Automatic Road Crack Detection Using Random Structured Forests

Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen

***Abstract*—Cracks are a growing threat to road conditions and have drawn much attention to the construction of intelligent transportation systems. However, as the key part of an intelligent transportation system, automatic road crack detection has been challenged because of the intense inhomogeneity along the cracks, the topology complexity of cracks, the inference of noises with similar texture to the cracks, and so on. In this paper, we propose CrackForest, a novel road crack detection framework based on random structured forests, to address these issues. Our contributions are shown as follows: 1) apply the integral channel features to redefine the tokens that constitute a crack and get better representation of the cracks with intensity inhomogeneity; 2) introduce random structured forests to generate a high-performance crack detector, which can identify arbitrarily complex cracks; and 3) propose a new crack descriptor to characterize cracks and discern them from noises effectively. In addition, our method is faster and easier to parallel. Experimental results prove the state-of-the-art detection precision of CrackForest compared with competing methods.**

***Index Terms*—Road crack detection, structured learning, machine learning, random structured forests, crack descriptor, crack characterization.**

I. Introduction

CRACK is a form of road distresses that may potentially reduce the road performance and threaten the road safety [1]. Governments have made a great effort to achieve the goal of constructing a high quality road network. They are now, more than ever, fully aware of the need for adequate road inspection and maintenance. Crack detection is an essential part of road maintenance systems and has attracted growing attentions in recent years. As it is known, traditional manual road crack detection approaches are very time-consuming, dangerous, labor-intensive and subjective [2]–[5]. Thus, the slow and subjective traditional procedures have been replaced gradually by automatic crack detection, which is developed for fast and reliable crack analysis in intelligent transportation systems [6]. Automated crack detection systems can quantify the quality of road surfaces and assist in prioritizing and planning the maintenance of the road network and thereby accomplish the objective of preserving the roads in good condition and extending the service life.

With the development of image processing techniques, road crack detection and recognition have been widely discussed in the past few decades [7]–[11]. In early methods [12], [13], researchers usually use threshold-based approaches to find crack regions based on the assumption that real crack pixel is consistently darker than its surroundings. These methods are very sensitive to noises, since only brightness feature is taken into consideration. Moreover, these approaches are performed on individual pixels. Lack of global view also makes these methods unsatisfying.

In terms of the current methods [5], [8], [9], [11], [14], [15], most researchers try to suppress the inference of noises by incorporating features such as gray-level value [11], the mean and the standard deviation value [5], [6], [16]. In addition, to improve the continuity of the existing methods, researchers attempt to conduct crack detection from a global view by introducing methods such as Minimal Path Selection (MPS) [17]–[19], Minimum Spanning Tree (MST) [20], [21], Crack Fundamental Element (CFE) [22], [23] and so on. These methods can partly eliminate noises and enhance the continuity of detected cracks.

However, these methods do not perform well while dealing with cracks with intensity inhomogeneity or complex topology. A possible explanation is that the used features only roughly capture the gray-level information but some unique characteristics of crack may not be presented and utilized properly. Besides, local structured information is ignored by existing methods. In fact, cracks in a local image patch are highly interdependent, which often contain well-known patterns, such as longitudinal, transverse, diagonal and so on. Therefore, structured learning is proposed to solve similar problems in recent years. For example, in [24], researchers apply structured learning to semantic image labeling where image labels are also interdependent.

In order to overcome these two shortages mentioned above, we propose a novel road crack detection method (called CrackForest) based on random structured forests, which is

Manuscript received October 17, 2015; revised February 10, 2016; accepted April 4, 2016. Date of publication May 18, 2016; date of current version November 23, 2016. This work was supported by the National Natural Science Foundation of China under Grants 91546201, 71331005, 71110107026, and 61402429. The Associate Editor for this paper was A. Eskandarian. *(Corresponding author: Zhiquan Qi.)*

Y. Shi is with the School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100049, China; with Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China; and also with University of Nebraska Omaha, Omaha, NE 68106 USA (e-mail: yshi@ucas.ac.cn).

L. Cui is with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 101408, China (e-mail: lmcui932@163.com).

Z. Qi is with the Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China (e-mail: qizhiquan@foxmail.com).

F. Meng and Z. Chen is with the School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: im_feixie@163.com; wxzmczs@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2016.2552248

superior to other state-of-the-art detecting techniques like CrackTree [20], CrackIT [6], FFA [25] and MPS [17], [19]. CrackForest incorporates complementary features from multiple levels to characterize cracks and to take advantage of the structured information in crack patches. In specific, we first extend the traditional road crack detection feature set by introducing the integral channel features [26] to re-define crack tokens with structured information. After that, we apply random structured forests [27] to exploit such structured information. Random structured forests predict a patch crack of structured tokens that are aggregated across the image to compute our preliminary crack detection result. In this step, the structured tokens assigned to each image patch can be obtained simultaneously. Then, the structured tokens are used to construct the crack descriptor which consists of two statistical histograms to characterize cracks with arbitrary topology. With the crack descriptor, a classification method is applied to discriminate the cracks from noises. In addition, we also propose a quantitative evaluation method for road crack detection task. Extensive experiments demonstrate the efficiency of CrackForest on real road crack dataset and our method shows state-of-the-art precision.

II. Related Work

In this section, we first give a brief review of crack detection, after that, the related crack characterization methods are discussed. Crack characterization exploits the spatial distribution of image tokens composing the detected cracks and thereby transforms the structured tokens into discrete labels.

A. Crack Detection

Numerous papers have been written on road crack detection over the past 30 years. Early works [1], [28]–[30] are mainly based on intensity-thresholding for its simplicity and efficiency. Most recent work explores crack detection under more challenging conditions and can be divided into five branches: methods based on saliency detection, textured-analysis, wavelet transform, minimal path and machine learning. An assessment of various pavement distress detection methods can be found in [31] and [32].

Salient Detection: Salient regions are visually more conspicuous due to their contrast with the surroundings. Although existing methods [33], [34] demonstrate their effectiveness in detecting salient regions in the Berkeley database [35], they perform poor on the completeness and continuity of detected crack.

Textured-Analysis: Since road surface images are often highly textured, textured-analysis methods [8], [36], [37] are introduced in road crack detection. In order to distinguish the cracks and the backgrounds, [8], [36] use the Wigner model, and [37] uses classification method. These methods use a local binary pattern operator to determine whether each pixel belongs to a crack and the local neighbor information is not taken into consideration. Therefore, the cracks with intensity inhomogeneity can not be detected precisely.

Wavelet Transform: Wavelet transform is applied to separate distresses from noises [38]. In [4], complex coefficient maps are built by a 2D continuous wavelet transform, wavelet coefficients maximal values are obtained for crack detection. As a result, differences between crack regions and crack free regions could be raised up. However, due to the anisotropic characteristic of wavelets, these approaches may not handle the cracks with low continuity or high curvature properly.

Minimal Path Selection: Give both endpoints of the curve as user's input, minimal path based method can extract simple open curves in images, that is first proposed by Kass *et al.* [39]. In [40], Kaul *et al.* propose a method that is able to detect the same types of contour-like image structures with less prior knowledge about both the topology and the endpoints of the desired curves. To avoid false detections that are assimilating loops, Amhaz *et al.* [17], [19] propose an improved algorithm to select endpoints at the local scale and then to select minimal paths at the global scale. It can also detect the width of the crack. In [25], Nguyen *et al.* propose a method which takes into account intensity and crack form features for crack detection simultaneously by introducing Free-Form Anisotropy.

Machine Learning: With the increasing size of image data, machine learning based methods [3], [5], [15], [41]–[43] have become an important branch in detecting road cracks. In [3], artificial neural network models are used to separate crack pixels from the background by selecting proper thresholds. [41] deals with the detection of poorly contrasted cracks in textured areas using a Markov random field model. In [43], Cord *et al.* use AdaBoost to distinguish images of road surfaces with defects from road surfaces based on textual information with patterns. For all these methods, the training and classification are conducted on each sub-image and as local method, they have drawbacks in finding complete crack curves over the whole image.

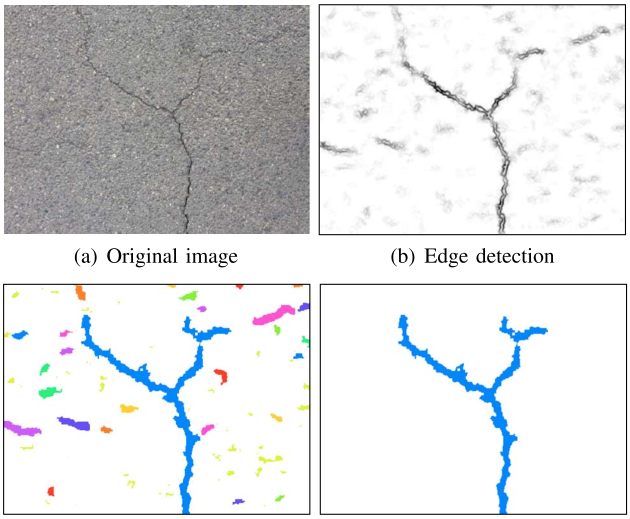
B. Crack Characterization

Existing methods on crack characterization are mainly based on shape descriptor, crack seeds and assigning crack type on each image block.

Reference [14] gives the definition of cracks based on mathematical morphology and proposes that a crack is thought to be a succession of saddle points with linear features. But this definition is pretty vague. Reference [2], [44] use the direction indices of each pixels and extensible directions for each direction to characterize cracks. A chromosome representation is applied to encode the different ensemble of directions and its extensible directions. Therefore, a crack can be represented as a long sequence of 0 and 1.

Reference [31], [42] categorize the cracks into five types: longitudinal, transverse, diagonal, block, and alligator. Reference [42] uses a neural network based method to search patterns of various crack types horizontally and vertically. Reference [31] uses curves and buffers to describe certain regions of a crack. Reference [9] uses longitudinal, transverse, or diagonal crack seeds to identify longitudinal and transverse cracks. Orientation and strength information are taken into consideration by [20], which largely improves the diversity of crack seeds.

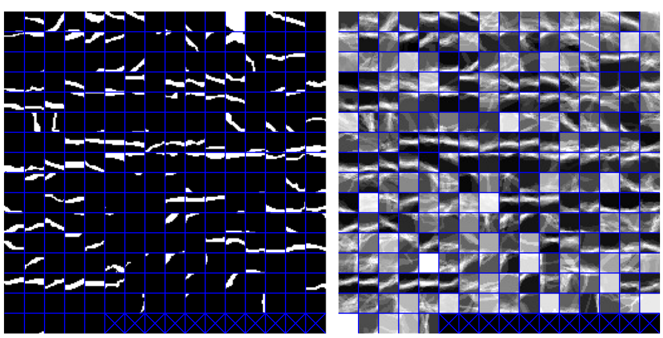
In [6], cracks are classified into three types as defined by the Portuguese Distress Catalog. They use two block feature



(a) Original image (b) Edge detection

(c) Binarization (d) Crack detection

Fig. 1. Consider the pavement surface shown in (a). (b) Preliminary detection results after applying random structured forests. Darker color indicates that the pixel is more likely to contain a crack. After eroding and dilating, the result is shown in (c). (d) Final result after the classification stage.



(a) (b)

Fig. 2. Examples of tokens learned from a manually labeled image database. (a) Most representative token for each token set. (b) Mean contour structure for each token set.

including the mean and the standard deviation values of pixel-normalized intensities to categorize an image block as longitudinal, transversal or miscellaneous. Reference [5] computes CTA (Conditional Texture Anisotropy) values over the distribution of the mean and the standard deviation values calculated on pixels to distinguish crack pixels from defect free pixels.

However, there are two main drawbacks in these methods. On the one hand, new types of crack cannot be generated. By applying the structured tokens, we extend the crack types into thousands of dimensions. On the other hand, these methods perform poor on the cracks with complex topology. To address this issue, we propose a novel crack descriptor to describe the cracks with arbitrary complex topology.

III. Automatic Road Crack Detection

In this section, we will introduce our novel crack detection method which can take advantage of the structured information of cracks. Fig. 3 shows the overall procedure of our proposed method. This framework can be divided into three parts: In the first part, we extend the feature set of traditional crack

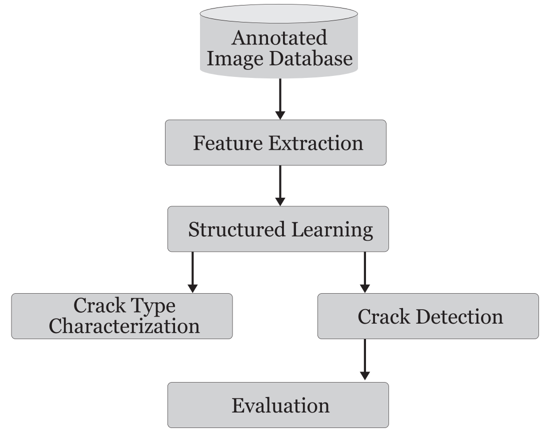


Fig. 3. Procedure of the proposed automatic road crack detection method.

detection method by introducing the integral channel features. These features extracted from multiple levels and orientations allow us to re-define representative crack tokens with richer structured information. In the second part, random structured forests are introduced to exploit such structured information, and thereby a preliminary result of crack detection can be obtained. In the third part, we propose a new crack descriptor by using the statistical character of tokens. This descriptor can characterize the cracks with arbitrary topology. And a classification algorithm (KNN, SVM or One-Class SVM) is applied to discriminate cracks from noises effectively. Please find the results in every part of our method in Fig. 1.

A. *Structured Tokens*

Token (segmentation mask) indicates the crack regions of an image patch. Current block-based methods [6], [38] are usually used to extract small patches and calculate mean and standard deviation value on these patches to represent an image token. These traditional features are computed on gray level images and applied to describe the brightness and gradient information. However, local structured information is not taken into consideration. So in the first step, we re-define the tokens by introducing the integral channel features which incorporate the color, gradient information from multiple levels and facets.

1) Learning the Tokens: Assume that we have a set of images I with a corresponding set of binary images G representing the manually labeled crack edge from the sketches. We use a 16×16 sliding window to extract image patches $x \in \mathcal{X}$ from the original image. Image patch x which contains a labeled crack edge at its center pixel, will be regarded as positive instance and vice versa. $y \in \mathcal{Y}$ encodes the corresponding local image annotation (crack region or crack free region), which also indicates the local structured information of the original image. These tokens cover the diversity of various cracks, which are not limited to straight lines, corners, curves, etc.

From Fig. 4, we can see the extracted image patches and their hand drawn contour tokens. These image patches and tokens will be used to train CrackForest later.

2) Feature Extraction: To describe the above tokens, features are computed on the image patches x extracted from the

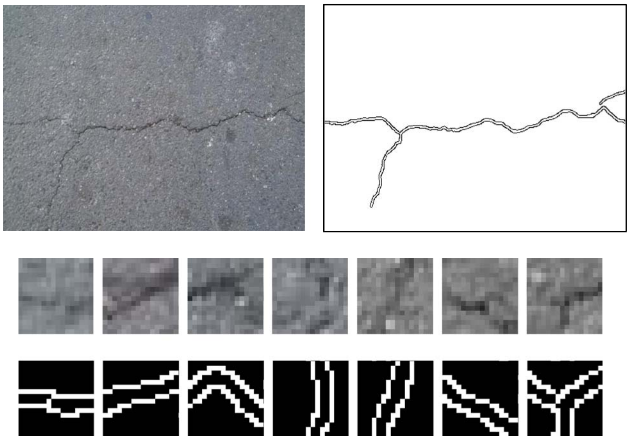


Fig. 4. (Top) Example of original image and its ground truth. (Bottom) Example of extracted image patches and their hand-drawn contours. Notice the variety of sketches.

training images I, and considered to be weak classifiers in the next step.

We use mean and standard deviation value as features. Two matrices are computed for each original image: the mean matrix M_m with each block's average intensity and the standard deviation matrix STD_m with corresponding standard deviation value std. Each image patch yields a mean value and a 16×16 standard deviation matrix.

To characterize the cracks more comprehensively, we also apply a set of channel features composed with color, gradient and oriented gradient information. Integral channel features not only perform better than other features including histogram of oriented gradient (HOG), but also achieve fast detecting results and integrate heterogeneous sources of information [26].

3 color, 2 magnitude and 8 orientation channels, for a total of 13 channels yield 3328 candidate features. Each of the channel captures a different aspect of information. Self-similarity features are compute for each channel. These features capture the portion that an image patch contains similar textures based on color or gradient information [45]. Texture information is computed on a $m \times m$ grid over the patch. These differences yield $\binom{5\cdot 5}{2} = 300$ more features per channel.

B. Structured Learning

In previous step, a set of tokens y which indicate the structured information of local patches, and features which describe such tokens, are acquired. In this step, we cluster these tokens by using a state-of-the-art structured learning framework, random structured forests, to generate an effective crack detector. Random structured forests can exploit the structured information and predict the segmentation mask (token) of a given image patch. Thereby we can obtain the preliminary result of crack detection.

In random structured forests, each decision tree $f_t(x)$ classifies an image patch $x \in \mathcal{X}$ by recursively branching left or right down to the tree until a leaf is reached. And the class of the node is assigned to patch x. The leaf stores the prediction of the input x, which is a target label $y \in \mathcal{Y}$ or a distribution over $\mathcal{Y}$. By training such a tree, tokens with the same structure will be gathered at one leaf. We use the most representative token in each leaf to represent the token class. The class number of tokens equals to the number of leaves.

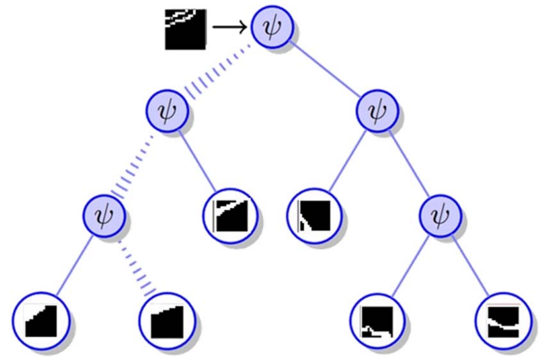


Fig. 5. Routing path of an image patch.

A forest T can be seen as an ensemble of decision trees f_t. Each tree $f_t(x)$ gives a prediction of a sample $x \in \mathcal{X}$. The final class prediction of multiple trees is integrated by a majority voting algorithm. A leaf $L(\pi) \in f_t$ can assign a class prediction for samples it is reached by, where π stands for the most represented token in the leaf. Each node $N(h, f_t^L, f_t^R) \in f_t$ is associated with a binary split function

$$h(x, \theta_j) \in \{0, 1\} \tag{1}$$

with feature θ_j for each node j. If $h(x, \theta_j) = 0$, sample x should be branched to the left sub-tree f_t^L, otherwise the right sub-tree f_t^R.

1) Class Prediction: Given a tree $f_t \in T$, the class prediction of an image patch $x \in \mathcal{X}$ can be obtained by recursively branching it forward until a leaf is reached. An intuitive example has shown in Fig. 5. The prediction function $\psi(x|f_t) : \mathcal{X} \to \mathcal{Y}$ for node j is

$$\psi\left(x|N\left(h, f_t^L, f_t^R\right)\right) = \begin{cases} \psi\left(x|f_t^L\right), & \text{for } h(x,\theta_j) = 0 \\ \psi\left(x|f_t^R\right), & \text{for } h(x,\theta_j) = 1 \end{cases}$$
$$\psi\left(x|L(\pi)\right) = \pi. \tag{2}$$

The final class prediction of x is obtained from the prediction of each tree as the one receiving the majority voting.

2) Randomized Training: Each tree is trained individually. For a given node N_j and training set $S_j \subset \mathcal{X} \times \mathcal{Y}$, the goal is to find the optimal feature θ_j that results in a good split of the data. In other words, the discrepancy of tokens in the same leaf should be as small as possible. We apply information gain to measure this discrepancy and maximize the information gain to choose θ_j. The form of information gain for node j is defined as follow:

$$I_j = I\left(S_j, S_j^L, S_j^R\right) \tag{3}$$

where $S_j = S_j^L \cup S_j^R$, $S_j^L = \{(x, y) \in S_j | h(x, \theta_j) = 0\}$ stands for a set of samples that reaches the left sub-tree of the current node and $S_j^R = \{(x, y) \in S_j | h(x, \theta_j) = 1\}$ refers to the other set of samples that reaches the right sub-tree.

Whether a terminal node should be further split depends on the maximum depth, the minimum size of node or the entropy of the class distribution. If the node is no longer splitting, a leaf is grown where the class prediction π is set to the most representative token in the training data. Otherwise a node $N(h, f_t^L, f_t^R)$ is grown where h is a split function regulated by parameter θ_j, maximizing the information gain about the label distribution due to the split $\{S_j^L, S_j^R\}$ of the training data S.

For multi-class classification ($\mathcal{Y} \subset \mathcal{Z}$), the definition of information gain is

$$I_j = H(S_j) - \sum_{k \in \{L,R\}} \frac{|S_j^k|}{|S_j|} H\left(S_j^k\right) \tag{4}$$

where $H(S_j) = -\sum_y p_y \log(p_y)$ denotes the Shannon entropy and p_y stands for the proportion of elements in S with label y. Alternatively, Gini impurity $H(S_j) = \sum_y p_y(1 - p_y)$ can also be applied in Equation (4).

Individual decision tree tends to overfit, which may negatively affected accuracy. To overcome this drawback, random structured forests combine multiple decision trees together to assign the final label. Random structured forests have shown promising flexibility and generalization ability, and most importantly, this method is easy to parallel and extremely fast.

The randomness is embodied by randomly subsampling the data used to train each tree and each node, and randomly subsampling the features used to split each node. In order to maintain the diversity of trees, only a small pool of features is used to select the optimal θ_j when choosing the split function.

3) Structured Mapping: Random structured forests change the discrete outer space of the traditional decision forests into a structured space $\mathcal{Y}$. While dealing with structured label $y \in \mathcal{Y}$ directly may cause significant computing expense, the structured labels $y \in \mathcal{Y}$ at a leaf is mapped into a set of discrete labels $c \in \mathcal{C}$, where $\mathcal{C} = \{1, \ldots, k\}$. Given the discrete label space $\mathcal{C}$, information gain can be calculated efficiently via (4). We first map the label space $\mathcal{Y}$ into a intermediate space $\mathcal{Z}$

$$\Pi : \mathcal{Y} \to \mathcal{Z}. \tag{5}$$

Define $z = \Pi(y)$ in space $\mathcal{Z}$ as a $\binom{16 \cdot 16}{2} = 32640$ dimensional vector, which encodes every pair of pixels in the segmentation mask y. The computational cost of z appears to be significant.

While the dimension of z is still very high, we randomly select 256 dimension of z to train each split function, using a distinct reduced mapping function at each node j

$$\Pi_\phi : \mathcal{Y} \to \mathcal{Z}. \tag{6}$$

Then we apply PCA reduction to map 256 dimensions of z into 5 dimensions, with the first dimension being the most significant factor. To obtain the discrete label $c \in \mathcal{C}$ of each structured label y, we use the first dimension of each intermediate label z to cluster into two sets. Labels in the same cluster are assigned to the same label c. With the label c, standard information gain can be calculated at each node.

After the random structured forests are trained, the structured labels y are gathered at the leaves of each tree (see Fig. 2). An image patch is routed though each tree based on the split function until a leaf is reached. The most representative token in the leaf is assigned to the image patch. Fig. 6 shows an intuitive example. We select the token which has the lowest variance with others as the most representative token.

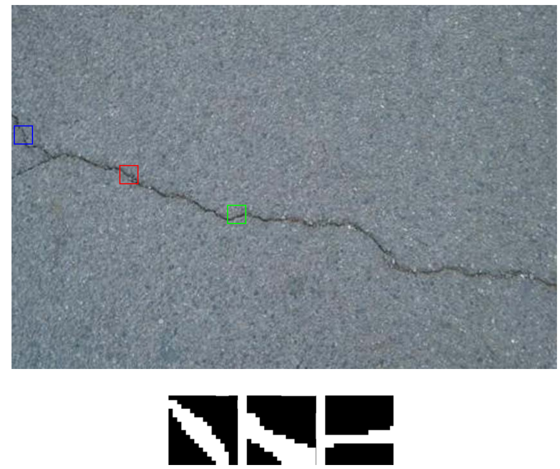
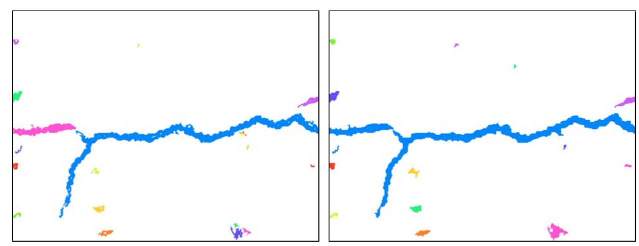


Fig. 6. Assigning y to each image patch. The image patches have been assigned to the tokens below (both from left to right).



(a) (b)

Fig. 7. (a) Binarization result based on threshold when $\alpha = 0.1$ (removing pixels of low probability according to the given probability map). (b) Result after erosion and dilation with a 4×4 rectangular structuring element.

4) Binarization: After the structured mapping, each image patch x is assigned to a structured label y. Due to the overlapping, the result of detection is a map, where each element indicates the probability that the corresponding position in the original image is on crack region. So we use a threshold α to obtain all the possible regions. A high α value may cause the incontinuity of cracks and the ignorance of inapparent cracks. Therefore, we choose $0.1 \leqslant \alpha \leqslant 0.2$ in this paper. Fig. 7(a) shows the binarization result when $\alpha = 0.1$.

We conduct the erosion and the dilation operation on the preliminary edge detection results to make the cracks as connective as possible. The inside of the crack is filled and the fragments are connected. Moreover, some of the noises are eliminated. From Fig. 7(b), we can see that small fragments of the detected region have merged together and the continuity of the crack has been improved.

C. Crack Type Characterization and Detection

Each image patch is assigned to a structured label y (segmentation mask) after structured learning. Although we obtain a preliminary result of crack detection so far, a lot of noises are generated due to the textured background at the same time. Traditional thresholding methods mark small regions as noises according to their sizes. However, in this way, many inconspicuous cracks may be removed by mistake.

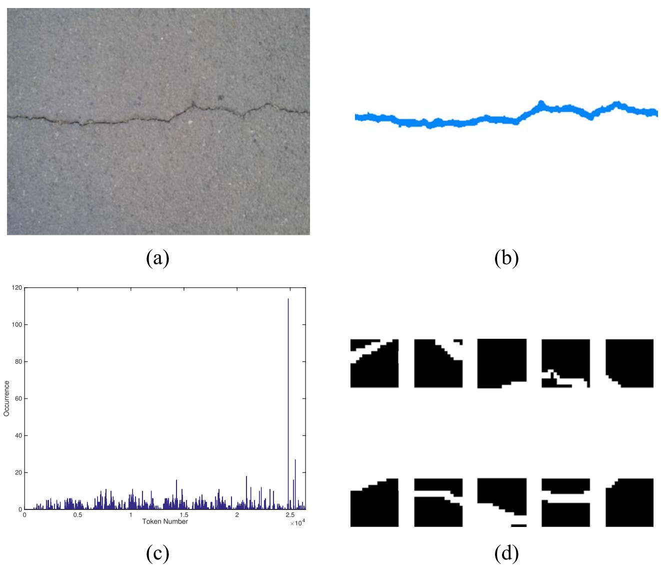


Fig. 8. (a) Original image. (b) One of the detected regions. (c) Statistical feature histogram of the detected region. (d) Appearance of the ten most frequent tokens look.

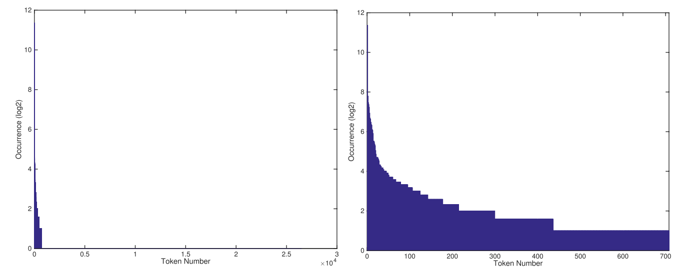
Cracks have a series of unique structural properties that differ from noises. Based on this thought, we propose a novel crack descriptor by using the statistical feature of structured tokens in this section. This descriptor consists of two statistical histograms, which can characterize cracks with arbitrary topology. By applying classification method like SVM, we can discriminate noises from cracks effectively.

1) Crack Descriptor: Existing crack characterization methods categorize cracks into several types, such as longitudinal, transverse, diagonal, block, and alligator. However, the descriptor proposed, which consists of hundreds of dimensions respectively, has greatly broadened the range of representable crack. What is more, the crack is no longer limited to a few types, we extend the types of crack into thousands of kinds.

We use 26443 structured tokens obtained in the structured learning procedure to characterize the cracks. The statistical histogram and the neighborhood histogram of these tokens within a crack can be calculated precisely.

Statistical Feature Histogram: After the structured learning procedure, we can obtain the token map. Each point in the map indicates the label of token that the 16×16 image patch around the corresponding position is assigned to. Statistical feature histogram in Fig. 8 reflects the composition of the crack comprehensively. Each dimension of this histogram represents the number of a certain token.

The token number from the training result is numerous. After plotting the overall occurrence of each token in Fig. 9(a), we notice a long tail effect of the token distribution. After analyzing the statistical information of appeared tokens, we find that over 90% occurrences of all the tokens are centered on 708 specific tokens. The occurrences of most tokens make up only a small percentage of all. Therefore, we only use these 708 tokens to construct the statistical feature histogram and the statistical neighborhood histogram. Fig. 9(b) shows the occurrence of these tokens.



(a) The occurrence of all the tokens in descending order

(b) The occurrence of the 708 most frequent tokens in descending order

Fig. 9. Statistical feature histogram showing the occurrence (in logs) of each token (sorting in descending order of occurrence). (a) Statistical feature of all 26443 tokens. (b) Only the most representative tokens are shown.

Statistical Neighborhood Histogram: The statistical neighborhood histogram captures the neighborhood information of two tokens. We calculate the co-occurrence of each pair of tokens only when they are adjacent. There would be $\binom{708}{2} = 250278$ token pairs without reduction. Furthermore, we also find the long tail effect of this distribution. Over 90% occurrences of all the token pairs are centered on 956 specific token pairs. Thus, only these token pairs will be used in the following section.

2) Crack Detection: With the two histograms for each separated region, we can characterized cracks with arbitrary topology. In this section, we will introduce how to discriminate the noises from cracks by using the two histograms.

Vectorization: The distribution of occurrence and co-occurrence are scaled to [0, 1]. Hence, each detected region is presented as a long vector with $708 + 956 = 1664$ dimensions.

Classification: We consider the crack detection procedure as a classification problem. The crack regions are assigned to class $+1$ and the crack free regions are assigned to class -1. By applying KNN(k-Nearest Neighbor), SVM (Support Vector Machine) with linear kernel and One-Class SVM with linear kernel, we obtain the classification model which can discriminate cracks from noises effectively. The results of our algorithm using SVM are shown in Fig. 10.

IV. Experiments

In this section, we analyze the performance of our proposed method. Part of the Matlab code is supported on Piotr's Computer Vision Toolbox [46] and Structured Edge Detection Toolbox [27]. All the experiments are conducted on a desktop with AMD FX(tm)-4300 Quad-Core Processor and 4G RAM.

In order to evaluate our method, we compare it with the traditional method (Canny [47]), and the state-of-the-art road detection methods (CrackTree [20], CrackIT [48], FFA [25] and MPS [17], [19]).

We show results on two datasets measuring accuracy performance. We demonstrate the cross dataset generalization of our approach by testing on each dataset using CrackForest learned on the other.

Unlike other edge detection tasks, the evaluation of crack detection performance is difficult. Thereby we define two kinds of evaluation indicators for crack detection.

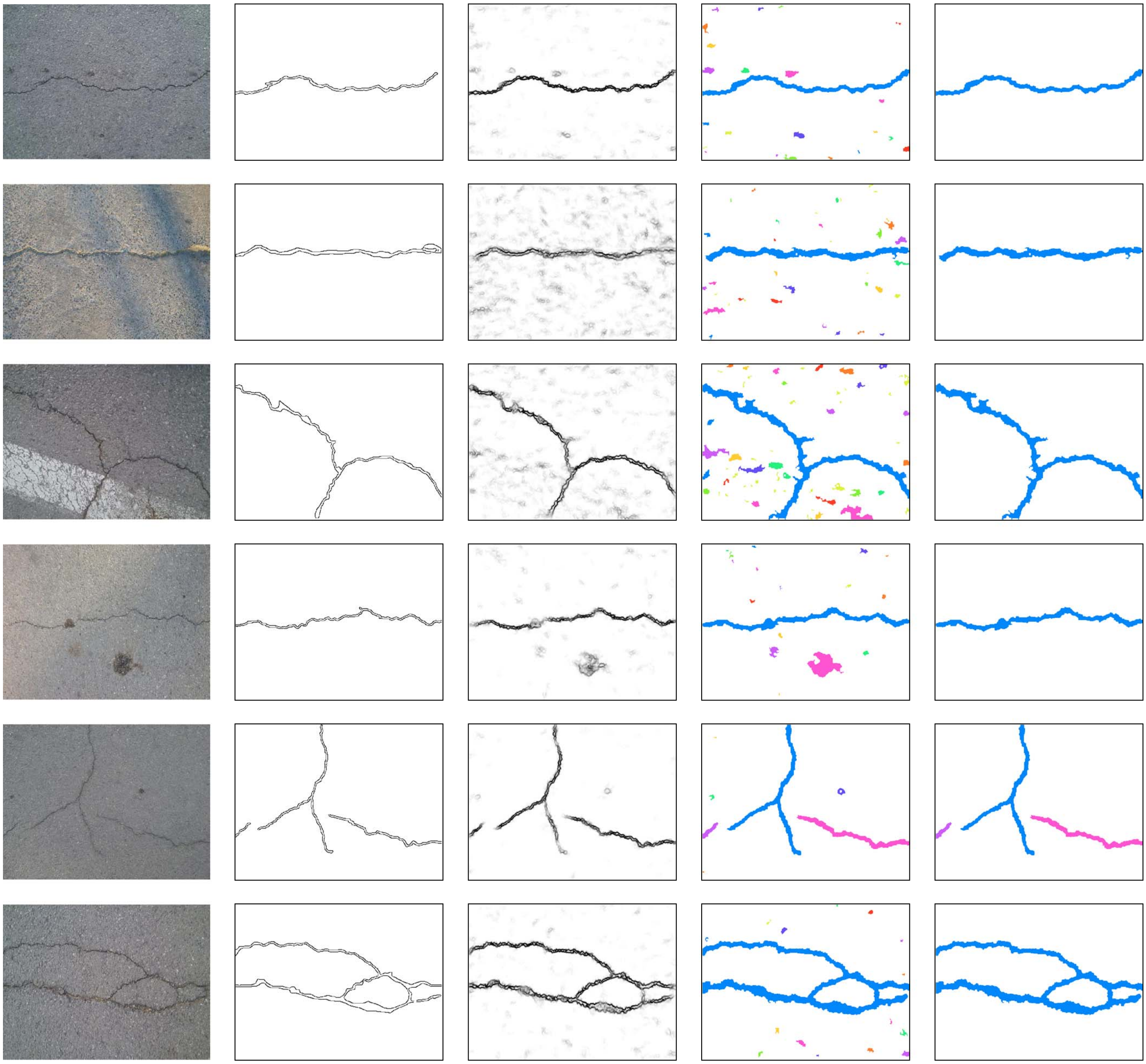


Fig. 10. Part of the results of road crack detection using our proposed method. Notice that our method can eliminate the influence of oil stains, shadows, and complex background, effectively, and can cope with miscellaneous crack topology.

Crack Detection Accuracy: We use precision, recall and F1 Score to evaluate the performance of different crack detection algorithms.

The precision and recall can be computed on true positive (TP), false negative (FN) and false positive (FP)

$$\mathrm{Pr}_{\mathrm{pixel}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{7}$$

$$\mathrm{Re}_{\mathrm{pixel}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{8}$$

$$F1_{\mathrm{pixel}} = \frac{2\times \mathrm{Pr}_{\mathrm{pixel}} \times \mathrm{Re}_{\mathrm{pixel}}}{\mathrm{Pr}_{\mathrm{pixel}} + \mathrm{Re}_{\mathrm{pixel}}}. \tag{9}$$

Assume that the detected pixels which are no more than five pixels away from the manually labeled pixel are true positive pixels.

The precision, recall and F1 Score on detect region can be similarly computed by (7) and (8)

$$\mathrm{Pr}_{\mathrm{region}} = \frac{\mathrm{TP}_r}{\mathrm{TP}_r+\mathrm{FP}_r} \tag{10}$$

$$\mathrm{Re}_{\mathrm{region}} = \frac{\mathrm{TP}_r}{\mathrm{TP}_r+\mathrm{FN}_r} \tag{11}$$

$$F1_{\mathrm{region}} = \frac{2\times \mathrm{Pr}_{\mathrm{region}} \times \mathrm{Re}_{\mathrm{region}}}{\mathrm{Pr}_{\mathrm{region}} + \mathrm{Re}_{\mathrm{region}}} \tag{12}$$

Crack Continuity Assessment: We define the "Continuity Index (CI)" as a degree of continuity. It measures how much the detected regions are connected if they belong to the same crack. Denote M as the number of images in the testing set. N_i

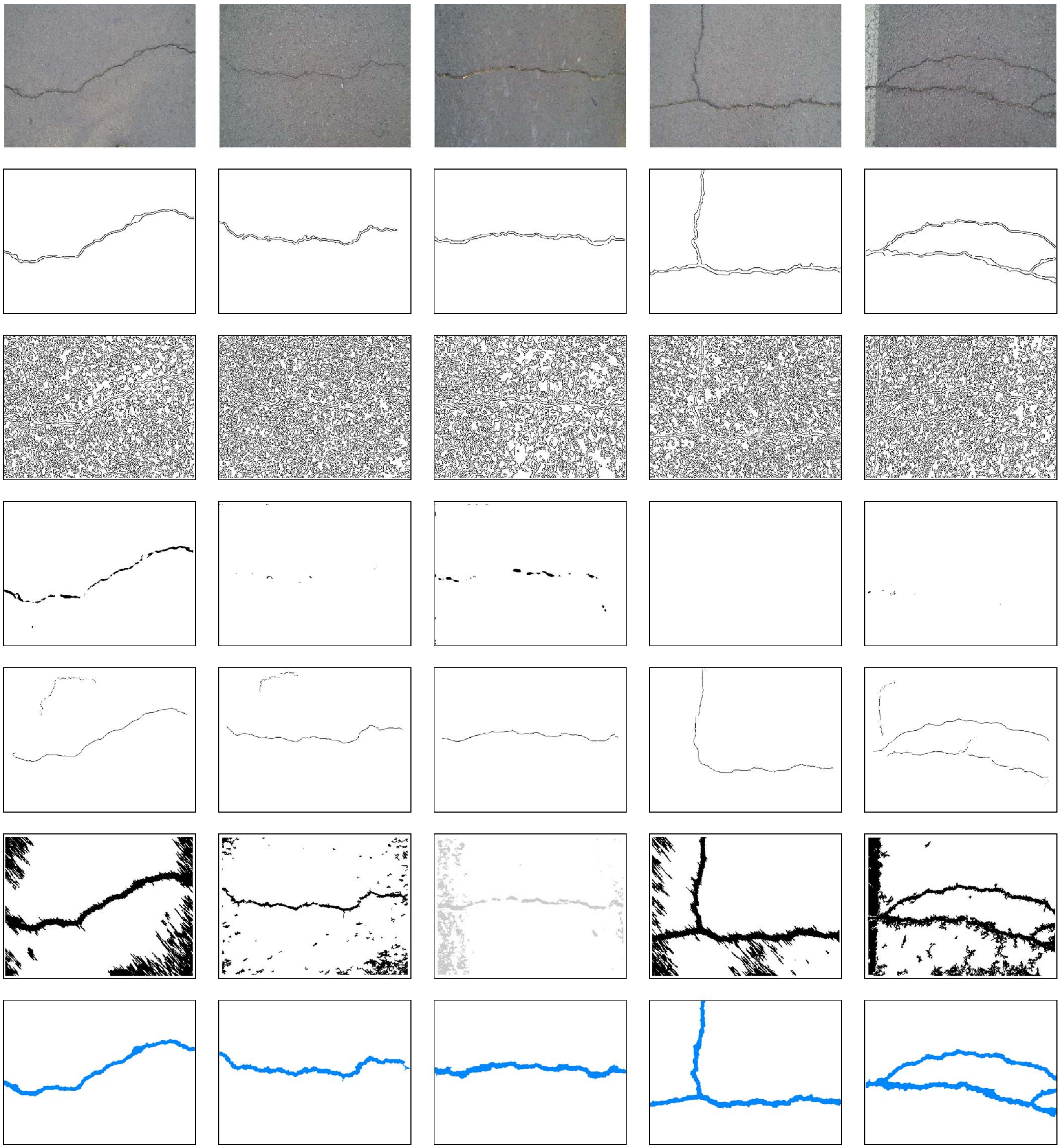


Fig. 11. Results of different algorithms on CFD (from top to bottom: original image, ground truth, Canny, CrackIT, CrackTree, FFA, and CrackForest).

as the number of ground truth cracks in the ith image and n_{ij} as the number of true positive regions that cover the jth ground truth crack in the ith image

$$\mathrm{CI} = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{1}{N_i}\sum_{j=1}^{N_i}\frac{1}{n_{ij}}\right) \tag{13}$$

The continuity is better as CI gets closer to 1.

A. CFD Results

We propose an annotated road crack dataset called CFD. This dataset is composed of 118 images, which can generally reflect urban road surface condition in Beijing, China. Each image has hand labeled ground truth contours. All the images are taken by an iPhone5 with focus of 4mm, aperture of f/2.4 and exposure time of 1/134s. The width of the images ranges from 1 to 3 mm. From Fig. 10, we can see that these images contain noises such as shadows, oil spots and water stains.

TABLE I
CRACK DETECTION RESULTS EVALUATION ON CFD

Method	Pr_{pixel}	Re_{pixel}	$F1_{pixel}$	Pr_{region}	Re_{region}	$F1_{region}$	CI
Canny	12.23%	22.15%	15.76%	0.05%	0.22%	0.08%	0.004
CrackTree	73.22%	76.45%	70.80%	84.35%	85.24%	84.79%	0.22
CrackIT	67.23%	76.69%	71.64%	93.43%	91.22%	92.31%	0.32
FFA	78.56%	68.43%	73.15%	91.55%	85.58%	88.46%	0.58
CrackForest (KNN)	80.77%	78.15%	79.44%	90.88%	93.72%	92.28%	0.62
CrackForest (SVM)	**82.28%**	**89.44%**	**85.71%**	95.75%	**95.62%**	**95.68%**	**0.67**
CrackForest (One-Class SVM)	81.25%	86.45%	83.77%	**96.73%**	92.53%	94.58%	0.65

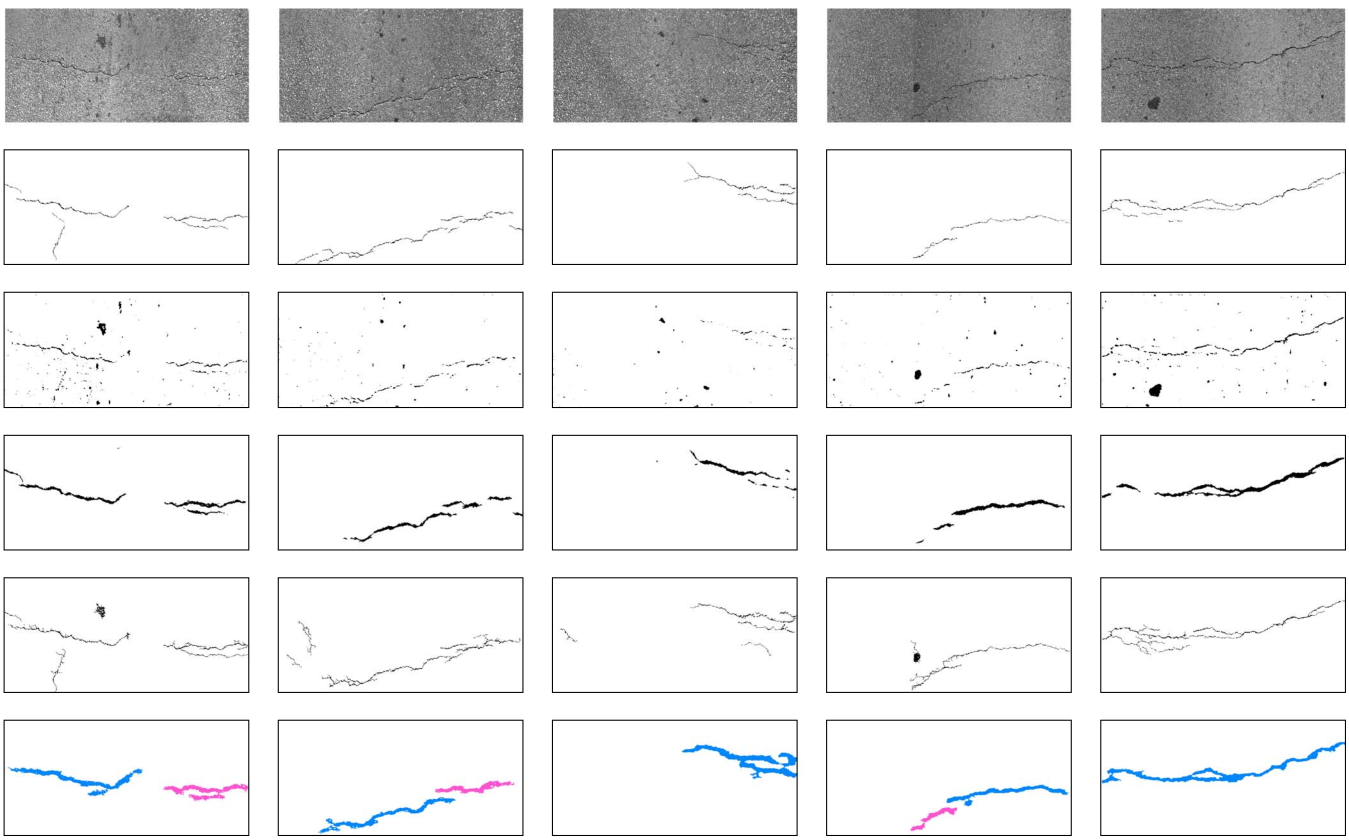


Fig. 12. Results of different algorithms on AigleRN (from top to bottom: original image, ground truth, CrackIT, FFA, MPS, and CrackForest).

We use the 60%/40% training/testing split with the images reduced to 480 × 320. Example detections on CFD are shown in Fig. 10. The first column lists the original images. The corresponding manually labeled cracks are shown in the second column as ground truth. The third column shows the preliminary detection results after applying random structured forests. Darker color indicates that the pixel is more likely to contain a crack. After the binarization, crack pixels with less confidence are removed. The use of crack descriptor allows us to transform each detected region into a vector. By applying classification method such as SVM, we can eliminate the noise regions and keep the crack regions effectively. The final detection results are shown in column 5. Our method is robust to noise.

Five methods are conducted on this dataset: Canny, CrackIT, CrackTree, FFA and CrackForest. Results are shown in Fig. 11 and summary statistics are in Table I. As it can be observed intuitively, our method outperforms the alternatives. Traditional edge detection method Canny is not suitable for road crack detection due to its high sensitivity. CrackIT does not perform well on low-resolution and low-contrast images. As a result, it fails to detect most of the crack pixels in the images. The accuracy of CrackTree is acceptable. But it may hallucinate a crack that does not exist. In addition, the width of the crack can not be observed. As for FFA, it may falsely detect landmarks as defects.

Our method CrackForest performs better than the alternatives. To be specific, CrackForest (SVM) gives both good precision and recall.

B. AigleRN Results

AigleRN dataset [49] contains 38 images with ground truth. We use 60% for training and the rest for testing.

We compare four methods on this dataset: CrackIT, FFA, MPS and CrackForest. Example AigleRN results are shown in Fig. 12 and Table II. Although CrackIT can detect most

TABLE II
CRACK DETECTION RESULTS EVALUATION ON AIGLERN

Method	Pr_{pixel}	Re_{pixel}	$F1_{pixel}$	Pr_{region}	Re_{region}	$F1_{region}$	CI
CrackIT	76.85%	74.32%	76.56%	86.52%	76.47%	81.19%	0.35
FFA	73.22%	87.52%	79.73%	84.35%	85.24%	84.79%	0.67
MPS	86.66%	**90.06**%	88.33%	87.52%	90.23%	88.85%	0.77
CrackForest (KNN)	89.47%	82.83%	86.02%	87.45%	**92.34%**	**89.83%**	0.76
CrackForest (SVM)	**90.28**%	86.58%	**88.39%**	**90.32**%	86.32%	88.27%	**0.87**
CrackForest (One-Class SVM)	85.09%	83.67%	84.37%	89.73%	88.29%	89.00%	0.85

TABLE III
CROSS-DATASET GENERALIZATION TEST FOR CRACKFOREST. TRAIN/TEST INDICATES THE TRAINING/TESTING DATASET USED

TRAIN/TEST	Pr_{pixel}	Re_{pixel}	$F1_{pixel}$	Pr_{region}	Re_{region}	$F1_{region}$	CI
AigleRN / AigleRN	90.28%	86.58%	88.39%	90.32%	86.32%	88.27%	0.87
CFD / AigleRN	87.36%	85.02%	86.17%	87.43%	85.52%	86.46%	0.79
CFD / CFD	82.28%	89.44%	85.71%	95.75%	95.62%	95.68%	0.67
AigleRN / CFD	81.27%	87.43%	84.24%	92.37%	94.33%	93.34%	0.65

of the cracks, a lot of noises are still remained. Besides, the continuity of the detected cracks is not very good. As for FFA, the precision is acceptable. But when it comes to detecting cracks with complex topology, FFA is less competitive. MPS performs well on detecting light cracks, but it may hallucinate a crack that does not exist. CrackForest shows promising results on most of the indicators. To be specific, CrackForest (SVM) still gives both better precision and recall.

C. *Cross Dataset Generalization*

To study the ability of our approach to generalize across datasets, we ran a final set of experiments. In Table III, we show results on AigleRN using CrackForest trained on CFD and also results on CFD using CrackForest trained on AigleRN. Note that images in the CFD and AigleRN datasets are qualitatively quite different, see Figs. 11 and 12, respectively.

In Table III, top, results on AigleRN of the AigleRN and CFD trained models are compared. The precision and recall do not fluctuate much using two datasets for training. Results on CFD of the CFD and AigleRN models, shown in Table III, bottom, are likewise similar.

The experimental results show that CrackForest could serve as a general purpose crack detector without the necessity of retraining.

V. CONCLUSION

In this paper, we propose an effective and fast automatic road crack detection method, which can suppress noises efficiently by learning the inherent structured information of cracks. Our detection framework builds upon representative and discriminative integral channel features and combines this representation with random structured forests. This also allows us to train our framework in a completely supervised manner from a small training set. More importantly, we can characterize cracks and eliminate noises marked as cracks by using two feature histograms proposed.

Our innovation is shown as follows: Firstly, to capture the inherent structure of the road crack, we apply integral channel features to enrich the feature set of traditional crack detection. Secondly, the introducing of random decision forests makes it possible to exploit such structured information and predict local segmentation masks of the given image patch. Thirdly, a crack descriptor, which consists of two statistical histograms, is proposed to characterize the structured information of cracks and discriminate cracks from noises. In addition, we also propose an annotated road crack image dataset which can generally reflect the urban road surface condition in China and two indicators to evaluate the performance of crack detection methods.

Experimental results prove the effectiveness of our method in suppressing noises compared to several competing methods. Our approach yields promising processing speed and state-of-the-art accuracy.

Source code is available online: https://github.com/cuilimeng/CrackForest. Our annotated road crack image dataset CFD is also available online: https://github.com/cuilimeng/CrackForest-dataset.

VI. LIMITATIONS AND FUTURE WORK

In our experiments, CrackForest has proven to be quite promising. However, it does have some limitations:

- Our method has only performed on static images so far. The video streaming is not taken into consideration. In the future, we will test our method on video datasets.
- The width of the crack is not measured in our method. We will focus on the severity level assessment in the future work.

ACKNOWLEDGMENT

Thanks professor Qin Zou and professor Manuel Avila for their kind help. And thanks professor Sylvie Chambon and Dr. Rabih Amhaz for their valuable discussions.

REFERENCES

[1] H. Oliveira and P. L. Correia, “Automatic road crack segmentation using entropy and image dynamic thresholding,” in *Proc. 17th EUSIPCO*, Aug. 24–28, 2009, pp. 622–626.

[2] H. Cheng, J.-R. Chen, C. Glazier, and Y. 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.

[3] H. Cheng *et al.*, "Novel approach to pavement cracking detection based on neural network," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1764, pp. 119–127, 2001.

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

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

[6] 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.

[7] H. Oh, N. W. Garrick, and L. E. Achenie, "Segmentation algorithm using iterative clipping for processing noisy pavement images," in *Proc. 2nd Int. Conf. Imaging Technol., Tech. Appl. Civil Eng.*, 1998, pp. 138–147.

[8] M. Petrou, J. Kittler, and K. Song, "Automatic surface crack detection on textured materials," *J. Mater. Process. Technol.*, vol. 56, no. 1–4, pp. 158–167, Jan. 1996.

[9] Y. Huang and B. Xu, "Automatic inspection of pavement cracking distress," *J. Electron. Imag.*, vol. 15, no. 1, pp. 013 017–013 017, 2006.

[10] S. Cafiso, A. Di Graziano, and S. Battiato, "Evaluation of pavement surface distress using digital image collection and analysis," in *Proc. 7th Int. Congr. Adv. Civil Eng.*, 2006, pp. 1–10.

[11] M. Gavilán *et al.*, "Adaptive road crack detection system by pavement classification," *Sensors*, vol. 11, no. 10, pp. 9628–9657, Oct. 2011.

[12] M. S. Kaseko and S. G. Ritchie, "A neural network-based methodology for pavement crack detection and classification," *Transp. Res. C, Emerging Technol.*, vol. 1, no. 4, pp. 275–291, Dec. 1993.

[13] Q. Li and X. Liu, "Novel approach to pavement image segmentation based on neighboring difference histogram method," in *Proc. CISP*, 2008, vol. 2, pp. 792–796.

[14] N. Tanaka and K. Uematsu, "A crack detection method in road surface images using morphology," in *Proc. MVA*, 1998, vol. 98, pp. 17–19.

[15] H. Oliveira and P. L. Correia, "Supervised strategies for crack detection in images of road pavement flexible surfaces," in *Proc. EUSIPCO*, 2008, pp. 25–29.

[16] C. Koch and I. Brilakis, "Pothole detection in asphalt pavement images," *Adv. Eng. Inf.*, vol. 25, no. 3, pp. 507–515, Aug. 2011.

[17] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, "A new minimal path selection algorithm for automatic crack detection on pavement images," in *Proc. IEEE ICIP*, 2014, pp. 788–792.

[18] 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 ICIP*, 2014, pp. 783–787.

[19] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, "Automatic crack detection on 2D pavement images: An algorithm based on minimal path selection," *IEEE Trans. Intell. Transp. Syst.*, 2015, 24p, DOI: 10.1109/TITS.2015.2477675. hal-01206038.

[20] 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.

[21] K. Fernandes and L. Ciobanu, "Pavement pathologies classification using graph-based features," in *Proc. IEEE ICIP*, 2014, pp. 793–797.

[22] Y.-C. J, C. Jiang, and Y. Huang, "Multiscale crack fundamental element model for real-world pavement crack classification," *J. Comput. Civil Eng.*, vol. 28, no. 4, pp. 388–396, Jul. 2012, Art. no. 04014012.

[23] Y. J. Tsai, C. Jiang, and Z. Wang, "Implementation of automatic crack evaluation using crack fundamental element," in *Proc. IEEE ICIP*, 2014, pp. 773–777.

[24] P. Kontschieder, S. R. Bulo, H. Bischof, and M. Pelillo, "Structured class-labels in random forests for semantic image labelling," in *Proc. IEEE ICCV*, 2011, pp. 2190–2197.

[25] T. S. Nguyen, S. Begot, F. Duculty, and M. Avila, "Free-form anisotropy: A new method for crack detection on pavement surface images," in *Proc. 18th IEEE ICIP*, 2011, pp. 1069–1072.

[26] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *BMVC*, 2009, pp. 91.1–91.11.

[27] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Proc. IEEE ICCV*, 2013, pp. 1841–1848.

[28] H.-D. Cheng and M. Miyojim, "Automatic pavement distress detection system," *Inf. Sci.*, vol. 108, no. 1–4, pp. 219–240, Jul. 1998.

[29] A. Ayenu-Prah and N. Attoh-Okine, "Evaluating pavement cracks with bidimensional empirical mode decomposition," *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, Mar. 2008, Art. no. 861701.

[30] H. Zhao, G. Qin, and X. Wang, "Improvement of canny algorithm based on pavement edge detection," in *Proc. 3rd Int. Conf. CISP*, 2010, vol. 2, pp. 964–967.

[31] Y.-C. Tsai, V. Kaul, and R. M. Mersereau, "Critical assessment of pavement distress segmentation methods," *J. Transp. Eng.*, vol. 136, no. 1, pp. 11–19, Jan. 2009.

[32] S. Chambon and J.-M. Moliard, "Automatic road pavement assessment with image processing: Review and comparison," *Int. J. Geophys.*, vol. 2011, 2011, Art. no. 989354.

[33] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk, "Salient region detection and segmentation," in *Computer Vision Systems*. Berlin, Germany: Springer-Verlag, 2008, pp. 66–75.

[34] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE CVPR*, 2009, pp. 1597–1604.

[35] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

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

[37] Y. Hu and C.-X. Zhao, "A local binary pattern based methods for pavement crack detection," *J. Pattern Recognit. Res.*, vol. 1, no. 20103, pp. 140–147, 2010.

[38] J. Zhou, P. S. Huang, and F.-P. Chiang, "Wavelet-based pavement distress detection and evaluation," *Opt. Eng.*, vol. 45, no. 2, Feb. 2006, Art. no. 027007.

[39] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, Jan. 1988.

[40] 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.

[41] P. Delagnes and D. Barba, "A Markov random field for rectilinear structure extraction in pavement distress image analysis," in *Proc. Int. Conf. Image Process.*, 1995, vol. 1, pp. 446–449.

[42] B. J. Lee and H. Lee, "Position-invariant neural network for digital pavement crack analysis," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 19, no. 2, pp. 105–118, Jan. 2004.

[43] 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.

[44] L. Ying and E. Salari, "Beamlet transform-based technique for pavement crack detection and classification," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 25, no. 8, pp. 572–580, Nov. 2010.

[45] J. J. Lim, C. L. Zitnick, and P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection," in *Proc. IEEE CVPR*, 2013, pp. 3158–3165.

[46] P. Dollár, Piotr's Computer Vision Matlab Toolbox (PMT). [Online]. Available: http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html

[47] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

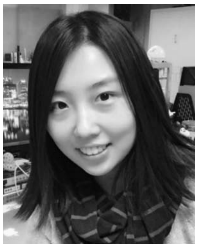
[48] H. Oliveira and P. L. Correia, "Crackit—An image processing toolbox for crack detection and characterization," in *Proc. IEEE ICIP*, 2014, pp. 798–802.

[49] S. Chambon, AigleRN. [Online]. Available: http://www.irit.fr/~Sylvie.Chambon/Crack_Detection_Database.html

Yong Shi received the Ph.D. degree in management science and computer system from The University of Kansas, Lawrence, KS, USA.

He is currently a Professor with the Chinese Academy of Sciences, Beijing, China, where he serves as the Director of Research Center on Fictitious Economy and Data Science. He is also a Professor and a Distinguished Chair of Information Technology with the College of Information Science and Technology, University of Nebraska Omaha, Omaha, USA. His research interests include data mining, information overload, optimal system designs, multiple-criteria decision making, decision support systems, and information and telecommunications management.

Dr. Shi is the Editor-in-Chief of *International Journal of Information Technology and Decision Making* and *Annals of Data Science*.

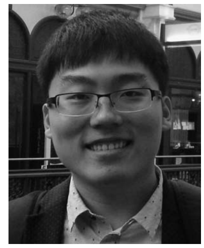


Limeng Cui is currently working toward the Ph.D. degree with University of Chinese Academy of Sciences, Beijing, China, with a focus on computer vision and machine learning.

Her work lies in image processing, object detection and scene recognition.

Zhiquan Qi is an Assistant Professor with the Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing, China.

His research interests include object detection, object tracking, change detecting, and machine learning.



Fan Meng is currently working toward the Ph.D. degree with University of Chinese Academy of Sciences, Beijing, China.

His research interests include edge detection, image segmentation, road crack detection, and proportion learning.

Zhensong Chen is currently working toward the Ph.D. degree with University of Chinese Academy of Sciences, Beijing, China.

His research is focused on image segmentation and proportion learning.