A Conditional Wasserstein Generative Adversarial Network for Pixel-level Crack Detection using Video Extracted Images

Qipei Mei, Mustafa Gül

Abstract—Automatic crack detection on pavement surfaces is an important research field in the scope of developing an intelligent transportation infrastructure system. In this paper, a novel method based on conditional Wasserstein generative adversarial network (cWGAN) is proposed for road crack detection. A 121-layer densely connected neural network with deconvolution layers for multi-level feature fusion is used as generator, and a 5-layer fully convolutional network is used as discriminator. To overcome the scattered output issue related deconvolution layers, connectivity maps are introduced to represent the crack information within the proposed cWGAN. The proposed method is tested on a dataset collected from a moving vehicle equipped with a commercial grade high speed camera. This dataset is challenging because the images containing cracks also include the disturbance of other objects. The results show that the proposed method achieves state-of-the-art performance compared with other existing methods in terms of precision, recall and F1 score.

Index Terms—Crack detection; deep learning; conditional Wasserstein generative adversarial network; Connectivity map

I. INTRODUCTION

Cracks on road surfaces are early signs for potential damage in the pavements and in the supporting structures. They serve as a good indicator to assess the current condition of the transportation infrastructure. Defects in road surfaces may delay traffic and even cause safety issues if they are severe. In addition, our road infrastructure must be improved significantly to support the autonomous vehicles of the future in the scope of smart cities. The current common practice in road surface survey is mainly based on visual inspection, which has limitations like high costs and low efficiency. Such defects as cracks or potholes may be present for a considerable amount of time before they are repaired.

In this context, the automation of crack or defect detection on pavement surfaces is invaluable and a vast amount of research has been conducted in this field [1, 2]. One of the most promising methods for automated crack/defect detection is image-based methods using cameras due to the low cost and accessibility of cameras [3]. However, it is a challenging task to distinguish the cracks from the background on images. It is difficult to find a general approach that works for most of the pavement surfaces since the cracks usually have irregular shape, the illumination conditions change for different images and there is always noise like stains or shadows from other objects that can interrupt the analysis.

In recent years, deep learning based methods have attracted much attention due to their superior performance in object detection. Girshick et al. [4] introduced a deep neural network called regions-based convolutional neural network (R-CNN). In their paper, they introduced the concept of region proposals to resolve the problem of selecting a large number of regions. A two-step detection, i.e. first generating a series of candidate regions and then conducting classification and regression on these proposed regions, was conducted in their paper. A number of other algorithms were inspired by the idea of R-CNN [5-7]. Fast R-CNN [5] directly fed an image to a convolutional neural network (CNN) to generate the proposed regions in order to achieve better performance and lower computational time. Liu et al. [8] developed a single shot multibox detector (SSD) algorithm for object detection in real time. Instead of two-step detection, SSD speeds up the detection process by eliminating the region proposal network. They achieved similar performance with R-CNN but with significantly increased speeds. Another widely used object detection algorithm is called you only look once (YOLO) [9, 10]. It has evolved to its third generation with many improvements. YOLO is also a real time object detection algorithm.

Researchers have made attempts to apply various deep learning algorithms to crack detection [11-13]. However, since cracks do not have a certain shape and usually have extremely large aspect ratio, the crack detection task is very different than other object detection tasks. Also, the publicly available datasets specifically designed to evaluate crack detection algorithms are limited. Furthermore, most of the datasets have been simplified comparing to the ones that could be encountered in real life. For example, some datasets controlled the light conditions [14], some manually exclude any disturbance and focus only on pavement surfaces using static images [15-17], and some was created for other algorithms and simply do not have enough images for deep learning [1, 16].

In this paper, a novel deep learning algorithm based on conditional Wasserstein generative adversarial network (cWGAN) is proposed to detect the cracks at pixel level. The algorithm will be pretrained on a large general dataset called ImageNet [18] and on a small crack dataset called CFD [15], and will then be trained and tested a new dataset, EdmCrack600. EdmCrack600 dataset includes 600 images extracted from

Qipei Mei and Mustafa Gül are with the Department of Civil and Environmental Engineering, University of Alberta, Edmonton, AB T6G 2R3 Canada (email: qipei@ualberta.ca; mustafa.gul@ualberta.ca).
more than 20 hours videos collected by our research group during driving and is fully annotated at pixel level. It considers factors like illumination variation, noise, interruption of other objects, etc. To the best of the authors’ knowledge, this is currently the largest and the most challenging publicly available crack dataset which is annotated at pixel level. The dataset, EdmCrack600, will be made public to benefit the community [19].

The contribution of this paper includes: 1) a novel method on the basis of cWGAN is applied for pixel level crack detection; 2) Connectivity maps are introduced to replace traditional binary crack masks to better consider the connectivity of the pixels and improve the performance; 3) A new challenging pixel-level annotated dataset is introduced to consider the real life situation [19].

In this paper, section II will review some related work for crack detection. In section III, the novel deep learning based algorithm will be explained. Section IV will describe the details and the collection procedure of the new dataset. Then, experimental results, conclusions will be presented at last in sections V and VI.

II. RELATED WORK

A. Rule based Techniques

In general, there are three major paths for crack detection utilizing images, rule based, machine learning based and deep learning based methods. In rule based methods, different combinations of filters and image processing techniques are applied to identify the cracks in images.

Gavilán et al. [20] proposed an approach combining a series of image processing techniques. First, the image was preprocessed to enhance the linear features, and non-crack feature detection was conducted to eliminate confusing area like joints or filled cracks on pavements. Then, a seed-based approach combining multiple directional non-minimum suppression with symmetry check was proposed. Zou et al. [21] developed a three step method called CrackTree. In their method, the shadow was first removed using a geodesic based algorithm. Then, a probability map was created based on tensor voting. Finally, recursive tree-edge pruning was conducted on the minimum spanning tree generated on the probability map to identify cracks. Amhaz et al. [16] introduced an improved minimal path selection algorithms with a refined artifact filtering step so that the thickness of the crack pattern can be estimated. Their approach showed superior performance than another 5 existing methods in their paper.

Overall, the major advantage of rule based methods is that neither annotation nor training process is required, so it is easier to implement the methods and verify the performance. The biggest disadvantage of this kind of methods that most of the features are handcrafted on some given datasets. In general, they cannot consider all the variation in real life images, and in most cases one method may work in one certain situation but will not work in another.

B. Machine Learning-based Techniques

Realizing the complexity in texture of pavement surfaces, variation in the illumination and the irregularity in shapes of the cracks, researchers tend to seek machine learning based algorithms for crack detection starting last decade. Comparing with traditional rule based techniques, machine learning based algorithms can implicitly consider a variety of the factors that could affect the appearance of cracks in the training process.

Hu et al. [22] treated the pavement as texture surface and cracks as inhomogeneity, and used texture analysis and shape descriptors to extract features. Support vector machines were used to classify whether a sub-region was crack or non-crack. Mathavan et al. [23] applied an unsupervised learning algorithm called self-organizing map to the crack images. Texture and color properties were integrated within the self-organizing map to distinguish cracks from background. Shi et al. [15] proposed a crack detection method based on random structured forests. In their method, integral channel features were introduced to learn the crack tokens with structured information. Then, random structured forest was applied to process the tokens and find the cracks.

C. Deep Learning based Techniques

Deep learning, as a branch of machine learning, has drawn much attention in last few years due to its superior performance in object detection and semantic segmentation [6, 10]. They were first time applied to crack detection task in 2016 [24]. In general, deep learning based crack detection methods can be categorized into two groups, i.e., region based and pixel based methods.

The region based method is less computational intensive and has been studied by a number of researchers. Cha et al. [25] developed a CNN and applied it to 40,000 regions with a resolution of 256×256 pixels for training. The algorithm can detect cracks by classifying each region separately. Gopalakrishnan et al. [26] utilized a per-trained deep CNN model and applied transfer learning to hot-mix asphalt and Portland cement concrete pavement images. Their algorithm can identify whether an image has crack or not in it. Hoang et al. [27] compared a CNN model with metaheuristic optimized edge detection algorithm. They showed that the performance of CNN was significantly better than edge detector.

However, the region based methods can only provide information about the existence of cracks and rough shape and location depending on the size of regions. The value of crack detection decreases if the accurate pattern and location of the cracks cannot be given. To resolve this issue, pixel-level crack detection are studied. Ni et al. [28] developed a method comprising two deep neural networks. The first neural network was called GoogLeNet which served as a feature extractor. Then, a second neural network including bilinear deconvolution layer and eltwise operation layer were used for pixel-level crack detection. Fei et al. [13] designed a deep neural network consisting of a preprocess layer, eight convolutional layers, and one output layer. With invariant spatial size through all layers, the method can achieve pixel level crack detection. Yang et al. [12] utilized a fully convolutional neural network (FCN) to realize the pixel level detection. Through the encoder and decoder process, the output was guaranteed to be the same size.
as input. Therefore, the prediction was included in the output probability map.

The deep learning based algorithms have shown great potential in solving crack detection problems on pavement surface. However, there are still remaining challenges due to the issues such as inhomogeneity of cracks, complexity of illumination conditions, and similarity of appearance between cracks and pavement textures. In the authors’ opinion, one of the biggest restrictions that holds back the fast development of novel algorithms is the lack of high quality and challenging datasets with complete annotations. In above studies, the researchers either tested their methods on their own datasets [12, 25-28] or very simple publicly available datasets [13]. In this context, it is difficult to compare the performance among algorithms and for new researchers to test their methods. In this paper, a challenging dataset collected by our group will be introduced in section IV.

III. METHODOLOGY

A. Overall Procedure

The proposed method based on cWGAN is described in Fig. 1. The method consists of two neural networks which are termed as generator and discriminator. In this setting, the generator outputs connectivity maps for the identification of cracks, while the discriminator checks if the connectivity maps are ground truth (“real”) or prediction (“fake”). Two networks are trained alternately to reach a Nash equilibrium after convergence [29].

In the method, taking color image patches as input, a DenseNet121 with deconvolution layers for multiple-level feature fusion is applied as generator. Unlike other deep learning based crack detection methods, the generator outputs 8 connectivity maps instead of a binary probability mask. This will be explained in following sections. Then, the connectivity maps and the original patch will be fed into the discriminator to check if this is a “fake” or “real” output.

In this paper, the cWGAN will be trained on patches which are subregions of the original large images to overcome the issues related to insufficient training data. The crack detection of the whole image will be integrated from the results coming from the patches. A post processing technique including a standard depth first search (DFS) algorithm to find connected components and to threshold out connected components with a small number of pixels is applied to the output the generator. The reason for this processing is because the cracks are usually connected components with a large number of pixels but noise has much fewer connected pixels.

B. Connectivity Maps

In this paper, deconvolution layers are used for upsampling and pixel level identification similar to some other studies [12, 30] for computational efficiency. However, it is realized that deconvolution layers are likely to generate scattered output (see Fig. 2(b)), i.e. the crack segments are not strictly connected. This is due to the mechanism of deconvolution layers where the predicted label of a pixel is solely dependent on the pixel values of a local region in original patch but is not explicitly related to the predicted labels of its neighboring pixels. Some studies suggested morphological operations, i.e., dilation and erosion, to resolve this issue [15]. However, as shown in Fig. 2(c) and (d), the performance is highly dependent on the selection of the size of morphological operations. If the size is too small, the gaps are not fully filled. If the size is too large, unnecessary parts will be considered as cracks.

This issue comes from the definition of cross entropy loss function currently used in many deep neural networks for crack detection.
detection [30, 31]. Taking Fig. 3 as an example, the crack pixels are labelled as 1 and the non-crack pixels are labelled as 0 in the ground truth. If the neural network mistakenly predicts one pixel within crack as 0, it is not different than predicting a non-crack as 1 in terms of loss function. However, in reality, an isolated wrong prediction is easier to fix than scattered prediction in crack segments.

To resolve this issue, we transform the crack detection into a connectivity problem inspired by [32]. Starting from the ground truth binary mask, each pixel should have 8 neighboring pixels. We generate 8 connectivity maps to reflect the relationship between a pixel and its 8 neighbors. As presented in Fig. 4, a regular ground truth binary crack mask is converted to 8 connectivity maps. For instance, one element in A2 connectivity map is 1 only if the corresponding element in ground truth binary mask is 1 and its left neighbor is 1 as well. During the training process, the ground truth connectivity maps are compared with predicted connectivity maps as one source to update the weights of the deep neural networks. The loss function based on the connectivity maps which is termed as $L_{content}$ could be written as equation (1) below.

$$L_{content}(G) = E_{x,y} \left[-y \log G(x) - (1-y) \log(1-G(x))\right]$$

where $G$ represents the generator. It takes $x$ as input and generates $G(x)$. The true label (ground truth connectivity maps) of input $x$ is termed as $y$. Also, at pixel level, $y_{A_k}(i,j)$ is the true label of a pixel at $i$ and $j$ in the connectivity map $A_k$ and $\hat{y}_{A_k}(i,j)$ is the predicted label for the corresponding pixel.

With the help of connectivity maps, more weights will be given to the pixels within crack segments and less weights are given to isolated pixels. In this way, the predictions are forced to be connected to each other. As can be seen in Fig. 5, the performance of deep neural network trained with regular binary mask and our proposed connectivity maps are compared. The results based on connectivity maps are more robust and less scattered because the connectivity maps force the predictions to be connected.

C. Generator

The generator is the deep neural network for crack detection. In the proposed method, as shown in Fig. 6, a DenseNet121 [33] is used as feature extractor and 3 deconvolution layers are applied for multi-level feature fusion to generate target connectivity maps.

The DenseNet121 consists of a standalone convolutional layer, a max pooling layer, 4 dense blocks and 3 transition blocks. The convolutional layer was first proposed by LeCun [34], which is now widely used for computer vision problems. Similar to filters in traditional image processing techniques, a convolutional layer is applied to the input in a sliding window form. Unlike a fully connected layer, the sparsely connected neurons in a convolutional layer can lead to better efficiency and performance. Max pooling layer replaces the value of the input feature at a certain location with its neighboring features. It can reduce the size of features and make the features invariant to small translations.

One characteristic of DenseNet121 that distinguishes it from other deep neural networks is the application of the dense block. A dense block consists of a number of convolutional layers which are densely connected with each other in a feed-forward fashion. A $1 \times 1$ convolutional layer and a $3 \times 3$ convolutional layer form a basic component in a dense block. Each dense block has multiple such components, and each component is directly connected with all following basic components within this block using skip connections except the mainstream chain-like connections. In DenseNet121, the dense blocks 1, 2, 3 and 4 (see Fig. 6) have 6, 12, 24 and 16 basic components, respectively.

The dense block does not change the height and width of the features. To follow an encoder-decoder schema for pixel level crack identification, transition blocks are applied to reduce the size of features. A transition block composes of a $1 \times 1$ convolutional layer and a $2 \times 2$ average pooling layer with a
Unlike traditional cGANs, the log functions are removed to achieve a Wasserstein distance following the suggestion from [38]. During the training process, the weights of the discriminator are clipped to a range \([-C, C]\) to fulfill the requirement Lipschitz constraint \([38]\) where \(C\) is a constant. Also, similar to [36], we add a content loss directly with the ground truth to improve the quality of the generated images. The formulae to calculate these metrics are given in equation (3).

\[
\begin{align*}
\text{precision} & = \frac{TP}{TP + FP} \\
\text{recall} & = \frac{TP}{TP + FN} \\
\text{F1 score} & = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\end{align*}
\]

In above equations, \(TP\) is true positive, \(FP\) is false positive, and \(FN\) is false negative. Following the definition given in [15], the \(TP\) is defined as the number of crack pixels that are within 5-pixel distance of a ground truth crack pixel. \(FP\) is the number of crack pixels that are beyond 5-pixel distance of a ground truth crack pixel. \(FN\) is the number of non-crack pixels that match the ground truth non-crack pixels.

### IV. Data Collection

To resolve the issues regarding the lack of reliable and complete datasets described in section I, a total number of 600 challenging road images are collected. The images are split into 420/60/120 for training, validating and testing purposes. The images are extracted from more than 20 hours videos taken from different roads in Edmonton, Canada at different times over two months. The dataset covers factors like weather condition, illumination conditions, shadows from other objects, texture difference among different pavement surfaces, etc. It aims to capture the real situations one could encounter while conducting the inspection using commercial grade camera, so no specific restrictions are applied during the collection process.

During the collection of this dataset, a GoPro Hero 7 Black was mounted beside the license plate on the rear of the Honda Pilot 2017 (see Fig. 8). During the data collection, the vehicle is driving at traffic speed (40 kph – 80 kph), and 240 fps frame rate and 1/3840 sec shutter speed is used for the camera. In several previous studies [30, 40], the camera was mounted
behind the windshield in the front of the car. There are two main reasons that we use a different configuration: 1) the windshield can reflect the light inside of the car and reduce the quality of the image; 2) the front camera is farther from the ground, a large part of its field of view (FOV) is blocked by the hood of the car. The front-mount configuration sacrifices too much spatial resolution corresponding to our analysis as follows.

As presented in Fig. 9, two configurations are illustrated. In rear-mount configuration, the angle of camera is set to 45° to balance the spatial resolution and scanned area. In front-mount configuration, the camera is facing forward like in previous studies [30, 40]. In these two configurations, the spatial resolution defined as number of pixels in unit length can be calculated as in equation (4). The spatial resolution represents how much details can be captured by the camera.

\[
\rho = \frac{1}{d} \left[ \frac{\tan(\alpha + \Delta\theta + \theta)}{m} - \frac{\tan(\alpha + \Delta\theta)}{m} \right]
\]

(4)

where \(d\) is the distance from the center of camera lens to the ground, \(\Delta\theta\) is the angle from the bottom line of FOV, \(\theta\) is the FOV, \(m\) is the total number of pixels in vertical direction and \(\alpha\) is the angle between bottom line of FOV and vertical line.

In this paper, the GoPro Hero 7 black has a FOV of 69.5°. The image has a resolution of 1920×1080 pixels. Therefore, \(\alpha\) for rear-mount configuration is 45°-69.5°/2=10.25° and for front-mount configuration is 90°-69.5°/2=55.25°. The vertical distance to the ground is 1.5 m for front-mount configuration and 1 m for rear-mount configuration. According to the above information, Table I is calculated. In Table I, the percentage of FOV, \(\Delta\theta/\theta\), and the spatial resolution (pixel/cm) for different mounting strategies are shown.

### Table I: Comparison Between Different Mounting Strategies

| Percentage of FOV, \(\Delta\theta/\theta\) | Spatial Resolution (pixel/cm) |
|--------------------------------------|-------------------------------|
|                                      | Rear-mount | Front-mount |
| 0%                                   | 8.62       | 1.93 (blocked) |
| 25%                                  | 6.99       | 0.53        |
| 50%                                  | 4.45       | 0.00        |
| 75%                                  | 1.91       | N/A         |
| 100%                                 | 0.28       | N/A         |

For rear-mount configuration, the spatial resolution is 45°-69.5°/2=10.25° and for front-mount configuration is 90°-69.5°/2=55.25°. The vertical distance to the ground is 1.5 m for front-mount configuration and 1 m for rear-mount configuration.

### Table II: Comparison Among Different Datasets

| Dataset             | No. Images | Resolution  | Device                  | Colored | Environmental effect* | Non-pavement region** | Pixel level annotation | Traffic speed | Extracted from video |
|---------------------|------------|-------------|-------------------------|---------|-----------------------|-----------------------|------------------------|---------------|----------------------|
| CFD [15]            | 118        | 480×320     | iPhone 5                | yes     | yes                   | no                    | yes                    | no            | no                   |
| Aigle-RN [16]       | 38         | 991×462     | professional camera     | no      | no                    | no                    | yes                    | yes           | no                   |
| Crack500 [17]       | 500        | 2,000×1,500 | LG-H345                 | yes     | no                    | no                    | yes                    | no            | no                   |
| GAPs [42]           | 1969       | 1920×1080   | professional camera     | no      | no                    | no                    | yes                    | yes           | yes                  |
| Cracktree200 [21]   | 206        | 800×600     | unknown                 | yes     | yes                   | no                    | no                     | no            | no                   |
| GaMM [1]            | 42         | 768×512     | professional camera     | no      | yes                   | no                    | yes                    | yes           | yes                  |
| CrackIT [43]        | 84         | 1536×2048   | optical device          | yes     | no                    | yes                   | unknown                | no            | no                   |
| JapanRoad [6, 40]   | 9,053      | 600×600     | LG-5X                   | yes     | yes                   | yes                   | yes                    | no            | yes                  |
| EdmCrack600         | 600        | 1920×1080   | GoPro 7                 | yes     | yes                   | yes                   | yes                    | yes           | yes                  |

*Environmental effect includes shadows, occlusions, low contrast, noise, etc.
**Non-pavement region means the region of image that does not belong to pavement, such as cars, houses, sky.
FOV is corresponding to percentage of image regarding the image bottom in vertical direction. For instance, Δθ/θ of 50% means the centerline of the image in vertical direction. It is seen from the table that the spatial resolution decreases dramatically as the percentage of FOV increase. This makes sense because the pavement is farther from the camera. Comparing these two configurations, we can see the front-mount configuration has significantly less spatial resolution than rear-mount configuration overall. This is because the front-mount camera is farther from the ground. Also, the 0% to 25% region is most likely to be blocked by the hood. Therefore, we choose rear-mount configuration to better utilize the GoPro camera. Some sample images are shown in Fig. 10. It is seen that the collected dataset is more difficult than most of the publicly available ones. The existence of shadows, illumination changes, stains and other objects make the crack detection challenging.

The dataset can be found here [19]. A comparison of this dataset and other publicly available dataset is given in Table II. It is seen that only GAPs [14] and JapanRoad [40] datasets consist of more images than our dataset. However, those two datasets are not pixel-level annotated. The cracks in their datasets are annotated by bounding boxes. In authors’ opinion, the bounding box is a not good way to annotate crack because of the irregular shape of cracks. Too many details will be lost if a rectangular bounding box is used to cover the cracks.

To the best of the authors’ knowledge, our dataset, EdmCrack600, is the largest crack dataset which is annotated at pixel level. It is also a very challenging one because of all the factors that are taken into consideration during the data collection process. The challenges include: 1) Change of weather conditions; 2) Significant environmental effects and noise: shadows, occlusion, stains, texture difference, low contrast because of overexposure; 3) Blurring effect due to moving of the car and the poor lighting condition.

V. EXPERIMENTS

A. Pretraining on ImageNet and CFD datasets

From a number of previous studies, it is well accepted that pretraining on irrelevant large datasets in advance before tackling the task can help improve the performance of the deep learning based algorithms [45]. This strategy is called transfer learning. In this paper, the proposed generator is first pretrained on a large object detection dataset called ImageNet [18]. It should be noted that the ImageNet dataset does not have a category related to pavement cracks.

Then, the whole proposed method is again pretrained and tested on a small crack dataset called CFD which was introduced by Shi et al. [15]. This dataset consists of 118 pavement images with resolution of $480 \times 320$ pixels. The images are taken by iPhone 5 with focus of 4 mm and aperture of $f/2.4$. In this paper, the dataset is split into 60%40% for training and testing. More details of the dataset can be found in [15].

The training losses of generator and discriminator are presented in Fig. 11. For better visualization, a 5-element moving average is taken on all the curves. As can be seen in Fig. 11(a), the generator loss has two components, one comes from the cWGAN and the other comes from the content loss described in equation (1). It is seen that the content loss continuously decrease as the training proceeds. The cWGAN loss for generator first decreases and then increases since the discriminator has learned to distinguish the “fakes” from the original images. The discriminator loss is shown in Fig. 11(b).

| Method                      | Precision | Recall | F1 Score |
|-----------------------------|-----------|--------|----------|
| Rule and machine learning based |           |        |          |
| Canny[15]                   | 12.23%    | 22.15% | 15.76%   |
| CrackTree[15]               | 73.22%    | 76.45% | 70.80%   |
| FFA[44]                     | 78.56%    | 68.43% | 73.15%   |
| CrackForest[15]             | 82.28%    | 89.44% | 85.71%   |
| MFCD[44]                    | 89.90%    | 89.47% | 88.04%   |
| Deep learning based         |           |        |          |
| ResNet152-FCN               | 87.83%    | 88.19% | 88.01%   |
| VGG19-FCN                   | 92.80%    | 85.49% | 88.53%   |
| CrackNet-V [13]             | 92.58%    | 86.03% | 89.18%   |
| Proposed method             | 96.79%    | 87.75% | 91.96%   |

![Fig. 11. Losses of the proposed method](image)

![Fig. 12. Sample results for CFD dataset](image)
“reals”. Looking at Fig. 11(b), the loss for discriminator is low at the beginning but increases afterwards. This is because initially the generator is not well trained, and the discriminator can easily distinguish the generated output from the ground truth. However, as the training proceeds, the generator can output predictions that are more difficult to distinguish. In this context, the loss for discriminator starts to increase.

Some sample images along with the ground truth and prediction are presented in Fig. 12. It is seen that the proposed method can identify the cracks with high accuracy. Table III compares the results from the proposed method with other methods. The results from all other methods are reported in their papers except ResNet152-FCN [30] and VGG19-FCN [12]. We can see that the proposed method outperforms other methods on CFD dataset in terms of precision and F1 score with large margin.

B. Performance on EdmCrack600 dataset

After pretraining on ImageNet and CFD datasets, the proposed method is further trained and tested on EdmCrack600 dataset. The losses for training and validation sets are presented in Fig. 13. In the figures, as the training proceeds, we can see the content loss for generator barely reduces, but the cWGAN loss decreases. This demonstrates the superior training performance of the proposed method than traditional encoder-decoder networks because there is an additional source for weight updating. The discriminator loss increases as the training goes on because the predictions output by the generator become more difficult to distinguish.

The performance of the proposed method in terms of precision, recall and F1 score is presented in Table IV. The Sobel and Canny detectors are standard edge detection techniques [46]. CrackIT was proposed by Oliveira and Correia [43, 47] using a series of image processing techniques. ResNet152-FCN [30] and VGG19-FCN [12] created encoder-decoder networks as suggested by [35] with ResNet152 and VGG19 as backbone networks, respectively. U-Net was introduced by [48] for crack detection. All seven methods are tested on a desktop with Intel 8700k CPU, 32GB memory and Nvidia Titan V GPU with 5120 CUDA cores where rule based and machine learning methods are run on CPU and deep learning based method are run on GPU. We can see in the table the proposed method outperforms other methods including other deep learning based methods with large margin.

Some sample results from the proposed method and existing methods are presented in Fig. 14. We can see that rule based methods cannot tackle with such complex situations where the cracks are mixed with illumination changes, shadows of trees, etc. The deep learning based methods perform significantly better. In these methods, the illumination change and the texture of the pavement surfaces are not identified as crack. However, ResNet152-FCN, VGG19-FCN and U-Net which utilize binary crack mask generates scattered output as described in section III. Also, the noise appears at different locations in the results from those three methods. The proposed method overcomes the abovementioned issues using connectivity maps and DFS based thresholding, which results in more than 5% improvement in terms of F1 score. Regarding computational efficiency, the proposed method is slightly slower than VGG19-FCN but faster than ResNet152-FCN and U-Net.

In the dataset, the images are taken in perspective view. The parts that are farther from the center of the image have lower resolutions. In this study, the perspective is not taken into consideration during the training and testing process, but it is meaningful to know how the perspective view affects the performance of the proposed method. In Fig., all 120 images with 1920×1080 pixels in the test set are split into 16×9 grids.

| Method         | Precision | Recall | F1 Score | Efficiency (sec/image) |
|----------------|-----------|--------|----------|------------------------|
| Rule and machine learning based |           |        |          |                        |
| Canny          | 1.69%     | 34.17% | 3.14%    | 0.12                   |
| Sobel          | 3.00%     | 15.24% | 4.66%    | 0.04                   |
| CrackIT        | 12.33%    | 7.14%  | 4.75%    | 6.71                   |
| Deep learning based |          |        |          |                        |
| ResNet152-FCN  | 78.98%    | 56.51% | 62.78%   | 1.94                   |
| VGG19-FCN      | 80.22%    | 59.93% | 65.18%   | 1.33                   |
| U-Net          | 76.33%    | 70.88% | 71.52%   | 2.58                   |
| Proposed method| 80.88%    | 76.64% | 76.98%   | 1.56                   |

TABLE IV

Comparison of performance for different methods on EdmCrack600 dataset
The precision, recall and F1 score are calculated for each small region separately for all 120 test images. The heat maps are generated for all three metrics where red means 100% and blue stands for 0%. The gray color represents no existence of cracks in that area. Looking at the Fig. 15(a), there is no significant difference in different regions in terms of precision except the top left corner. This means the precision is not very sensitive to the spatial resolution of the image. However, Fig. 15(b) shows that the recall is more sensitive to the location. The parts that are closer to the edges and corners have lower recall, which means the false negative is higher in these regions. This shows that the proposed method is unlikely to predict the pixels that are too far from the centerline as cracks. This is because of the distortion and low resolution at the edges of images. As a combination of precision and recall, the F1 score has similar pattern as recall.

VI. CONCLUSIONS

In this paper, a novel method based on cWGAN for pavement crack detection is proposed. Connectivity maps are introduced to overcome issues related to traditional binary crack mask. The proposed method is first pretrained on ImageNet [18] and CFD dataset [15], and then trained and tested on EdmCrack600 dataset collected from a moving vehicle by our group. The following conclusions are drawn from this study:

1) This study shows that deep learning based method is ready for road crack detection in complex environments.
2) The proposed cWGAN based method outperforms other existing methods on CFD in terms of precision and F1 score and EdmCrack600 datasets in terms of precision, recall F1 score;
3) The connectivity maps introduced in this study successfully overcome the issues about scattered output in deconvolution layers;
4) The performance of the proposed method is higher near the center of the images due to the high resolution.

Despite the success of the proposed method in this study, there are still limitations that needs to be addressed. For instance, the current version of the proposed method can only be used for crack detection. In the future, we will improve this method to detect multiple defects on the road simultaneously. Also, we will investigate more complex situations such as images taken at poor light condition and taken by low-speed camera with more significant blur issues.

REFERENCES

[1] S. Chambon and J.-M. Moliard, "Automatic road pavement assessment with image processing: review and comparison," International Journal of Geophysics, vol. 2011, 2011.
[2] A. Mohan and S. Poobal, "Crack detection using image processing: A critical review and analysis," Alexandria Engineering Journal, vol. 57, no. 2, pp. 787-798, 2018.
[3] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," Advanced Engineering Informatics, vol. 29, no. 2, pp. 196-210, 2015.
[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580-587.
[5] R. Girshick, "Fast R-CNN," in 2015 IEEE International Conference on Computer Vision (ICCV), 7-13 Dec. 2015 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.
[6] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961-2969.
Qipei Mei (M’19) is working towards his Ph.D. degree in structural engineering at the University of Alberta, Edmonton, AB. He received an MSc degree in structural engineering from the same university in 2014. Also, he received an MS degree in computer science from Georgia Institute of Technology, Atlanta, GA, in 2018 and the B.E. degree in civil engineering from the Huazhong University of Science and Technology, Wuhan, China in 2011. His research lies in applying advanced computer vision techniques on infrastructure monitoring.

Mustafa Gül is an Associate Professor in the Department of Civil and Environmental Engineering at the University of Alberta in Canada. He received his Ph.D. degree in civil engineering and M.Sc. degree in electrical engineering at the University of Central Florida. He also received his MSc and BSc in Civil Engineering from Boğaziçi University in Turkey. His research interests lies in developing technologies for creating a smart, sustainable and resilient infrastructure network in the context of smart cities.