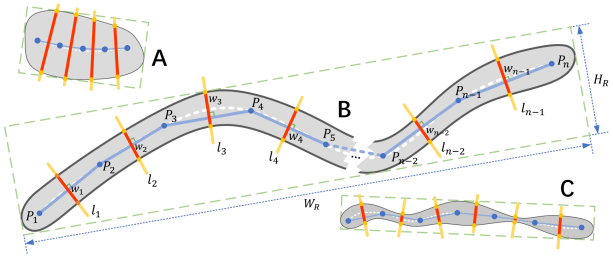

Fig. 5. Illustration of geometry verification and the appearance are exaggerated to some extent. We think the candidates A and C are not crack due to their large width and variance. We regard candidate B as a crack due to its enough length, small width, and variance.

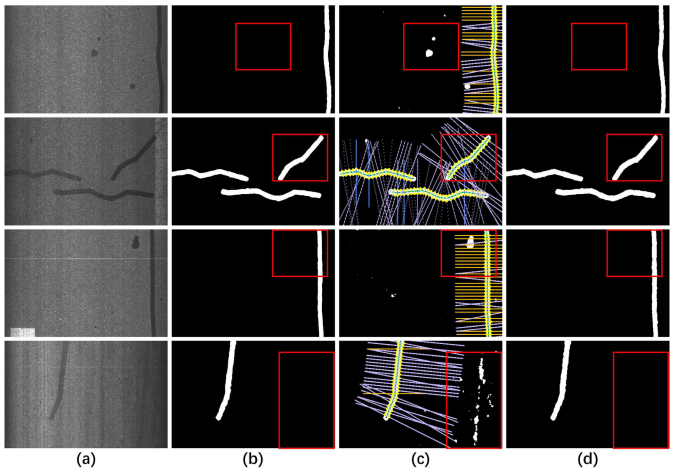

Fig. 6. We eliminate interference effectively while keeping cracks as shown in red rectangles. (a) Images. (b) Ground truth. (c) Initial masks without geometric filtering. (d) Final masks after filtering.

with fixed intervals and construct a complete graph. We find the minimal spanning tree with Kruskal algorithm and regard the route connecting all nodes with minimum total length as the optimal path (pale blue line in Fig. 5). Each segment can be regarded as a local linear fit to the skeleton in such a multi-segment polyline. Then we find the perpendicular bisector l_i $(i = 1, 2, \cdots, n-1)$ for each segment (yellow line in Fig. 5) and calculate the distance (read line in Fig. 5) between intersection points (yellow points in Fig. 5), and let it represent the width w_i of this segment. For a crack candidate with n sample points on a skeleton, we calculate its total length I_L, crack width I_W, and width variance I_V as Eq. 8.

$$\begin{cases} I_L = \sum_{i=1}^{n-1} \sqrt{(P_{i+1} - P_i)^2} \\ I_W = median(w_i), \quad i = 1, 2, \cdots, n-1 \\ I_V = \dfrac{\sum_{i=1}^{n-1} (w_i - w_m)^2}{n-1} \end{cases} \tag{8}$$

where $median()$ indicates the median width of the skeleton, and $w_m = \frac{1}{n-1} \sum_{i=1}^{n-1} w_i$ represents the mean width. We believe a sufficient length is a prerequisite for further judgment and qualify a candidate if its I_L is longer than the given threshold T_L. Then we score this candidate with the following geometry indicators: (a) I_W; (b) I_V; (c) I_A - the aspect ratio of the minimum enclosing rectangle, reflecting the general shape of crack; (d) I_R - the area ratio of the crack region and minimum enclosing rectangle, suggesting the similarity to a rectangle; (e) I_D - ratio of the square of the crack perimeter to its area, expressing the similarity to a circle. I_A, I_R, I_D are calculated with certain geometry properties as Eq. 9.

$$I_A = \frac{W_R}{H_R}, I_R = \frac{S_C}{S_R}, I_D = \frac{P_C^2}{S_C}, \tag{9}$$

where W_R, H_R, S_R are the width, height, and area of the minimum enclosing rectangle; P_C, S_C are the perimeter and area of the crack candidate region. The overall judgment score J_S is calculated with selected indicators as Eq. 10.

$$\begin{aligned} J_S = \alpha \mathbb{I}(I_W \le T_W) + \beta \mathbb{I}(I_V \le T_V) + \gamma \mathbb{I}(I_A \ge T_A) \\ + \delta \mathbb{I}(I_R \le T_R) + \varepsilon \mathbb{I}(I_D \ge T_D), \end{aligned} \tag{10}$$

where α, β, γ, δ, ε are weighting coefficients and T_W, T_V, T_A, T_R, T_D are thresholds for corresponding indicators. $\mathbb{I}(\cdot)$ is the indicator function. Finally, we get the score J_S and regard the candidate as a crack if it is larger than the given threshold T_S. Fig. 6 shows some examples of geometry filtering.

3) Sample Preparation: Compared with manual labeling, we obtain extensive pixel-wise crack masks (positive samples) and non-crack masks (negative samples) automatically from LRR, which significantly improves the efficiency and achieves dataset generation without or little manual intervention. Moreover, compared with common organization strategies adopted in current public crack datasets, we follow practices in famous autonomous driving dataset KITTI [76], and the well-known simultaneous localization and mapping (SLAM) dataset EuRoC [77]. We make several improvements for better application, where the most notable one is that

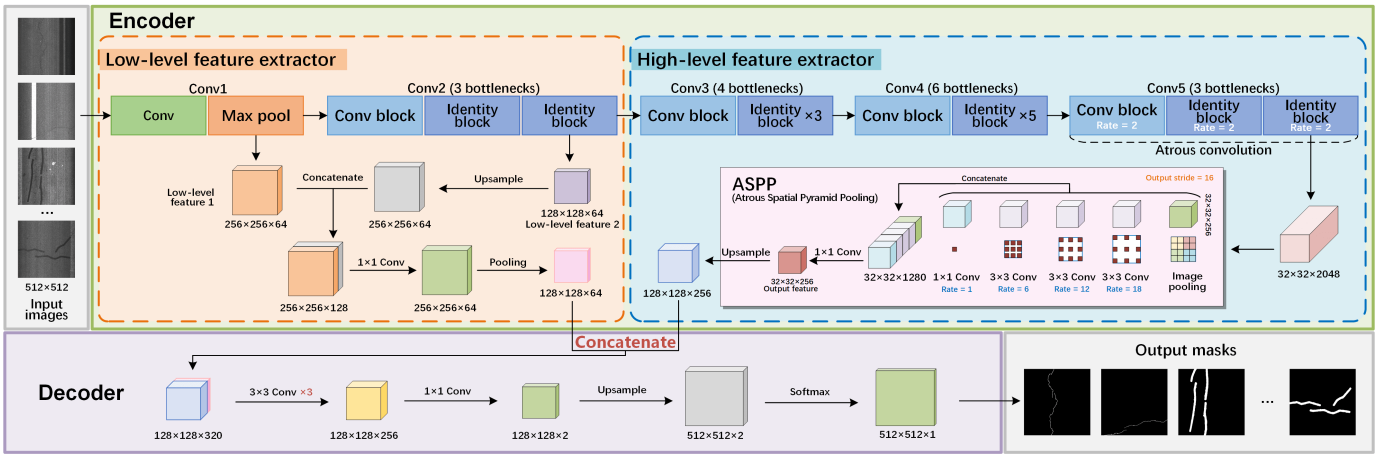

Fig. 7. Our DCNN for crack detection. We extract multi-level features in Encoder, while leveraging on ASPP for better receptive field and contextual information. Then we concatenate the low-level and high-level feature maps for Decoder, and output predicted binary masks.

TABLE II

PROPERTIES IN METADATA FILE FOR BRIEF DESCRIPTION OF IMAGES AND CRACKS IN OUR PEARL DATASET

Category	Property	Type	Description
Image -related	Name	Char	Name of the image file.
	Width	Int	Image width in pixel.
	Height	Int	Image height in pixel.
	Contain Crack	Bool	Description on whether an image contains crack or not.
	Number	Int	Total number of cracks in image.
	Difficulty	Char	Difficulty level of this scene, enumeration: easy, medium and hard.
	Division	Char	Division for deep learning, enumeration: train, validation and testing.
Crack -related	ID	Int	Crack index in an image.
	Type	Int	Enumeration: 0-Non-crack, 1-String crack, 2-Strip crack.
	BBox	Int	The top-left and bottom-right coordinates of the bounding box.

we organize samples in a flexible and scalable manner of "image + metadata". Each sample in our dataset has a raw image (stored in JPEG format), a pixel-level mask (stored in lossless PNG format), and a metadata file (stored in XML format). In the metadata file, we record various properties for a fast grasp of major information, including the number of cracks, difficulty, crack type, and bounding box range, as listed in Table II. We also develop graphical toolkits with common functions for better use of our dataset. For example, we can easily filter and reorganize the samples according to recorded property of difficulty levels.

C. DCNN for Crack Detection

We believe that it is of great benefit to regard crack detection as semantic segmentation rather than object detection or other tasks. First, semantic segmentation methods achieve the highest accuracy with pixel-wise masks while only bounding boxes could be given by object detection methods. Second, cracks vary in grayscale, shape, and size, it is hard and unfeasible to define them as certain objects with a similar appearance. With quickly advancing in computer vision, semantic segmentation based on deep learning has achieved rapid development and performs robustly on various datasets. DeepLabV3+ model [78] is the latest improved version of the DeepLab series of networks leveraging contextual information. Inspired by them, we adopt the skip-connection and propose an end-to-end neural network with ASPP for pavement crack detection. We extract and concatenate low and high-level features to exploit the deep pattern of cracks and the background, rather than with simple structures adopted in many existing works. Moreover, we adopt atrous convolution with different scales for better perception without losing image details. The network follows encoder-decoder structure, as shown in Fig. 7.

In the Encoder part, we use both low-level and high-level feature extractors due to the complexity and diversity of the pavement and cracks. The low-level feature extractor retains a large amount of global information including the road background information and grayscale patterns which are critical for crack detection. Therefore, we choose ResNet50 [79] as the backbone structure of the first two convolution layers. Meanwhile, to preserve a sufficient amount of initial global information, we concatenate output feature maps from the first and second convolution layers. Then this feature map passes through an 1×1 convolution with 64 channels, which has the same size as the high-level feature map. The high-level feature extractor mainly exploits deep patterns of cracks and understands semantics in images through its deeper structure. The complete five convolution layers in ResNet50 are adopted, each of which contains a different number of bottleneck blocks. Similar to the SDDNet [53], we adopt atrous convolution with a dilation rate of two in the fifth layer for a wider receptive field along with different scales rather than common image pyramid construction due to the loss of tiny cracks during resampling. Then the ASPP module takes the feature maps extracted from the first five convolution layers and aggregates multi-scale contextual information. For details of the ASPP module, see [80]. We use 3×3 kernels but with different atrous rates to capture different size features. The ASPP module has four branches with rates of 1, 6, 12,

18 and one pooling layer. These five feature maps in ASPP are then concatenated into one feature map with 256 channels and pass through an 1×1 convolution layer before upsampling to the same size as the feature map from low-level.

In the Decoder part, we achieve end-to-end pixel-level semantic segmentation by fusing the high-level and low-level feature maps from Encoder part. A 3×3 convolutional dimensionality reduction calculation is performed on the connected feature map three times to obtain a new feature map with 256 channels. The obtained feature map is passed through an 1×1 convolution layer to reduce dimension to two. Then the obtained features are upsampled to the same size as the input image. After that, a softmax normalization is performed on the upsampled feature map for pseudoprobabilities of the crack class. Finally, we classify each pixel in the image and get the final prediction result for crack detection.

With these structures, our network has a strong ability of crack description and generalization in various samples. Extent experiments on images from different fields such as industrial manufacturing further demonstrates the good performance of proposed neural network, as shown in Fig. 10.

D. Implementation Detail

1) LRR for Crack Detection: Both low rank analysis and post-processing are implemented with C++ language in a modular way, leveraging on OpenCV and Eigen libraries for image processing and matrix operation. Moreover, we manage parameters with configuration files and package the program as a dynamic link library (DLL) on the Windows platform. We calculate the parameter λ in Eq. 2 as suggested [71]:

$$\lambda = 1/\sqrt{max(n_1, n_2)}, \tag{11}$$

where n_1, n_2 are the dimension of a matrix respectively. For practical problems, it is often possible to improve performance by choosing λ according to prior knowledge about the problem. Here we adopt the weight calculated by Eq. 11 for strip cracks and set 0.04 as empirical for string cracks. For SVD operations in inexact ALM method, we perform partial SVD [81] for a smaller memory cost rather than the full SVD.

In post-processing, the geometric thresholds are fairly loose and can be re-calculated according to various applications, since they are only for the sample generation before the training stage and have little effect on the final results from the proposed network. Specifically, we set $T_L = 100$, $T_W = 30$ (string crack) or 100 (strip crack), $T_V = 5$ with the resolution of $1mm/pixel$ in our dataset. For the remaining thresholds, we take the given T_L, T_W and T_V into Eq. 9 to estimate their values. For the weighting coefficients, we set $\alpha = 0.3$, $\beta = 0.3$, $\gamma = 0.1$, $\delta = 0.1$, $\varepsilon = 0.2$.

2) Training for DCNN: We use grayscale patches of size 512×512 as inputs and augment training samples in our dataset with random strategies (flipping and grayscale stretching). After that, We have 112,478 samples (56,810 positive samples and 55,668 negative samples respectively) for training. Moreover, we adopt the Adam optimizer and follow the mini-batch strategy with a batch size of 8 for better learning. Finally, we train the neural network 150,000 times on the NVIDIA Titan RTX GPU for about twenty hours.

TABLE III
PUBLIC PAVEMENT CRACK DATASETS WITH IMAGE NUMBER, RESOLUTION, AND ANNOTATION INFORMATION

Dataset	Number	Resolution	Annotation	Image
CrackTree [28] (2012)	206	800 × 600 (0.48MP)	Pixel-wise (no width)	
CrackIT [36] (2014)	56	2048 × 1536 (3.14MP)	Block-wise	
CrackForest [37] (2016)	118	480 × 320 (0.15MP)	Pixel-wise	
AELLT [27] (2016)	66	991 × 462 (0.45MP)	Pixel-wise	
GAPsV2 [45] (2019)	2468	1920 × 1080 (2.07MP)	Bounding box (Some are pixel-wise)	
CRACK500 [46] (2020)	500	2560 × 1440 (3.68MP)	Pixel-wise	
Our Dataset - PEARL (2023)	14812	2048 × 2048 (4.19MP)	Pixel-wise	

IV. EXPERIMENTS AND ANALYSIS

We now apply the proposed method and compare it with representative traditional and learning methods on various datasets including public ones and ours. Extensive experiments demonstrate that our method significantly outperforms representative approaches and works effectively on a wide range of complex scenarios. We believe that such results could be very valuable to users, especially to the traffic management sectors.

A. Experiments Settings

1) Evaluation Datasets: We collect representative crack datasets as follows and summarize them in Table III. We include the image number, the image size (width × height) and corresponding Megapixel (MP), the annotation level (pixel, block or bounding box), and the sample for a fast grasp.

CrackTree (D-CT) [28] includes 206 pavement images involving some hard scenes such as shadows and low contrast. It has pixel-wise annotation but without crack width.

CrackIT (D-CI) [36] contains 56 images with a resolution of $1mm/pixel$. Ground truth of crack is labeled in blocks (75×75 pixel) and provided with the CrackIT toolbox.

CrackForest (D-CF) [37] is composed of 118 images taken in Beijing, China with hand-labeled ground truth contours. The dataset is split into 60%/40% for training and testing.

AELLT (D-ALT) [27] combines Aigle-RN, ESAR, LCMS, LRIS, and Tempest2 due to few pavement images (38, 15, 5, 3, 5 images with pixel-wise annotation) in each dataset.

GAPsV2 (D-GAP2) [45] includes 2,468 images, covering pavement distress such as cracks, potholes, and inlaid patches. All distress is enclosed by a bounding box as ground truth.

TABLE IV
METRICS FOR CRACK DETECTION IN IMAGE-LEVEL CLASSIFICATION AND PIXEL-LEVEL SEGMENTATION

Metrics	Image-level Classification	Pixel-level Segmentation
TP	Detected crack images that contain cracks indeed.	Intersection of crack region in ground truth mask and estimated mask.
FP	Detected crack images that do not contain cracks in fact.	Wrongly detected crack region in estimated mask.
TN	Detected non-crack images that do not contain cracks indeed.	Intersection of non-crack region in ground truth mask and estimated mask.
FN	Detected non-crack images that contain cracks in fact.	Missing detected crack region in ground truth mask.

TABLE V
COMPARISON OF IMAGE-LEVEL CLASSIFICATION (F1 SCORE), BEST RESULTS ARE MARKED IN BOLD

Datasets	M-CI	M-CF	M-FPN	M-Ours
D-CT (800 × 600)	91.01%	**100%**	99.51%	**100%**
D-CI (2048 × 1536)	96.97%	92.31%	92.31%	**100%**
D-CF (480 × 320)	**100%**	95.58%	99.57%	**100%**
D-ALT (991 × 462)	85.96%	99.24%	99.13%	**99.44%**
D-GAP2 (1920 × 1080)	95.53%	98.91%	99.31%	**99.66%**
D-CK500 (2560 × 1440)	98.67%	97.45%	99.60%	**100%**
D-PEARL (2048 × 2048)	56.64%	58.47%	63.15%	**98.20%**
Mean Value ↑	73.61%	75.88%	78.83%	**98.87%**

CRACK500 (D-CK500) [46] collects 500 images with pixel-wise annotation in Temple University using cell phones, and is divided into training, validation, and testing subsets.

Our Dataset involves 14,812 pavement images captured by professional inspection vehicles in China and we call it **PEARL** (**P**ix**E**l **A**nnotated c**R**ack from **L**RR). The dataset is divided into training, validation, and testing subsets with around 60%, 10%, and 30% percentages. We provide both pixel-wise and bounding box annotations for each sample.

2) Evaluation Metrics: We consider that complete crack detection involves two aspects: image-level classification and pixel-level segmentation, where the former indicates whether an image contains cracks and the latter describes the pixel-wise range. To achieve a unified quantitative evaluation for both levels in a coarse-to-fine manner, we define the TP (true positive), FP (false positive), TN (true negative), and FN (false negative) for these two aspects in Table IV, respectively. Then, we calculate *Precision*, *Recall* and *F-Score* using Eq. 12, which are widely adopted in aforementioned works.

$$\begin{cases} Pr = \dfrac{TP}{TP + FP} \\ Re = \dfrac{TP}{TP + FN} \\ F_{score} = (1+\omega^2) \cdot \dfrac{Precision \cdot Recall}{\omega^2 \cdot (Precision + Recall)}, \end{cases} \tag{12}$$

where ω is a weight adjusting between $Precision$ and $Recall$, and we calculate F1 score by setting $\omega = 1$. Moreover, for better evaluation of pixel-wise segmentation, we adopt the mIoU metric [55]. It should be noticed that CrackIT and GAPsV2 do not have pixel-level annotation, therefore we convert estimated binary masks to corresponding block-wise masks and bounding boxes for a fair evaluation. Moreover, we divide large test samples into several blocks with 512 × 512 pixels for our method, then merge them for the same size as the input original images before final evaluation.

3) Compared Methods: Since few open-source methods exists in the field of crack detection, we chose two representative traditional methods and one deep learning method as following. We also compare with other methods evaluated on public datasets. Moreover, we replace the backbone of our proposed network with classic structures including FCN, FPN, U-Net, Inception and ResNet for ablation studies.

CrackIT (M-CI) [36] is a comprehensive set of image processing algorithms for the detection and characterization of pavement distresses, which is implemented in Matlab.

CrackForest (M-CF) [37] characterizes cracks and discerns them from noises based on random structured forests method. It involves training and detection stages.

FPHBN (M-FPN) [46] integrates multi-level features with feature pyramid and hierarchical boosting network for crack detection, which is implemented in Caffe framework.

Methods on public datasets. Some researchers adopt minimal path selection (**M-MPS**) [27] for crack detection and evaluate on Aigle-RN dataset. While some propose a neural network (**M-GAP**) [45] and test on GAPsV2 dataset.

B. Results and Analysis

1) Evaluation for Image-Level Classification: Since compared methods do not have an explicit strategy for the image-level classification, we regard an image as non-crack if its mask is pure blank or with fewer crack pixels than the given threshold. Image-level classification results are summarized in Table V. Compared methods generally achieve high F1 scores besides on our dataset. Method M-CI and M-CF even achieve 100% F1 score on D-CF and D-CT datasets, respectively. However, on our dataset, the F1 score of traditional methods (M-CI and M-CF) decreases to 57.56% approximately. The reason is that we have hard samples with low intensity or strong noises, which brings challenges for these methods leveraging on statistical histograms or hand-crafted features. Take M-CI as an example, the TP, FP, TN, FN are 1277, 1833, 877, 122 on our test dataset. Many noise pixels are categorized as "crack", and the whole image is regarded as a "crack image", resulting in an increase of FP and significant degradation in classification performance. For the deep learning method M-FPN, it generally achieves a better F1 score than traditional methods due to the stronger ability in crack feature description. However, it is still not satisfying on our dataset with diverse samples due to the weak generalization led by a simple network structure. As a comparison, our method achieves excellent performance on all test datasets despite that we only train the network on our dataset. We even achieve 100% F1 scores on D-CT, D-CI, D-CF and D-CK500 datasets. The reason is that our dataset contains rich scenes and samples

TABLE VI
COMPARISON OF PIXEL-LEVEL SEGMENTATION PERFORMANCE (MIOU AND F1 SCORE), BEST RESULTS ARE MARKED IN BOLD. ONLY F1 SCORES OF M-MPS AND M-GAP ARE PROVIDED IN [27], [45]

Datasets	mIoU				F1 Score					
	M-CI	M-CF	M-FPN	M-Ours	M-CI	M-CF	M-FPN	M-Ours	M-MPS	M-GAP
D-CT (800 × 600)	29.08%	44.87%	51.59%	**80.37%**	31.38%	49.01%	54.74%	**82.88%**	-	-
D-CI (2048 × 1536)	76.39%	31.56%	41.94%	**86.48%**	88.11%	51.49%	47.69%	**90.88%**	-	-
D-CF (480 × 320)	60.29%	68.53%	41.15%	**82.52%**	72.51%	85.72%	43.56%	**86.10%**	-	-
D-ALT (991 × 462)	58.38%	50.22%	27.32%	**82.09%**	76.02%	69.16%	38.55%	**84.63%**	63.05%	-
D-GAP2 (1920 × 1080)	42.40%	31.18%	50.82%	**86.28%**	45.62%	34.98%	64.43%	**93.36%**	-	90.40%
D-CK500 (2560 × 1440)	28.27%	35.93%	62.26%	**81.22%**	29.41%	37.36%	64.91%	**82.88%**	-	-
D-PEARL (2048 × 2048)	30.27%	42.65%	56.34%	**93.66%**	52.86%	49.21%	58.57%	**97.75%**	-	-
Mean Value ↑	34.93%	39.05%	54.28%	**89.88%**	49.27%	44.71%	60.24%	**94.70%**	-	-

of different difficulties. Besides that, the well-designed low-level and high-level structure enables the network to grasp crack features from different aspects. The performance of the proposed network on our dataset is slightly lower than other datasets due to very hard samples. Generally, the classification performance focuses on the whole image level rather than the pixel level results. It reflects the robustness toward external interferences such as dark ruts or stains. It is worth noting that if a method tends to identify noise as cracks, the performance may be good on datasets with only crack images.

2) Evaluation for Pixel-Level Segmentation: Segmentation results (both mIoU and F1 Score) of compared methods are summarized in Table VI. The performance of compared methods varies on different datasets, while our network achieves the best segmentation performance on all test datasets. For traditional methods M-CI and M-CF, the performance is unstable and is fluctuated by test data significantly. They achieve good performance on certain datasets but fail on unfamiliar or hard samples. For example, M-CI achieves the 76.39% mIoU and 88.11% F1 score on D-CI while 28.27% mIoU and 29.41% F1 score on D-CK500. M-CF achieves the 68.53% mIoU and 85.72% F1 score on D-CF while 31.18% mIoU and 34.98% F1 score on D-GAP2. These results further suggests that methods based on hand-crafted features perform well on datasets similar to target scenes. However, in hard or complex scenes it usually fails due to the limited ability of crack feature extraction. For deep learning method M-FPN, it has a significant improvement on D-GAP2 and D-CK500 dataset compared with traditional methods (M-CI and M-CF), which confirms the better feature extraction ability. However, on datasets that have a large difference in appearance with D-CK500 (such as D-ALT), the mIoU and F1 score falls dramatically to 27.32% and 38.55%, respectively, due to incomplete learning of crack features. Moreover, M-FPN outputs the probability map that needs to be further processed and is not accurate enough for pixel-wise segmentation evaluation, resulting in a fairly low performance on all datasets. For M-MPS and M-GAP methods, we cannot test them on collected datasets due to unavailable source code, where we mark "−" in Table. VI. For M-MPS, they test their method on 38 images from the Aigle-RN dataset, where they get a precision of around 72.04% and recall of 64.24% [27]. Therefore, we calculate the corresponding F1 score of 67.92% for the approximation of M-MPS on D-ALT. For M-GAP, they achieve the best performance with the ResNet34 backbone. The network is trained for 80 epochs on the 50K subset of GAPsV2 dataset with a patch size of 160 × 160 [45]. For the proposed method, we achieve 93.66% mIoU and 97.75% F1 score, evaluated with generated binary masks in our PEARL dataset. On other datasets, we also achieve the best segmentation performance. Compared with other methods, our method performs stably on all datasets with good generalization, which is the result of diverse training samples in our dataset and the well-designed network structure. Segmentation results of crack and non-crack samples in various datasets are shown in Fig. 8.

3) Ablation Study and Further Analysis: To further analyze the performance, we conduct the following experiments.

a) Ablation on LRR: We compare classification and segmentation results generated by LRR in the first stage as summarized in Table VII. The binarization step is first replaced with classic Otsu (LRR-BinOtsu) and compared with our method (LRR-BinOurs), where both of them are without geometry verification. LRR-BinOurs surpasses LRR-BinOtsu by a large margin especially in pixel-level segmentation due to the robustness to many speckle noise, suggesting its superiority for pavement images. Then we enhance LRR-BinOtsu with Hough Transform [82], [83] to give simple geometry constraints (LRR-Ot.Ho.) and obtain better results, which demonstrate the necessarity of geometry verification. Finally, we propose tailored binarization and geometry verification for pavement cracks (LRR-FullOurs), and conduct experiments. With more precise and rigid filtering for interference, LRR-FullOurs performs better than LRR-Ot.Ho., which is the result of suitable integration of grayscale and geometry constraints. The outperformance of LRR in automatic crack sample generation compared with other traditional methods is demonstrated in accuracy, efficiency, and generalization with experimental results. First, we achieve the best performance (86.59% F1 score for image-level classification, 81.21% mIoU and 82.35% F1 score for pixel-level segmentation) on collected sequential images compared with two traditional methods (M-CI and M-CF), which is of great importance for reducing the labor cost and workload in practice. Second, we achieve higher FPS (0.871) on our dataset without GPU compared with M-CI and M-CF (Table IX), which is benefit from the fast theoretical convergence, the batch-processing manner, and

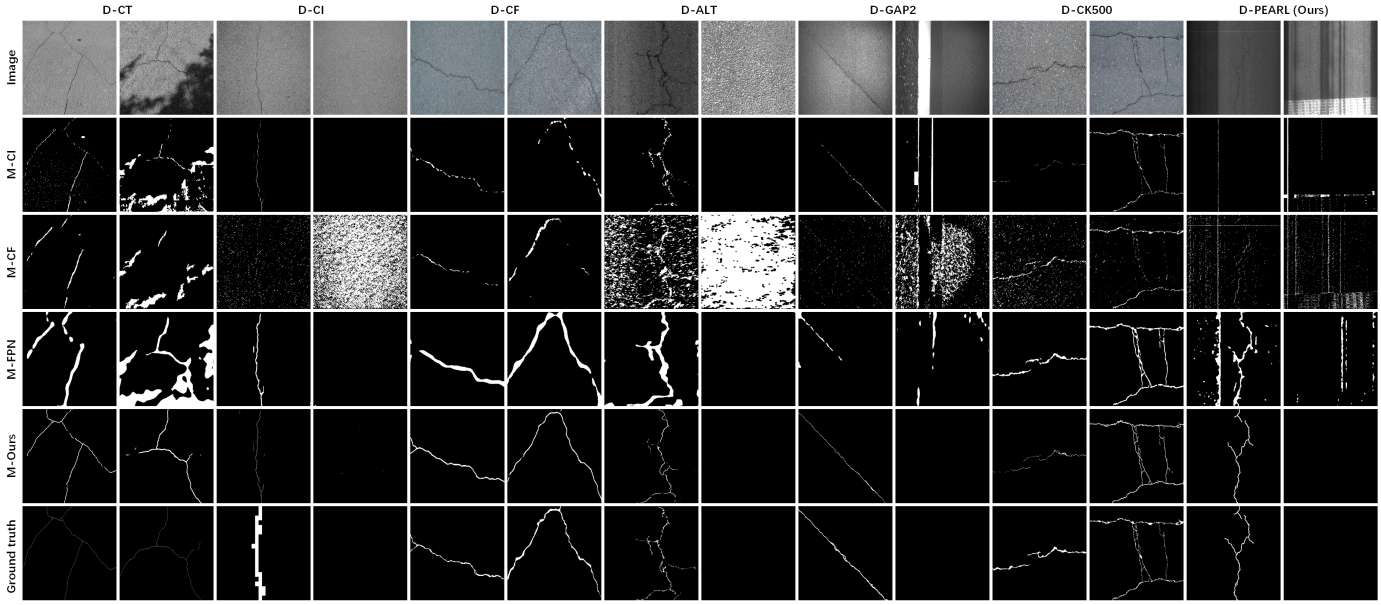

Fig. 8. Comparison of different methods on crack and non-crack samples from various datasets. Our method (M-Ours) performs best on all test datasets with the highest classification and segmentation performance (mIoU and F1 score), while M-CF and M-FPN are with much noise. The ground truth annotation of D-CI is not in pixel level but with blocks of 75 × 75 pixels.

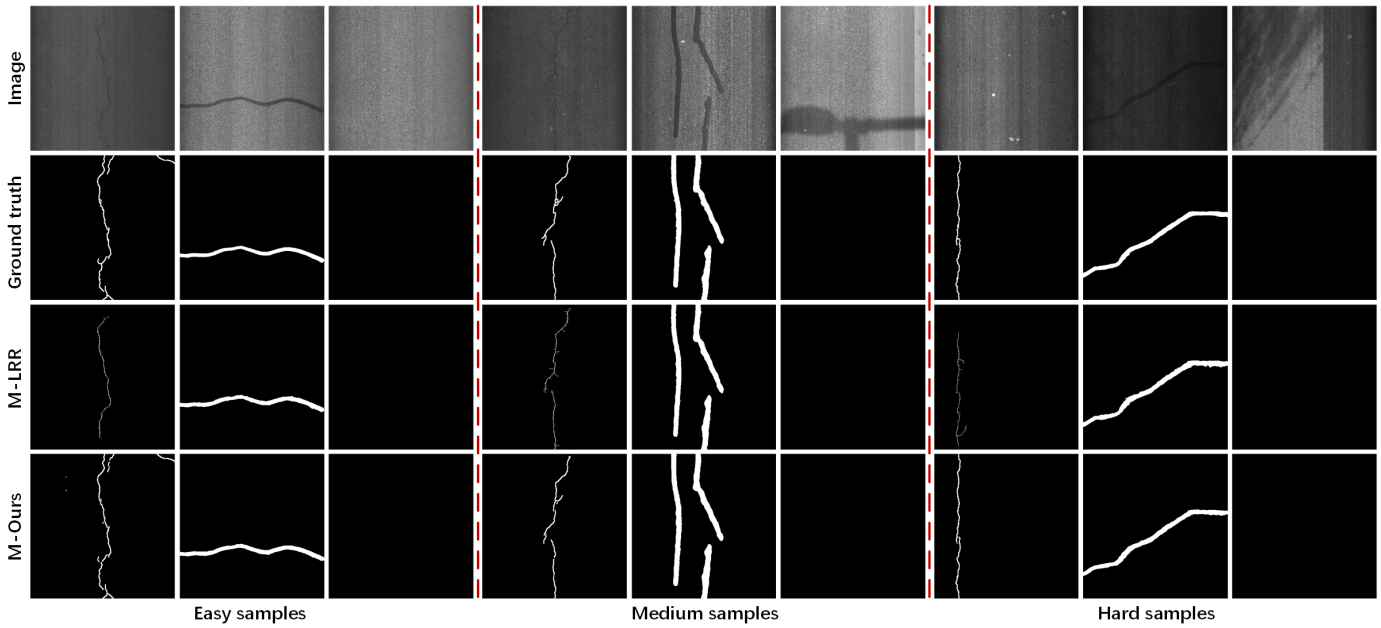

Fig. 9. Comparison of the proposed LRR and neural network on our PEARL dataset. M-LRR achieves satisfying performance, while the neural network (M-Ours) performs better. Moreover, the network has a more stable performance on samples of different difficulty levels.

the acceleration of computation. Third, we achieve better generalization to various images compared with traditional methods due to the weak assumption of LRR on data.

b) Ablation on DCNN: To verify the influence of different depth and perception field, we replace the backbone of the proposed network with FCN, FPN, UNet, InceptionV3, ResNet101, ResNet152, ResNet200, and ResNet50, which are widely adopted in semantic segmentation and crack detection [46]. According to experimental results, the network with ResNet50 (Ours-Res50) performs best in test pavement images. Specifically, we find that the deeper network (ResNet50-200) usually performs better compared with simple structures, such as FCN, FPN and UNet. But the low-level features that are potential for crack detection may get lost during the very deep convolution, resulting the degradation of performance. From the view of perception field, structures with wider field typically prevent the loss of details and lead to a better accuracy. Therefore, we select the ResNet50 as our backbone and enhance with ASPP, which achieves the best performance in both image-level classification and pixel-level labelling. In addition, we further analyze the performance of our network on samples of different difficulties, and re-calculate classification and segmentation results, as shown in Table VIII, some visual results are shown in Fig. 9.

TABLE VII

DIFFERENT ABLATION STUDY RESULTS OF OUR LRR AND NEURAL NETWORK ON THE PROPOSED PEARL DATASET

Methods	Image-level Classification			Pixel-level Segmentation			
	Pr.	Re.	F1	Pr.	Re.	F1	mIoU
LRR-BinOtsu	46.67%	100%	63.63%	1.98%	98.44%	3.83%	1.98%
LRR-BinOurs	63.64%	100%	77.78%	83.74%	87.99%	78.56%	68.79%
LRR-Ot.Ho.	50.95%	100%	68.11%	51.39%	95.27%	68.43%	52.11%
LRR-FullOurs	77.23%	98.53%	86.59%	87.31%	79.45%	82.35%	81.21%
Ours-FCN	91.98%	99.06%	95.39%	94.19%	93.82%	93.97%	89.74%
Ours-FPN	82.97%	98.49%	90.07%	94.49%	93.05%	93.68%	85.37%
Ours-UNet	87.67%	99.28%	93.12%	94.40%	93.33%	93.81%	87.68%
Ours-IncepV3	95.00%	98.49%	96.71%	94.09%	93.89%	93.96%	90.88%
Ours-Res101	89.45%	98.85%	93.92%	93.99%	93.92%	93.93%	88.64%
Ours-Res152	94.02%	98.35%	96.13%	93.95%	94.16%	94.03%	90.63%
Ours-Res200	89.91%	98.06%	93.81%	94.16%	93.74%	93.92%	88.86%
Ours-Res50	94.46%	100%	98.20%	97.95%	97.67%	97.75%	93.66%

TABLE VIII

CLASSIFICATION AND SEGMENTATION OF OUR NETWORK ON VARIOUS DIFFICULTY LEVELS IN OUR DATASET

Diff. level	Image-level Classification			Pixel-level Segmentation			
	Pr.	Re.	F1	Pr.	Re.	F1	mIoU
Easy	98.97%	100%	99.48%	98.31%	97.87%	98.09%	94.21%
Medium	96.57%	100%	98.26%	97.95%	97.67%	97.81%	92.32%
Hard	93.73%	100%	96.76%	97.78%	97.59%	97.78%	89.47%

For both classification and segmentation, the general performance slightly decreases along with the increase in sample difficulty. For example, the mIoU and F1 score of segmentation only decreases 4.74% and 0.41% from easy to hard level, respectively, which proves the strong ability of our network. Generally, LRR achieves 86.59% F1 score for classification, 81.21% mIoU and 82.35% F1 score for segmentation, which are good enough for many applications. Meanwhile, deep learning method achieves better performance due to more distinguishable features learned by the well-designed network. We find that despite satisfying results of LRR, it fails in some hard samples, resulting in a lower mIoU and F1 score than the proposed network.

c) Analysis on efficiency: We evaluate the computational efficiency of each method (a detailed description of test datasets can be found in Table III), as summarized in Table IX. All test methods are in original resolution as Table III shows. For the proposed network, we divide each test sample into blocks with 512 × 512 pixels and sum the processing time of all blocks to calculate the FPS of this sample. The test computer is equipped with an Intel i9-9900k CPU and NVIDIA Titan RTX GPU. To ensure the accuracy of the evaluation, we run each method three times and calculate the mean time. It can be seen that both M-CI and M-CF are very slow because they only use CPU for computation and without any acceleration techniques. In contrast, M-FPN is much faster than them due to the GPU implementation based on the Caffe framework. Despite only using CPU in our LRR, we optimize the algorithm as much as possible with approximation and parallel computation, resulting in a comparable or even better performance compared with GPU methods. Furthermore, our network uses GPU and parallel computation, which further releases the computation burden with highest efficiency. Moreover, we calculate the cost time $Time_{1M}$ for processing one million pixels (1MP) of compared methods, as summarized in Table IX.

TABLE IX

COMPARISON OF COMPUTATIONAL EFFICIENCY (UNIT: IMAGE/SECOND). BEST RESULTS ARE MARKED IN BOLD

Datasets	M-CI	M-CF	M-FPN	M-LRR	M-Ours
D-CT (800 × 600)	0.191	1.684	2.653	6.536	**10.417**
D-CI (2048 × 1536)	0.201	0.036	0.224	1.300	**3.472**
D-CF (480 × 320)	1.456	2.237	7.519	12.048	**28.571**
D-ALT (991 × 462)	0.407	1.462	3.861	4.505	**6.173**
D-GAP2 (1920 × 1080)	0.191	0.049	0.286	1.633	**3.205**
D-CK500 (2560 × 1440)	0.014	0.015	0.312	1.065	**2.564**
D-PEARL (2048 × 2048)	0.133	0.205	0.225	0.871	**1.524**
$Mean_{eff}$ ↑	0.370	0.812	2.154	3.994	**7.990**
$Time_{1M}$ ↓	6.518	6.282	1.036	0.345	**0.184**

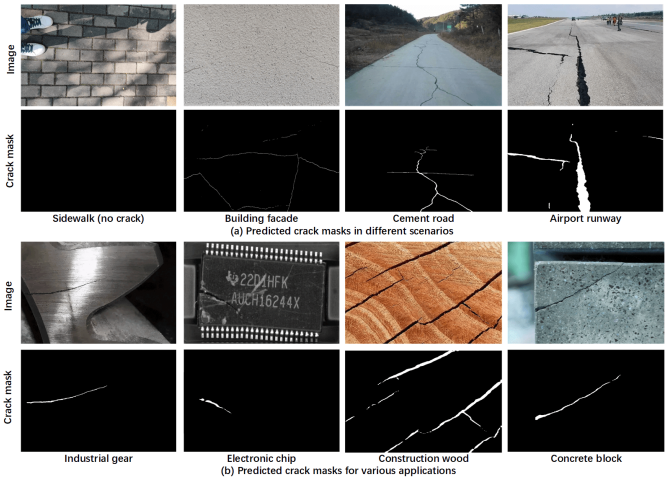

Fig. 10. Predicted crack masks of our DCNN in different scenarios and applications that varies significantly in color and appearance.

V. CONCLUSION AND PERSPECTIVES

In this paper, we propose a novel two-stage pavement crack detection and automatic sample generation method leveraging LRR and DCNN. In the first stage, we exploit latent clues that are valuable for crack detection in images with LRR. We get positive and negative crack samples with pixel-wise annotation after straightforward post-processing. In the second stage, we propose a DCNN and train with generated samples. Compared with LRR, DCNN has a better performance in hard samples. Extensive experiments are conducted to evaluate the performance of our method. We achieve image-level classification with the 98.20% F1 score and pixel-level segmentation with the 93.66% mIoU and 97.75% F1 score on our dataset. Despite we only train the network on our dataset, the compelling results on other datasets demonstrate the generalizability and superiority of our DCNN. Besides, we test

our DCNN on images from various scenarios and fields, such as industrial and electronic manufacturing, forestry, as shown in Fig. 10. The great performance further prove the great potential of our network in many applications. Last, to our best knowledge, our automatically generated dataset PEARL is currently the largest pavement crack dataset with pixel-wise annotation and multiple properties, which will be published to boost the community for both research and engineering.

There are also some limitations of this work. Currently, the weighting parameter for low rank and sparse parts is crucial for LRR performance and is currently determined by Eq. 11. However, it may not achieve the best performance or even poor results in very complex scenes. Despite the loose and easily-calculated thresholds for LRR post-processing, we still aim to achieve a fully threshold-free LRR for crack detection with better automatic strategies. Consequently, following directions exist after this work. First, we will corporate with engineering stakeholders closely to improve the robustness of procedures in the framework toward a full-automatic and operational level in practice. Second, the performance and efficiency of LRR depend on the weighting parameter and batch size to a certain extent. Determining how to select the optimal configuration for pavement crack detection accurately and automatically is an open problem. Third, despite cracks taking a majority of pavement distress, there are still other types such as potholes, applied patches. Determining how to achieve accurate detection and automatic sample generation of multi-type distress based on the proposed method is another valuable direction, which is important for autonomous driving and other fields.

ACKNOWLEDGMENT

The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University, and supported by Wuhan University - Huawei Geoinformatics Innovation Laboratory.

REFERENCES

[1] Y. Yu, J. Li, H. Guan, and C. Wang, "3D crack skeleton extraction from mobile LiDAR point clouds," in *Proc. IEEE Geosci. Remote Sens. Symp.*, Jul. 2014, pp. 914–917.

[2] Y. Yu, H. Guan, and Z. Ji, "Automated detection of urban road manhole covers using mobile laser scanning data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3258–3269, Dec. 2015.

[3] T. Yamada, T. Ito, and A. Ohya, "Detection of road surface damage using mobile robot equipped with 2D laser scanner," in *Proc. IEEE/SICE Int. Symp. Syst. Integr.*, Dec. 2013, pp. 250–256.

[4] L. Bursanescu and F. Blais, "Automated pavement distress data collection and analysis: A 3-D approach," in *Proc. Int. Conf. Recent Adv. 3-D Digit. Imag. Modeling*, 1997, pp. 311–317.

[5] J. Laurent, M. Talbot, and M. Doucet, "Road surface inspection using laser scanners adapted for the high precision 3D measurements of large flat surfaces," in *Proc. Int. Conf. Recent Adv. 3-D Digit. Imag. Modeling*, 1997, pp. 303–310.

[6] J. Laurent, D. Lefebvre, and E. Samson, "Development of a new 3D transverse laser profiling system for the automatic measurement of road cracks," in *Proc. Symp. Pavement Surf. Characteristics*, Portoroz, Slovenia, 2008, pp. 1–6.

[7] M. Monti, "Large-area laser scanner with holographic detector optics for real-time recognition of cracks in road surfaces," *Opt. Eng.*, vol. 34, no. 7, pp. 2017–2023, 1995.

[8] K. Wang and W. Gong, "Automated pavement distress survey: A review and a new direction," in *Proc. Pavement Eval. Conf.*, 2002, pp. 21–25.

[9] Z. Hou, K. C. P. Wang, and W. Gong, "Experimentation of 3D pavement imaging through stereovision," in *Proc. Int. Conf. Transp. Eng.*, Jul. 2007, pp. 376–381.

[10] K. C. P. Wang and W. Gong, "Automated real-time pavement crack detection and classification," Univ. Arkansas, Fayetteville, AR, USA, CHRPIDEA Program Project Final Rep., 2007.

[11] F. M. Nejad and H. Zakeri, "An optimum feature extraction method based on Wavelet–Radon transform and dynamic neural network for pavement distress classification," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 9442–9460, Aug. 2011.

[12] M. Quintana, J. Torres, and J. M. Menéndez, "A simplified computer vision system for road surface inspection and maintenance," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 608–619, Mar. 2016.

[13] Q. Li and X. Liu, "Novel approach to pavement image segmentation based on neighboring difference histogram method," in *Proc. Congr. Image Signal Process.*, 2008, pp. 792–796.

[14] H. Oliveira and P. L. Correia, "Automatic road crack segmentation using entropy and image dynamic thresholding," in *Proc. 17th Eur. Signal Process. Conf.*, Aug. 2009, pp. 622–626.

[15] L. Peng, W. Chao, L. Shuangmiao, and F. Baocai, "Research on crack detection method of airport runway based on twice-threshold segmentation," in *Proc. 5th Int. Conf. Instrum. Meas., Comput., Commun. Control (IMCCC)*, Sep. 2015, pp. 1716–1720.

[16] H. D. Cheng, J.-R. Chen, C. Glazier, and Y. G. Hu, "Novel approach to pavement cracking detection based on fuzzy set theory," *J. Comput. Civil Eng.*, vol. 13, no. 4, pp. 270–280, Oct. 1999.

[17] J. Tang and Y. Gu, "Automatic crack detection and segmentation using a hybrid algorithm for road distress analysis," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2013, pp. 3026–3030.

[18] M. Salman, S. Mathavan, K. Kamal, and M. Rahman, "Pavement crack detection using the Gabor filter," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 2039–2044.

[19] R. Medina, J. Llamas, E. Zalama, and J. Gómez-García-Bermejo, "Enhanced automatic detection of road surface cracks by combining 2D/3D image processing techniques," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 778–782.

[20] R. Kapela et al., "Asphalt surfaced pavement cracks detection based on histograms of oriented gradients," in *Proc. 22nd Int. Conf. Mixed Design Integr. Circuits Syst. (MIXDES)*, Jun. 2015, pp. 579–584.

[21] J. Zhou, "Wavelet-based pavement distress detection and evaluation," *Opt. Eng.*, vol. 45, no. 2, Feb. 2006, Art. no. 027007.

[22] P. Subirats, J. Dumoulin, V. Legeay, and D. Barba, "Automation of pavement surface crack detection using the continuous wavelet transform," in *Proc. Int. Conf. Image Process.*, Oct. 2006, pp. 3037–3040.

[23] K. Y. Song, M. Petrou, and J. Kittler, "Texture crack detection," *Mach. Vis. Appl.*, vol. 8, no. 1, pp. 63–75, Jan. 1995.

[24] S. Chanda et al., "Automatic bridge crack detection—A texture analysis-based approach," in *Proc. IAPR Workshop Artif. Neural Netw. Pattern Recognit.* Cham, Switzerland: Springer, 2014, pp. 193–203.

[25] W. Xu, Z. Tang, J. Zhou, and J. Ding, "Pavement crack detection based on saliency and statistical features," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 4093–4097.

[26] M. Avila, S. Begot, F. Duculty, and T. S. Nguyen, "2D image based road pavement crack detection by calculating minimal paths and dynamic programming," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 783–787.

[27] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, "Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2718–2729, Oct. 2016.

[28] 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, Feb. 2012.

[29] V. Kaul, A. Yezzi, and Y. Tsai, "Detecting curves with unknown endpoints and arbitrary topology using minimal paths," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 1952–1965, Oct. 2012.

[30] A. Chatterjee and Y. Tsai, "A fast and accurate automated pavement crack detection algorithm," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 2140–2144.

[31] Y.-C. Tsai, C. Jiang, and Y. Huang, "Multiscale crack fundamental element model for real-world pavement crack classification," *J. Comput. Civil Eng.*, vol. 28, no. 4, Jul. 2014, Art. no. 04014012.

[32] Y. J. Tsai, C. Jiang, and Z. Wang, "Implementation of automatic crack evaluation using crack fundamental element," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 773–777.

[33] Y. J. Tsai, A. Chatterjee, and C. Jiang, "Challenges and lessons from the successful implementation of automated road condition surveys on a large highway system," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2017, pp. 2031–2035.

[34] S. Chambon, "Detection of points of interest for geodesic contours: Application on road images for crack detection," in *Proc. Int. Joint Conf. Comput. Vis. Theory Appl.*, 2011, pp. 1–10.

[35] A. Cord and S. Chambon, "Automatic road defect detection by textural pattern recognition based on AdaBoost," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 27, no. 4, pp. 244–259, Apr. 2012.

[36] H. Oliveira and P. L. Correia, "CrackIT—An image processing toolbox for crack detection and characterization," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 798–802.

[37] 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, Dec. 2016.

[38] H. Oliveira and P. L. Correia, "Supervised strategies for cracks detection in images of road pavement flexible surfaces," in *Proc. 16th Eur. Signal Process. Conf.*, Aug. 2008, pp. 1–5.

[39] T. S. Nguyen, M. Avila, and S. Begot, "Automatic detection and classification of defect on road pavement using anisotropy measure," in *Proc. 17th Eur. Signal Process. Conf.*, Aug. 2009, pp. 617–621.

[40] C. V. Prabakar and C. K. Nagarajan, "A novel approach of surface crack detection using super pixel segmentation," *Mater. Today, Proc.*, vol. 42, pp. 1043–1049, Jan. 2021.

[41] J. J. Steckenrider and T. Furukawa, "A probabilistic superpixel-based method for road crack network detection," in *Proc. Sci. Inf. Conf.* Cham, Switzerland: Springer, 2019, pp. 303–316.

[42] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3708–3712.

[43] M. Eisenbach et al., "How to get pavement distress detection ready for deep learning? A systematic approach," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2039–2047.

[44] F.-C. Chen and M. R. Jahanshahi, "NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4392–4400, May 2018.

[45] R. Stricker, M. Eisenbach, M. Sesselmann, K. Debes, and H. Gross, "Improving visual road condition assessment by extensive experiments on the extended GAPs dataset," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[46] 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, Apr. 2020.

[47] V. Mandal, L. Uong, and Y. Adu-Gyamfi, "Automated road crack detection using deep convolutional neural networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 5212–5215.

[48] Y. Li, Z. Han, H. Xu, L. Liu, X. Li, and K. Zhang, "YOLOv3-lite: A lightweight crack detection network for aircraft structure based on depthwise separable convolutions," *Appl. Sci.*, vol. 9, no. 18, p. 3781, Sep. 2019.

[49] G. X. Hu, B. L. Hu, Z. Yang, L. Huang, and P. Li, "Pavement crack detection method based on deep learning models," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–13, May 2021.

[50] R. Roberts, F. Menant, G. Di Mino, and V. Baltazart, "Optimization and sensitivity analysis of existing deep learning models for pavement surface monitoring using low-quality images," *Autom. Construct.*, vol. 140, Aug. 2022, Art. no. 104332.

[51] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[52] 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, Mar. 2019.

[53] W. Choi and Y. Cha, "SDDNet: Real-time crack segmentation," *IEEE Trans. Ind. Electron.*, vol. 67, no. 9, pp. 8016–8025, Sep. 2020.

[54] Z. Fan et al., "Automatic crack detection on road pavements using encoder–decoder architecture," *Materials*, vol. 13, no. 13, p. 2960, Jul. 2020.

[55] D. H. Kang and Y.-J. Cha, "Efficient attention-based deep encoder and decoder for automatic crack segmentation," *Struct. Health Monitor.*, vol. 21, no. 5, pp. 2190–2205, Sep. 2022.

[56] Y. Inoue and H. Nagayoshi, "Crack detection as a weakly-supervised problem: Towards achieving less annotation-intensive crack detectors," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 65–72.

[57] Y. Benkhoui, T. El Korchi, and L. Reinhold, "UAS-based crack detection using stereo cameras: A comparative study," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2019, pp. 1031–1035.

[58] X. Xu and H. Yang, "Intelligent crack extraction and analysis for tunnel structures with terrestrial laser scanning measurement," *Adv. Mech. Eng.*, vol. 11, no. 9, pp. 1–7, 2019.

[59] H. Guan, J. Li, Y. Yu, M. Chapman, and C. Wang, "Automated road information extraction from mobile laser scanning data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 194–205, Feb. 2015.

[60] A. Zhang et al., "Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 10, pp. 805–819, Oct. 2017.

[61] Y. Fei et al., "Pixel-level cracking detection on 3D asphalt pavement images through deep-learning-based CrackNet-V," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 273–284, Jan. 2020.

[62] H. Feng et al., "GCN-based pavement crack detection using mobile LiDAR point clouds," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 1–10, Aug. 2022.

[63] H. Oliveira and P. L. Correia, "Automatic road crack detection and characterization," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 155–168, Mar. 2013.

[64] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk, "Salient region detection and segmentation," in *Proc. Int. Conf. Comput. Vis. Syst.* Cham, Switzerland: Springer, 2008, pp. 66–75.

[65] L. Ruff et al., "Deep semi-supervised anomaly detection," 2019, *arXiv:1906.02694*.

[66] J. Yu, D. Y. Kim, Y. Lee, and M. Jeon, "Unsupervised pixel-level road defect detection via adversarial image-to-frequency transform," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1708–1713.

[67] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 9, pp. 731–747, Sep. 2018.

[68] D. Kang, S. S. Benipal, D. L. Gopal, and Y.-J. Cha, "Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning," *Autom. Construct.*, vol. 118, Oct. 2020, Art. no. 103291.

[69] R. Ali and Y.-J. Cha, "Attention-based generative adversarial network with internal damage segmentation using thermography," *Autom. Construct.*, vol. 141, Sep. 2022, Art. no. 104412.

[70] K. Liu and B. M. Chen, "Industrial UAV-based unsupervised domain adaptive crack recognitions: From database towards real-site infrastructural inspections," *IEEE Trans. Ind. Electron.*, vol. 70, no. 9, pp. 9410–9420, Sep. 2023.

[71] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 1, pp. 1–37, 2009.

[72] Z. Gao, L.-F. Cheong, and Y.-X. Wang, "Block-sparse RPCA for salient motion detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 1975–1987, Oct. 2014.

[73] Z. Gao et al., "Synergizing appearance and motion with low rank representation for vehicle counting and traffic flow analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2675–2685, Aug. 2018.

[74] Z. Lin, M. Chen, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," 2010, *arXiv:1009.5055*.

[75] J. Wright and Y. Ma, *High-Dimensional Data Analysis With Lowdimensional Models: Principles, Computation, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2022.

[76] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[77] M. Burri et al., "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, Sep. 2016.

[78] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with Atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 801–818.

[79] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[80] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.
[81] J. Burkardt. *The Truncated Singular Value Decomposition*. Accessed: May 20, 2022. [Online]. Available: https://people.math.sc.edu/Burkardt/cpp_src/svd_truncated/svd_truncated.html
[82] L. Ramana, W. Choi, and Y.-J. Cha, "Fully automated vision-based loosened bolt detection using the Viola–Jones algorithm," *Struct. Health Monitor.*, vol. 18, no. 2, pp. 422–434, Mar. 2019.
[83] Y.-J. Cha, K. You, and W. Choi, "Vision-based detection of loosened bolts using the Hough transform and support vector machines," *Autom. Construct.*, vol. 71, pp. 181–188, Nov. 2016.

Zhi Gao received the B.E. and Ph.D. degrees from Wuhan University, Wuhan, China, in 2002 and 2007, respectively. In 2008, he joined the Interactive and Digital Media Institute, National University of Singapore (NUS), as a Research Fellow and the Project Manager. In 2014, he joined the Temasek Laboratories, NUS (TL@NUS), as a Research Scientist and the Principal Investigator. He is currently a Full Professor with the School of Remote Sensing and Information Engineering, Wuhan University. He has published more than 90 academic articles, which have been published in *International Journal of Computer Vision*, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *ISPRS Journal of Photogrammetry and Remote Sensing*, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEM, and other top journals. He received the prestigious "National Plan for Young Talents" Award and the Hubei Province Funds for Distinguished Young Scientists. In addition, he is a "Chutian Scholar" Distinguished Professor in Hubei. He serves as an Associate Editor for the *journal Unmanned Systems*.

Xuhui Zhao (Graduate Student Member, IEEE) received the master's degree from State the Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, China, where he is currently pursuing the Ph.D. degree with the School of Remote Sensing and Information Engineering. His research interests include low rank analysis, satellite video understanding, visual SLAM, and robotics.

Min Cao received the bachelor's degree from Wuhan University, Wuhan, China. He is currently a Professor with the Hubei University of Technology. He is the CEO of Wuhan Guanggu Zoyon Science and Technology Company Ltd. He is a Senior Engineer of photogrammetry and remote sensing. His research interest include electric power automation, precision engineering survey, highway inspection, and intelligent transportation system.

Ziyao Li received the master's degree from Wuhan University, Wuhan, China, where she is currently pursuing the Ph.D. degree with the School of Remote Sensing and Information Engineering. Her research interests include deep learning, computer vision, and applications of deep learning techniques in remote sensing images.

Kangcheng Liu (Member, IEEE) received the B.Eng. degree from the Harbin Institute of Technology (HIT) and the Ph.D. degree in mechanical and automation engineering from The Chinese University of Hong Kong (CUHK).

He is currently a Senior Robotics Engineer and a Research Fellow with Nanyang Technological University (NTU), Singapore. He has more than ten first-author publications and serves as a Reviewer as well as a Program Committee Member in prestigious conference and journals such as ICRA, ECCV, ACM Multimedia, IROS, and several IEEE/ASME TRANSACTIONS. His research interests include robotics, robot control, LIDAR-SLAM, machine learning and computer graphics for perception, and 3D Vision. He is the Program Session Chair of IEEE ICRA and IROS.

Ben M. Chen (Fellow, IEEE) received the B.Sc. degree in mathematics and computer science from Xiamen University, China, in 1983, the M.Sc. degree in electrical engineering from Gonzaga University, USA, in 1988, and the Ph.D. degree in electrical and computer engineering from Washington State University, USA, in 1991.

He was the Provost's Chair Professor with the Department of Electrical and Computer Engineering, National University of Singapore, before joining The Chinese University of Hong Kong (CUHK) in 2018. He was an Assistant Professor with the Department of Electrical Engineering, the State University of New York at Stony Brook, USA, from 1992 to 1993. He is currently a Professor of mechanical and automation engineering with CUHK. He has authored/coauthored more than 500 journal and conference papers, and a dozen research monographs in control theory and applications, unmanned systems, and financial market modeling. His current research interests include unmanned systems and their applications. He is a Fellow of Academy of Engineering, Singapore. He served on the editorial boards for a dozen international journals, including *Automatica* and IEEE TRANSACTIONS ON AUTOMATIC CONTROL. He is serving as the Editor-in-Chief for *Unmanned Systems*, the Editor for *International Journal of Robust and Nonlinear Control*, and an Editorial Member of *Science China Information Sciences*.