10.1016/j.eswa.2021.115718

Two-stage Convolutional Neural Network for Road Crack Detection and Segmentation [⋆]

Nhung Hong Thi Nguyen[*]

Faculty of Engineering and Information Technology, University of Technology Sydney, 15 Broadway, Utimo NSW 2007, Australia

Stuart Perry

Faculty of Engineering and Information Technology, University of Technology Sydney, 15 Broadway, Utimo NSW 2007, Australia

Don Bone

Faculty of Engineering and Information Technology, University of Technology Sydney, 15 Broadway, Utimo NSW 2007, Australia

Ha Thanh Le

Faculty of Information Technology, University of Engineering and Technology, Vietnam National University, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam

Thuy Thi Nguyen[*]

Faculty of Information Technology, Vietnam National University of Agriculture, Trau Quy, Gia Lam, Hanoi, Vietnam

[⋆] This work has been supported in part by the Joint Technology and Innovation Research Centre - a partnership between University of Technology Sydney and Vietnam National University.

[*] Corresponding author

Email addresses: NhungHongThi.Nguyen@student.uts.edu.au (Nhung Hong Thi Nguyen), Stuart.Perry@uts.edu.au (Stuart Perry), Donald.Bone@uts.edu.au (Don Bone), lthavnu@gmail.com (Ha Thanh Le), ntthuy@vnua.edu.vn (Thuy Thi Nguyen)

Abstract

Automatic detection of road cracks is an important task to support road inspection for transport infrastructure. Various methods have been proposed for road crack detection and segmentation, however, there is no established method for handling real road images that are noisy and of low quality. In this paper, a new method utilising a two-stage convolutional neural network (CNN) is proposed for road crack detection and segmentation in images at the pixel level. Our novel contribution is a framework where the first stage serves to remove noise or artifacts and isolate the potential cracks to a small area, and the second stage is able to learn the context of cracks in the detected area. This is hence more effective than learning over the entire original noisy image. Extensive experiments on real datasets including public sources and our collected dataset have been conducted. The experimental results show that the two-stage CNN model outperformed existing approaches, especially for noisy, low-resolution images, and imbalanced datasets. Our approach achieves an F1-measure of over 0.91 on three datasets.

Keywords: Convolutional Neural Networks, Deep learning, Crack detection, Crack segmentation, Crack condition survey

1. Introduction

Road crack detection is the process of inspecting and identifying cracks on a road surface for road condition evaluation and maintenance. Road crack detection can be performed manually by human eyes or automatically by machine vision. Human inspection requires an expert's knowledge, and is laborious and time consuming. Methods for automatic crack detection from road images have been developed to improve processing speed and obtain performance better than that of humans (Oliveira, 2013). This is a challenging task in computer vision and image processing that has been the subject of research for decades (Shahin and Kohn, 1979; Zhang et al., 2016; Zakeri et al., 2017; Mohan and Poobal, 2017; Liu et al., 2019). This paper will present an advanced machine learning approach for automatic road crack detection and segmentation for road condition evaluation.

A key aspect to the calculation of the cracking index is obtaining accurate estimates of the length and width of cracks, which is in turn enabled by precise pixel-level segmentation of cracks. However the actual calculation of the cracking index has subjective components not easily automated and hence is ongoing work and is not within the scope of this article. In this work we focus on accurate pixel-level segmentation of cracks using advanced machine learning methods to support the eventual automatic calculation of the cracking index. There is a potential for significant improvement in crack evaluation by moving from manual measurement of the crack dimensions and density to semi-automatic calculation of crack positions and structure using digital image processing and machine learning. Semi-automatic detection methods are more efficient for road surveys than human inspection (Oliveira, 2013; Mohan and Poobal, 2017).

Some existing methods only focus on either detection at the region level or segmentation at the pixel level, and try to optimize only the detection or segmentation performance. However, by concentrating only on detection or segmentation alone, these methods do not see the problem holistically, which results in a reduction in overall performance. In particular, it is difficult to achieve a significant result in the case of challenging data such as noisy images with weak crack features, and imbalanced data. By developing a novel method, addressing both detection and segmentation in a single framework, our solution addresses this gap in the prior art and shows substantially improved performance compared to previous approaches.

There are many proposed approaches to detect and segment cracks in road images. Traditional digital image processing (DIP) methods such as the Canny edge detector (Canny, 1987), and methods based on Gabor filters (Movellan, 2002) exploit the change in intensity between pixels to define the edge, so a crack is considered as a feature which responds to edge detection filters. However, these algorithms are sensitive to many small details in road images and hence don't provide effective noise rejection. In addition, the optimal parameters required to allow these filters to trade-off the removal of noise against weak crack preservation change from image to image.

On the other hand, machine learning approaches, especially models based on neural networks, are used widely for object detection. Convolutional Neural Networks (CNN) are one of the most powerful recognition methods. Some work uses CNNs for detection of cracks (Zhang et al., 2016; Fan et al., 2018) while some use CNNs for pixel-wise segmentation of cracks in images (Liu et al., 2019). In the training phase, images used for training are small sections of the overall image, wherein the positive inputs contain cracks while negative inputs do not contain cracks. The output of a detection model would be a binary decision as to whether a crack was or was not in the input image section. Nguyen et al. (2018b) proposed a CNN model for crack detection. The advantages of this CNN architecture is its ability to remove almost all noise and artifacts in the original image at a relatively large size of 750 x 1900 pixels, while detecting all image patches that contain cracks. However, a disadvantage of this model is that the detected cracks are not localised as precisely as in the ground truth. The CNN model of Zhang et al. (2016) achieves a high score but suffers from the problem that the detected cracks are larger than the ground truth cracks . Liu et al. (2019) show that the handling of thin cracks of several pixels in width and broken, intermittent cracks differs from the handling of wider crack regions, so post-processing is necessary to improve the results .

The above analysis shows that models that focus on only one step of detection or segmentation are not effective, especially with thin cracks in noisy and low resolution images. To tackle these problems, we propose a new method that comprises CNNs in a two-stage framework for both detection and segmentation of cracks in road images.

Figure 1 shows our framework for road crack detection and segmentation. The image acquisition can be done by a camera attached to a car, a special purpose drone, or a smart-phone. Following this, samples for training and

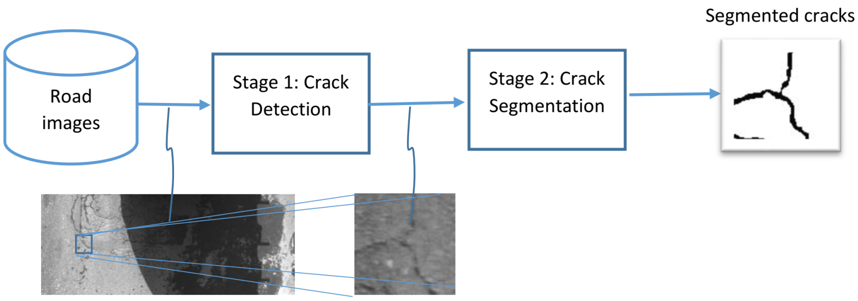


Figure 1: Our proposed framework for road crack detection and segmentation.

testing are created with the support of road experts. Two separated stages based on convolutional neural networks are trained from the samples, the first stage for detection and the second stage for segmentation of cracks. Previous work only focused on either detection or segmentation.

In this paper the two steps are done in one framework based on a machine learning approach. Our method shows a strong performance improvement when applied to unbalanced datasets such as crack datasets, where the number of crack pixels is substantially smaller than the number of non-crack pixels. Our main contributions are:

- A new two-stage architecture based on CNNs that is effective for noisy, low-resolution images, and imbalanced datasets. We show that the performance of this model is superior to models that use one stage for detection or segmentation exclusively.
- The model achieves superior performance compared to combining prior art detection and segmentation methods to create a two-stage approach, demonstrating that our specifically tailored two-stage model takes full advantage of a two-stage detection and segmentation paradigm.
- A new dataset of challenging road images containing cracks has been collected. The images are carefully labeled by experts for training and testing our model. The dataset will be made available to the research community.

2. Related works

Segmentation and detection methods for cracks in road images can be divided into two categories: the first one uses traditional digital image processing, and the second one uses machine learning approaches, especially deep neural networks. In this section, we will review related works for both detection and segmentation.

Traditional digital image processing (DIP) for crack detection and segmentation

Gabor filters are an image filtering method based on Gabor functions (Movellan, 2002). This filter is also used for image feature extraction (Ma et al., 2002; Kong et al., 2003). The Gabor filter is effective for texture segmentation (Bhoi and Solanki, 2011) and since the detection of cracks in a road image can be considered a form of texture segmentation, Gabor filters have been proposed for pavement crack detection and segmentation (Salman et al., 2013; Zalama et al., 2014). The advantage of this method is that most of the crack pixels are detected and so the crack is well segmented. However, this method is sensitive to noise.

Adaptive Thresholding is another common technique in many computer vision and graphics applications (Bradley and Roth, 2007). This technique works by comparing a pixel to the average of nearby pixels and thresholds the results to avoid boundaries across low gradient changes. This is a powerful technique for image processing because it is not sensitive to spatial changes and can remove noise. Fan et al. (2019) used a deep CNN model to detect areas of an image containing cracks, then applied the adaptive thresholding method to segment the cracks from the image . This method is simple and fast. Adaptive thresholding also achieves high accuracy on road images containing strong cracks. However, this technique cannot remove the small point-like noise characteristic of road image data.

Neural Networks and deep learning approaches for crack detection and segmentation

Deep learning is a family of machine learning methods based on multiple layers of artificial neural networks. Neural networks in machine learning are widely used in crack detection and segmentation, and these models have many advantages over traditional machine learning models (Zhang et al., 2016; Nguyen et al., 2018a; Fan et al., 2018; Mandal et al., 2018; Sobol et al., 2019). Deep learning models can learn features in images automatically, while traditional machine learning approaches need image features that are designed by users. Deep learning can also learn subjective defects which are hard to train —like minor product labeling errors. Recently, deep learning has become a powerful method that is used for detection and segmentation problems. There are 12 techniques that were examined for crack detection, six of which are based on neural networks including unsupervised and supervised methods (Oliveira, 2013). The accuracy of block-based crack detection is high, around 90% . However, this approach requires complicated pre-processing of the data before training and testing.

Deep learning methods based on CNN for detection are used in various publications for crack detection (Medina et al., 2014; Zhang et al., 2016; Fan et al., 2018; Maeda et al., 2018). Two crack datasets without pre-processing: CFD (Shi et al., 2016) and AigleRN (Chambon and Moliard, 2011), were used by Fan et al. (2018) to solves the problem of imbalanced data, with the ratio of positive pixels to negative pixels being 1:65 and 1:98.5 in the CFD dataset and the AigleRN dataset respectively. The training phase uses crack images with labels, and the experiments show that training ground truth with thinner crack labels leads to thinner output cracks (Fan et al., 2018). The small sample size cannot apply to low spatial resolution images that contain extensive noise and artifacts like the 2StageCrack dataset. Moreover, a high F1-score was achieved by Fan with an imbalanced training set with twice the number of negative samples compared to positive samples. This approach can lead to a high number of false positives (He and Garcia, 2009). Zhang et al. (2016) developed a deep learning method for crack detection, however the output crack detection probability map shows that the probability of actual crack pixels and some surrounding pixels are the same . Hence the detected crack is bigger than the real crack. There is a similar situation in Nguyen et al. (2018b), where the proposed method also outputs cracks much larger than the ground truth .

The *YOLO v2* model (Mandal et al., 2018) has been applied to automated road crack detection and classification. *YOLO v2* is based on the original YOLO deep learning method for real-time object detection (Redmon et al., 2016). The YOLO method works by dividing the input image into small boxes and predicting the coordinates of bounding boxes and class probabilities for these boxes. *YOLO v2* method achieved an F1 score of 0.8780 overall for crack and other damage detection on roads. However, this work focused just on detecting the area that contained the cracks and not on pixel-level segmentation. A model called RetinaNet (Ale et al., 2018), based on deep learning, has been proposed for road damage detection. RetinaNet can use different neural networks as the backbone for learning feature maps. A disadvantage of RetinaNet is this model detects some artifacts like paint, and line shadows as cracks. A two-module model was proposed to detect cracks rapidly (Park et al., 2019). The first module extracts cracks from road images at the same size as the original image, then patches that contain cracks are cropped, and the second module detects the cracks from the cropped patches. This proposal obtained a high precision and recall at 0.9774 and 0.9521, respectively. However, Park's model failed for images that contain both cracks and road markings, or images that have cracks on the border of the image.

The above listed crack detection approaches have a common point: using rectangular image region samples that contain cracks as positive examples and image region samples that do not contain cracks as negative examples to train an output model that classifies the rectangular regions as containing cracks or not. The accuracy of experiments is evaluated at the image region classification level, not the pixel classification level. Therefore, some segmentation methods are proposed to detect the crack at the smallest unit of image, the pixel. In the next sub-section we survey some segmentation methods that are used to segment objects and cracks.

FCN stands for Fully Convolutional Networks (Long et al., 2015). FCN transfers knowledge from VGG16 (Simonyan and Zisserman, 2014) to perform segmentation. In the FCN architecture, a 1×1 convolutional layer is used to convert the fully connected layers in VGG16 into fully convolutional layers that enable the classification net to output a low resolution heatmap. FCN is complicated and it takes a long time for training because of the large number of kernels in the convolutional layers.

U-Net is widely used for image segmentation (Ronneberger et al., 2015). The U-Net uses a segmentation model that includes two parts: a contracting part for computing features and an expanding part for spatially localizing patterns in images. In the U-Net architecture, the authors use *concat* layers that concatenate feature maps in the same level of both the encoder and decoder parts and improve the localization accuracy of objects. The U-Net model achieved a high IoU (Intersection over Union) result, at 0.9203 on the ISBI (IEEE International Symposium on Biomedical Imaging) cell tracking challenge.

SegNet is a deep convolutional encoder-decoder architecture for image segmentation (Badrinarayanan et al., 2017). SegNet makes use of a special technique in that it only stores *max-pooling indices*, therefore using less memory during training compared with fully convolutional networks. SegNet is effective in boundary delineation and the experiments show that SegNet outperforms all the other methods for most objects. *XNet* is another convolutional neural network that is used for the segmentation of medical X-Ray image data (Bullock et al., 2019). Although this net uses small datasets for training, the results show an overall accuracy of 92% and an AUC (Area Under the Curve) of 0.98. XNet obtains more than 90% in all evaluating indexes such as F1-Score and AUC when applied to three categories of X-ray images.

DeepCrack is a new model that utilizes a deep hierarchical neural network for crack segmentation in which Liu et al. (2019) proposed a deep hierarchical CNN for crack segmentation at the pixel level. The DeepCrack model uses the first 13 layers which correspond to the first 13 layers in VGG-16, but the fully connected layers and fifth pooling layer are removed to achieve meaningful side-output with different scales and decreased memory requirements and computation time. A guide filter based on *guided feathering* is used by He et al. (2012) to achieve the final refined prediction and to remove noise in the low level prediction.

In summary, the above image segmentation methods share some common characteristics including that they are built based on a backbone CNN. Some existing models only focus on either detection at the region level or segmentation at the pixel level, and try to optimize only the detection or segmentation performance. However, by concentrating only on detection or segmentation alone, it is difficult to achieve a high performance result in the case of challenging image data with noisy and weak features such as road images.

In this work, we will develop a new method for efficient learning and characterization of road images. The new model combines the two levels of detection and segmentation of cracks, and has the advantages of both state-of-the-art detection and segmentation methods in a single unified approach.

3. Proposed method

In the following we will present our proposed model that combines two stages of detection and segmentation. The first stage uses a CNN as a detection model that is trained on image patches to seek all regions that contain cracks, while also removing most of the noise and artifacts. The second stage involves segmentation of the crack at the pixel level in small patches instead of the whole original image. As a result, the combined model has the advantages of both the detection and segmentation approaches.

Figure 2 shows the proposed two-stage architecture for crack detection and segmentation at the pixel level.

CNN architecture for road crack detection

In the convolutional neural network, the convolution function is used for the *extraction of features* (LeCun et al., 1998). An input image is considered as a matrix $\mathrm{W}_{\mathrm{m,n}}$ of size $M \times N$. This input image is convolved with a kernel $\mathrm{k}_{\mathrm{p,q}}$ of size $P \times Q$. In our proposed approach, we use a square input image of

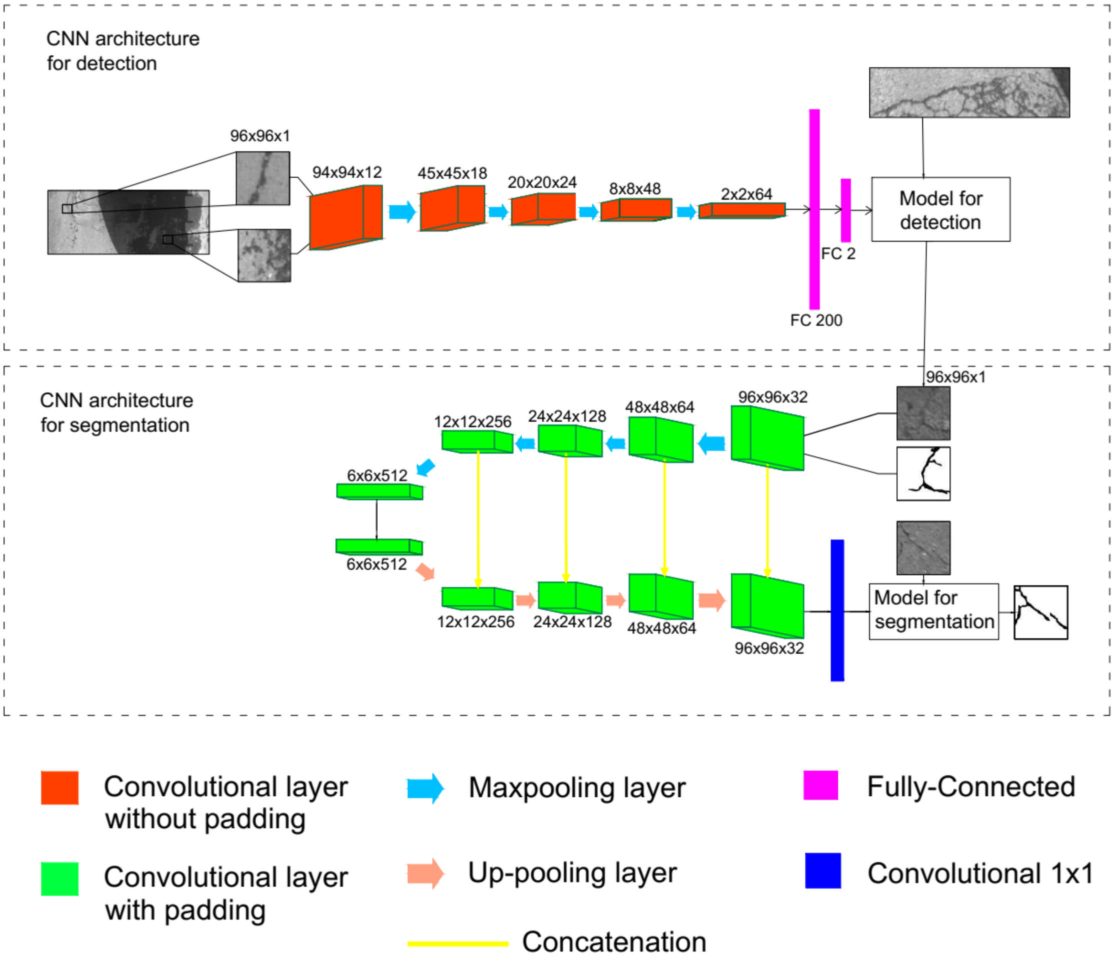


Figure 2: The proposed two-stage model for detection and segmentation.

size 96 × 96 pixels and a kernel of size 3 × 3. This operation can be written as equation (1), where $\mathrm{c_{i,j}}$ is the (*i*,*j*)th element of the convolutional layer.

$$c_{i,j} = f(\sum_{p=m}^{M}\sum_{q=n}^{N} w_{i,j} \cdot k_{i+p,j+q} + b_{m,n}) \tag{1}$$

where *f* is a transfer function, $\sum_{p=m}^{M}\sum_{q=n}^{N} w_{i,j} \cdot k_{i+p,j+q}$ is the convolution operator of the input and the kernel, and $b_{m,n}$ is the bias input to the layer.

Each *convolutional layer* represents a feature map of the image such as shape, edge or intensity. For enhancing useful features and emphasising weak crack features, a number of neural network layers must be used. In this work, we use a five-layer CNN model to extract the features of the crack samples. The number of layers is chosen empirically. It is shown that for our particular problem, a five-layer CNN architecture achieves the highest score. The number of kernels in the first, second, third, fourth and fifth CNN layer are 12, 18, 24, 32, and 64, respectively. A one-pixel stride is applied for all CNN layers. This helps to compute all of the features, even when the cracks are small. We did not use padding for the layers of the detection-stage CNN, which means samples used by the CNN are not given additional padding when convolved with the kernel. This technique reduces the size of each layer and hence the number of parameters. After each CNN layer, max-pooling layer with kernel size of 2×2 is added. A max pooling function selects the maximum value in each 2×2 block, also reducing the size of the layers of the network. In this way, the max pooling layer helps to decrease the number of weights and avoids over-fitting.

Integrating features is achieved by two *fully-connected (FC)* layers. While the previous CNN layers extract the crack features, the FC layers are used for collating all learned features. The first FC layer contains 200 neurons for flattening the features and arranging them into a vector. The feature vector collates all the feature information and high-weight elements from the prior convolutional layers. The second FC layer includes two neurons corresponding to the two classes to be classified: crack and non-crack.

Probability of samples being either a crack or non-crack region sample must be calculated and this is the final step to classify an image patch as crack or non-crack. In this work, we use the *softmax function* in the last layer. The output of this function is the probability that a given region sample contains a crack. The range of this value is from 0 to 1, representing the certainty of the region containing no-cracks or at least one crack, respectively.

Prevention of overfitting in this proposed model is achieved using a pooling technique. After each convolutional layer, a *max-pooling* is performed to decrease the number of weights. We also used image augmentation techniques such as shifting, zooming, flipping, and rotation of the image patches to diversify the training data, which also contributes to preventing overfitting.

CNN architecture for road crack segmentation

The proposed convolutional neural network architecture for segmentation (Figure 2) comprises two parts: a contraction or encoding part and an expansion or decoding part. Encoder-decoder models are used widely in image segmentation (Ronneberger et al., 2015; Badrinarayanan et al., 2017; Bullock et al., 2019). These models are also used for semantic segmentation of cracks in road image data.

The contracting part is designed to compute the features of objects in images. We utilize a model that contains five layers of convolutional neural networks for extracting image features. We use an architecture that increments the number of filters in each layer in a manner consistent with previous work on segmentation and object detection (Simonyan and Zisserman, 2014; Ronneberger et al., 2015; Nguyen et al., 2018b). As the network is built for segmentation, this task aims to detect and classify an object not only at the block level but also at the pixel level, so we do not use a small number of kernels as in previous work (Nguyen et al., 2018b). In addition, by contrast with U-Net, which contains a very large number of filters in each CNN layer, we decrease the number of kernels in each successive layer by a factor of two, such that there are 32, 64, 128, 256, 512 kernels in sequence from the first layer to the fifth layer. We used padding in all convolutional layers in the segmentation stage. This means that all samples are given additional padding when convolved with the kernel, so size of output segmented images is kept the same as the input images. A difference in the number of elements between the detection and segmentation stages is due to the use of padding in the segmentation stage that is not used in the detection stage. A max-pooling layer is added after each convolutional layer to decrease the number of weights and the sample size.

The expanding part is the inverse of the contraction process. While the contracting part is used for extracting features, the expanding part is used for spatially localizing patterns in images. We use a symmetric structure of decoder and encoder. The expanding part expands the input into a larger image as we pass through each layer. So, the expanding part can work as a *deconvolutional network*, which acts in some sense as the inverse of the *convolutional network*. In addition, we use *up-sampling* with a size of 2x2 to restore the size to that of the input samples. After the fifth layer, the output image is the same size as the input.

Convolution 1×1 can be used to increase or decrease the number of channels while maintaining the size of the image. In this architecture, a convolution layer with a 1×1 kernel that acts as a *sigmoidal activation function* is used. This layer acts to process the feature maps to generate a segmentation map and thus categorize every pixel of the input image.

The concatenation layer is used for concatenating two layers together along one axis. The aim of concatenating layers is to enhance the shuffling of information across many layers of the network. In this model, we use four concatenation layers, each concatenation layer concatenates a pair of convolutional and deconvolutional layers into a single layer that contains both feature maps of the two input layers.

4. Experiments and Results

4.1. Dataset preparation and experimental environment

For the purpose of comparing and assessing the proposed method, the following benchmark and self-created datasets are used.

DeepCrack dataset. This dataset contains 537 images for training and testing each with a size of 544x384 pixels (Liu et al., 2019). Ground truth images at the pixel level are available. We use this dataset in the detection and segmentation phases for assessing our proposed method.

CrackIT dataset. The CrackIT dataset (Oliveira, 2013) was collected in Portugal and Canada. This dataset has no ground truth at pixel level, so we initially used this dataset in the detection phase. We apply previous detection methods as well as our proposed detection method to this dataset and our dataset and compare the results of these methods. We also created ground truth at the pixel level for this dataset in order to further evaluate the performance of our approach. We used a tool from the Matlab commercial software package (The MathWorks, 2019) to allow experts to manually create the ground truth in the same manner as ground truth for the 2StagesCrack dataset as described below.

2StagesCrack dataset. This dataset was collected in Vietnam and the images were taken from a monochrome camera attached to a trailer, driven across a variety of roads (Nguyen et al., 2018b). The images were collected under a variety of weather conditions, and in many cases the road was not clean, so the dataset contains noise. Moreover, the monochrome camera used is low quality and attached to a high speed car, so the captured images are low resolution. In this work, we created our ground truth crack samples by annotating the images at the pixel level and denoted the resultant ground truth augmented dataset as the **2StagesCrack** dataset. The ground truth annotation for the dataset is supported by road experts. The *2StagesCrack* dataset is used in the experiments for the segmentation phase, and contains a total of 4000 samples, including 2000 original images of size 96×96 pixels and 2000 associated ground truth images. We used the "Image Segmenter" tool in the "Image Processing and Computer Vision" toolbox in the Matlab software (The MathWorks, 2019) to create binary images for training. Each original image is processed for ground truth with the "Draw Freehand" function in the Matlab Segmenter tool to outline the boundary of the cracks. The image is then binarised such that the crack pixels appear in

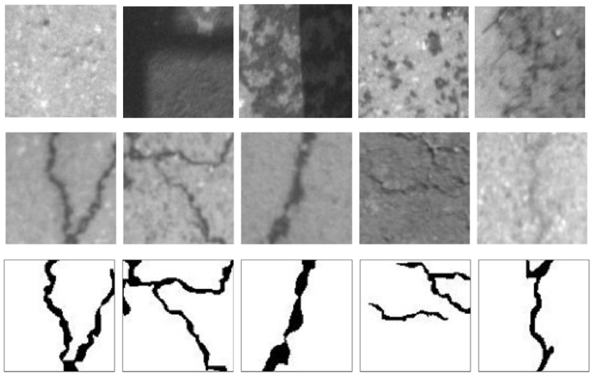


Figure 3: Examples of images and ground truth for the 2StagesCrack dataset.

black and the background pixels are in white.

Figure 3 shows examples of samples belonging to the *2StagesCrack* dataset. The first row shows negative samples, that contain no cracks and are used in the detection stage. The second row shows the positives samples, that contain some cracks and are used in the detection stage and segmentation stage as original images. The third row shows the associated ground truth images of the samples in the second row. In the two last rows, the first and second images show connected cracks and cracks with strong intensity. The first and the second images contain connected cracks. In addition, the cracks in the second row are thin in some parts. The third image shows a large crack suffering a degree of blur which may be due to the movement of the sensor vehicle when the image was captured. The fourth example is a discontinuous, thin and unclear crack, possibly affected by dirt on the road. The last example is a sample of a single, weak crack.

Figure 4 shows examples of samples belonging to the CrackIT dataset. The first row shows negative samples, the second row shows positive samples, the third row shows the original ground truth (Oliveira, 2013), and the last row shows the ground truth created for the experiments in this paper. The original ground truth contained only a one-pixel representation of the centerline of the crack. We enhanced the ground truth images to cover all pixels belonging to

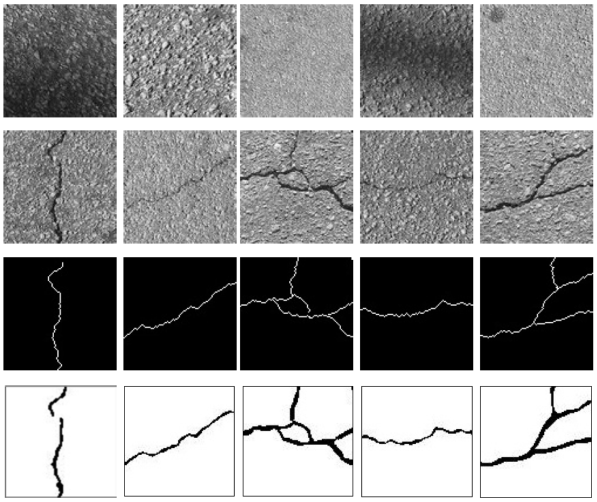


Figure 4: Examples of images and ground truth for the CrackIT dataset.

the crack in a manner that can be used for pixel-level segmentation.

Table 1 shows a comparison of the key features of the three datasets used in this work: CrackIT, Deepcrack, and 2StagesCrack. Each of these crack datasets are imbalanced. The total number of pixel cracks in whole dataset is less than 1% in the CrackIT dataset (Oliveira, 2013), 3.54% in the DeepCrack dataset (Liu et al., 2019), and around 0.3% in our dataset. With our strategy of detection at the sample level and segmentation at the pixel level, the imbalance present in the original images is mitigated, with the imbalance in the segmentation stage becoming 8.3%, 16% and 6% in the CrackIT, Deepcrack, and 2StagesCrack dataset, respectively. Table 1 also illustrates that the 2StagesCrack dataset is a large scale dataset, at low resolution and contains a variety of noise and artifacts.

To increase the number of samples and diversify the images, we applied data augmentation techniques to the images before the training phase such as rotations, shifts, zoom, and flipping to create additional data samples.

All experiments in this work are implemented on a PC with a single CPU Intel core i5 processor and processor speed of 2.81 GHz, 8GB of RAM, running the Windows 10 operating system.

Table 1: Comparison of datasets.

Datasets Characteristics	**CrackIT**	**Deepcrack**	**2StagesCrack**
Size of image (pixels)	600 ×800	544 ×384	750 ×1900
Spatial resolution	1 pixel 1 square mm	No information	4 pixels area 1 square mm
Noise and artifacts	Shadow, paint	No information	A lot of shadows from trees and cars; paint, water streaks, wheel streaks, waste material on road
Percentages of crack pixels	Less than 1%	3.54 %	0.31%
Other characteristics	Road and sidewalk images	Various types of image such as roads, building	Road images
	Strong intensity cracks	Strong intensity cracks	Various level of cracks, from weak to strong intensity; including cracks under shadows
	Thin crack	Large crack	From thin to large cracks
	Collected in Portugal	No information	Collected in Vietnam

4.2. Methods for evaluation

The commonly used metrics for evaluation of detection and segmentation methods are Precision *Pr*, Recall *Re* and F1-score *F1*. These are defined as below:

$$Pr = \frac{TP}{TP + FP} \tag{2}$$

$$Re = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2 \cdot Pr \cdot Re}{Pr + Re} \tag{4}$$

where *TP*, *FP*, *FN* are the numbers of True Positive, False Positive, and False Negative samples, respectively. In the detection stage, a positive sample is a region sample of the image that contains at least one crack, and a negative sample is an area that contains no cracks. In the segmentation phase, a positive sample is a crack pixel, and a negative sample is a background pixel.

In testing, with *N* patches (N×96×96 pixels) of images in total and *n* patches (n×96×96 pixels) among the *N* are detected as crack areas, it follows that (*N*-*n*) are non-crack areas. The second stage segments the crack pixels from the results of the detection phase, so the F_1 for the (*N*-*n*) non-crack samples is $\mathrm{F}_{1\text{-Det}}$, and the F_1 of the n crack patches is $F_{1\text{-Det}}$*$F_{1\text{-Seg}}$. For the experiments in this paper, we use a stride of one pixel in the detection stage, so the number of detected patches is equal to the number of input pixels to the segmentation stage. Since the degrees of freedom of $F_{1\text{-Det}}$ and $F_{1\text{-Seg}}$ are the same, they can be combined to create a final F_1-score. The final F_1-score for crack segmentation at the pixel level in whole image can be calculated as per equation (5):

$$F_{1\text{-final}} = F_{1\text{-Det}} * \frac{F_{1\text{-Seg}} * n + (N - n)}{N} \tag{5}$$

where $F_{1\text{-Det}}$ is the F_1 score of the first stage of detection and $F_{1\text{-Seg}}$ is the F_1 score of the second stage of segmentation.

Road crack datasets are imbalanced data, where the number of true crack pixels is very small compared with the number of true background pixels. F1-score compensates for this effect and is hence a reasonable method for evaluating the performance of the proposed method.

4.3. Results

Table 2 shows the *Precision* and *Recall* results of various models for the detection stage. We implemented a SVM (Support Vector Machine) with a linear-kernel and using HoG (Histogram of Oriented Gradients) features for crack detection. We also compared the results of the proposed model with some other traditional machine learning methods and some existing models. Our architecture used a smaller number of parameters than prior convolutional neural networks as shown in the last column of Table 2. The results show that the proposed model achieves the highest F1-score, at 92%.

Table 2: Precision and Recall of different models in the Detection stage

Method	CrackIT dataset			DeepCrack dataset			2StagesCrack dataset			Total parameters
	Pr	Re	F1-score	Pr	Re	F1-score	Pr	Re	F1-score	
Support Vector Machine	0.89	0.57	0.70	0.87	0.59	0.70	0.90	0.55	0.68	N/A
Decision Tree	0.82	0.59	0.68	0.80	0.65	0.72	**0.94**	0.51	0.66	N/A
Random Forest	**0.99**	0.54	0.69	**0.99**	0.50	0.66	0.98	0.52	0.68	N/A
Fan's method (Fan et al., 2018)	0.64	0.70	0.67	0.70	0.75	0.72	0.99	0.57	0.72	924,562
Zhang's method (Zhang et al., 2016)	0.74	0.97	0.83	0.83	0.86	0.84	0.85	0.82	0.77	205,466
Proposed method	0.92	**0.89**	**0.90**	0.93	**0.90**	**0.91**	0.92	**0.91**	**0.92**	**58,404**

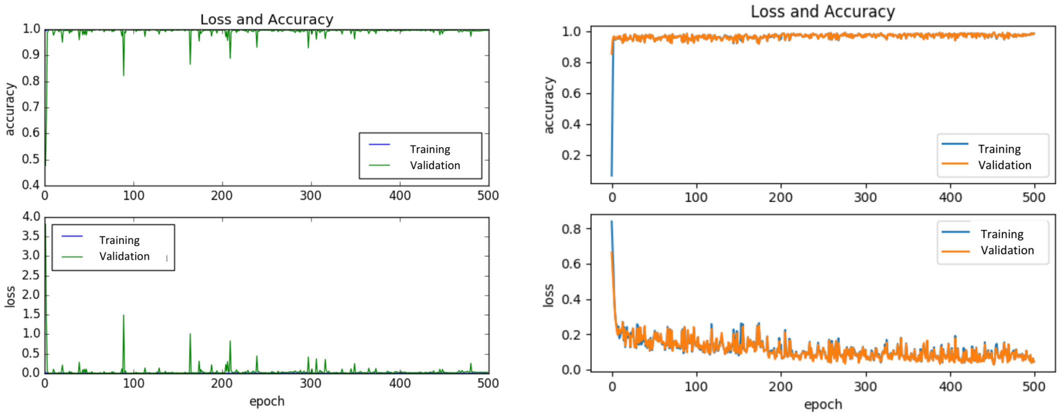
Table 3: Precision and Recall of different models in the Segmentation stage

Method	CrackIT dataset			DeepCrack dataset			2StagesCrack dataset			Total
	Pr	Re	F1-score	Pr	Re	F1-score	Pr	Re	F1-score	parameters
Gabor Filter	0.54	0.64	0.59	0.55	0.85	0.67	0.61	0.23	0.34	N/A
Adaptive Thresholding	0.17	0.82	0.28	0.25	0.90	0.40	0.53	0.31	0.38	N/A
K- Nearest Neighbour	0.52	0.48	0.49	0.66	0.50	0.57	0.69	0.28	0.40	N/A
SegNet	0.62	0.62	0.62	0.67	0.67	0.67	0.55	0.55	0.55	29,475,866
U-Net	0.71	0.84	0.77	0.87	0.73	0.79	0.49	**0.88**	0.63	31,031,685
Proposed Method	**0.88**	**0.85**	**0.87**	**0.93**	**0.94**	**0.93**	**0.70**	0.78	**0.74**	**7,672,549**

Table 3 shows the results of the proposed segmentation stage and four other models on the two datasets that have pixel-level ground truth available. Firstly, we used two classical DIP methods on the images, Gabor filter and Adaptive Thresholding. K-Nearest Neighbour (K-NN), one of the most fundamental and straightforward methods (Peterson, 2009), is used in our experiments for crack segmentation and compared with the proposed method. After that, we implemented two CNN models for segmentation, U-Net and SegNet. The results indicate that the proposed method has improved segmentation performance in both datasets. We also compare the number of parameters between the three models based on CNN. It is clear that the proposed model for segmentation needs a smaller number of parameters, while still achieving better performance.

Figure 5 shows the accuracy and loss of the proposed architecture in training for detection and segmentation for the 2StagesCrack dataset. Figure 5a and Figure 5b show the training and validation accuracy and loss of the detection model and segmentation model, respectively. These graphs indicate that our proposed models have high accuracy and low loss in both the detection and segmentation phases. For the model we selected, the convergence was rapid. This is consistent with the observations of Zhang et al. (2016), where less than 20 epochs where shown to be sufficient for convergence. A total of 500 epochs are shown here to demonstrate the stability of the model.

To calculate the final F1-score as equation 5, the $F_{1\text{-Det}}$ and $F_{1\text{-Seg}}$ of the detection phase and the segmentation phase of the different methods are shown in Table 2 and Table 3. If an original image is of size 1900x750 pixels, there are 155 non-overlapping region samples of size 96x96 pixels, so $N = 155$. Assuming



(a) Accuracy and loss during training for the detection stage for the 2StagesCrack dataset.

(b) Accuracy and loss during training for the segmentation stage for the 2StagesCrack dataset.

Figure 5: Accuracy and loss during training for the detection and segmentation stages for the 2StagesCrack dataset.

that the number of regions with cracks detected from the detection stage is 10, then $n = 10$. Table 4 shows the final F1-score of the various models when combining different detection methods with different segmentation methods when applying for the 2StagesCrack dataset. It is shown that:

- If the segmentation stage is only applied to the regions flagged as containing cracks by the detection step, the final performance is improved compared to the use of a single stage of segmentation applied to the entire image. While the segmentation model only achieves around 70% when applied to the entire image, the two-stage model achieves about 90% in terms of F1-score.

- The proposed architecture achieves superior performance in both the detection and segmentation steps alone compared to prior detection or segmentation methods. Hence, the proposed two-stage approach achieves superior performance compared to the other detection/segmentation combinations.

It can be seen that the $F_{\text{1-2stages}}$ score increases as n decreases. So, for the

Table 4: F1-score for different combinations of detection and segmentation for the 2StagesCrack dataset.

Detection methods Segmentation methods	Fan's method	Zhang's method	Proposed Detection stage
SegNet method	0.69	0.74	0.88
U-Net method	0.69	0.75	0.89
Proposed Segmentation stage	0.71	0.76	**0.91**

Table 5: The MCC score for the examined methods at the Segmentation stage for the three datasets.

Method	CrackIT dataset	DeepCrack dataset	2StagesCrack dataset
Gabor Filter	0.57	0.69	0.25
Adaptive Thresholding	0.60	0.63	0.31
K- Nearest Neighbour	0.44	0.47	0.33
SegNet	0.59	0.61	0.52
U-Net	0.77	0.75	0.72
Proposed Method	**0.85**	**0.92**	**0.74**

2StagesCrack dataset, which contains a lower density of cracks and a higher level of noise, this method achieves a high F1 score in total.

The MCC (Matthews Correlation Coefficient) is another metric for measuring the quality of binary classification (Matthews, 1975). The MCC score provides a less biased evaluation compared to F1-score when the dataset is negatively imbalanced (Chicco and Jurman, 2020). All datasets used in this paper are positively imbalanced datasets. However, for further comparison, we also calculate the MCC score in the segmentation stage for the three datasets, as is shown in Table 5. Our method achieves the highest MCC score for all given datasets.

Table 6 shows the total parameters of various models used for detection and segmentation, and testing time for the 2StagesCrack dataset. The testing time is the average time used for processing one sample in milliseconds. It can be seen that our architecture has a smaller number of total parameters and a shorter testing time in comparison with that of the combination of Zhang's method and U-Net model. In all experiments, we used the same hardware configuration for the testing phase. In addition to the number of total model parameters, there are various factors that affect the testing time, including the software library used (in our experiments we used *numba* instead of *numpy*), the hardware platform (CPU or GPU, in our experiments we used the CPU) and the pixel stride chosen during testing. After investigation, we found that a sizeable percentage of the testing time was taken by the time for software and hardware to extract samples from the images for testing. Hence testing time scales with the number of parameters only to a certain extent.

Table 6: Total parameters and testing time for the 2StagesCrack dataset.

Model		Total parameters	Average testing time per sample
Detection	Zhang's model for detection	205,466	19.2ms
	Proposed stage for detection	58,404	**13.5ms**
Segmentation	U-Net model for segmentation	31,031,685	153ms
	Proposed stage for segmentation	7,672,549	**103ms**
Two stages model	Two stages of Zhang and U-Net	31,052,151	172.2ms
	2StagesCrack model	**7,730,953**	**116.5ms**

In figures 6,7,8,9, 10 and 11, we show some typical examples of detection and segmentation results produced by different methods. The first five original images are from the 2StagesCrack dataset and have a size of 750x1900 pixels, the last images are from the CrackIT dataset and have a size of 600×800 pixels. Each of these figures contains 6 sub-figures: the original image, ground truth image, the detection result obtained by the method of Zhang et al. (2016), the result obtained from combining the detection method from Zhang et al. (2016) and the segmentation method from Ronneberger et al. (2015), detection result from the first stage of the proposed model, and the final image is the result by the proposed two-stage architecture. For post-processing, the method Otsu (1979) and some other techniques are applied to obtain the binary images from the segmented results (Ni et al., 2019). The results from Otsu or a median filter may be not stable across the different variety of inputs encountered. In our method, we apply an automatic threshold to classify pixels as cracks when their probability exceeds the F1-score of the positive crack segmentation and export the results as binary images. Our experiments show that a threshold that is equal to the F1-score achieves the best output images.

Figure 6 shows a typical example. The original image, Figure 6a, contains a single crack. There are many artifacts in this image, such as a shadow, and wheel tracks in the water. While the first stage of the proposed method detects the true crack with the surrounding area, the method from Zhang et al. (2016) also identifies one side of the shadow area as a crack. The proposed method is more robust compared to prior methods and less likely to misidentify shadows and artifacts on the road surface as cracks. Figure 7 shows a similar example with a single crack and once again, we can observe that the proposed method works very well.

Figure 8a is an example of connected cracks. The method from Zhang cannot detect all regions containing cracks, and detects a number of false negative regions. For the detected crack in Figure 8c, the U-Net model also has poor segmentation in parts that results in false negative samples.

Figure 9 shows other experimental results on connected cracks. The image in Figure 9a is a challenge because it was captured in weak light and the road surface is covered with water in many places. The first stage of the proposed model detects all patches that contain cracks but also detects some background pixels that surround the true crack pixels. Following this, the second stage subtracts the remaining non-crack pixels. In contrast, the two other methods ignore many true positive cracks.

Figure 10 is a special instance because the cracks are thin and in the shadows. These cracks were captured under poor illumination conditions because of the shadows, and this image contains white noise around the cracks. The proposed method detects and segments most of cracks, while the other methods recognize only a small proportion of the cracks.

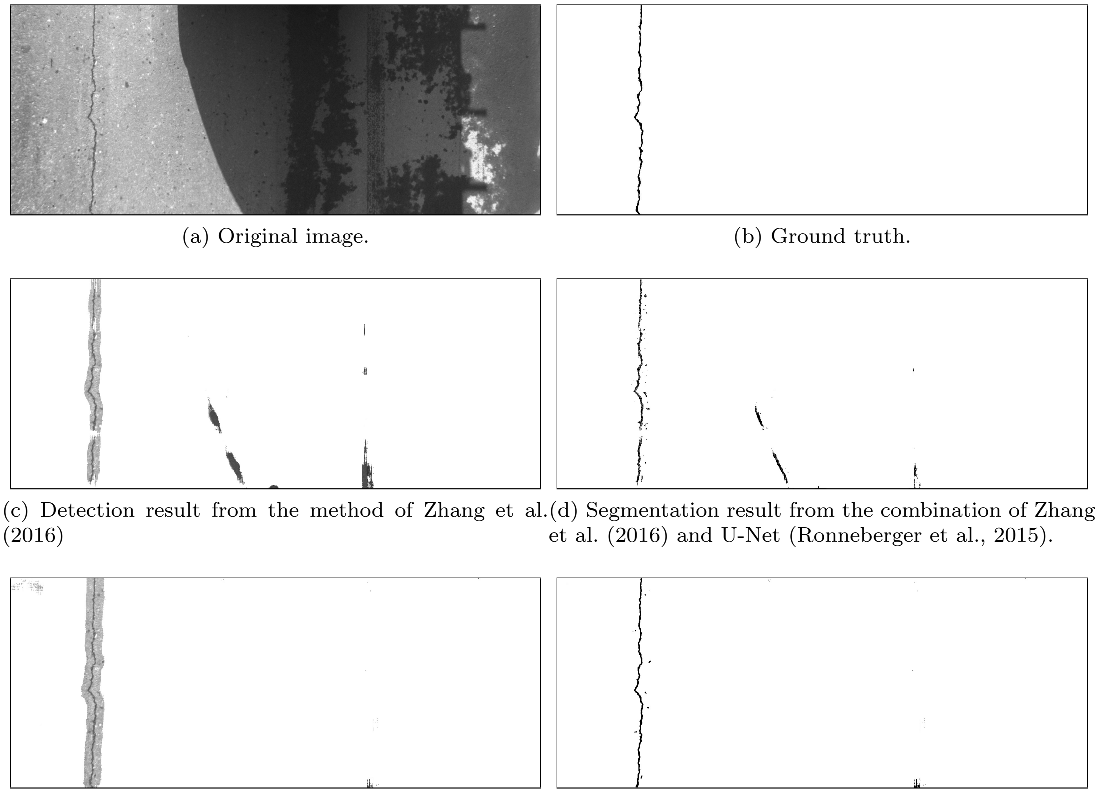
Figure 11 shows an example from the CrackIT dataset with thin cracks with a width of only one or two pixels. The results of Zhang's method shows a number of false positives, and the U-Net model segments many noise pixels as cracks. The proposed method detects and segments the thin cracks with few false positives.

5. Conclusion

This paper proposed a novel two-stage model based on CNN for segmentation at the pixel level of cracks in road images. This method integrates state of the art detection and segmentation into a single unified framework that outperforms existing approaches while significantly reducing the effective imbalance in the data and achieves this with reduced computational requirements.

Experimental results show that our method achieves a higher accuracy than either detection or segmentation alone. The experiments show that the two-stage model works well for noisy, low resolution road images, and imbalanced datasets with artifacts with an F1-score of more than 90% compared to a result of under 80% from other state of the art methods. This architecture may also be considered for other object detection problems that use low quality input data.

For future work, we will adapt our two-stage model for crack detection to other surfaces such as concrete tunnels and bridges, or steel surfaces. More studies should also be attempted to search for better CNN architectures for the proposed model. We also plan to improve our model with the addition of 3D data to create a multi-stage model that is based on CNNs for crack detection and segmentation on three dimensional (3D) images. In addition, future work in this area will involve automating the calculation of the cracking index to avoid the human judgement component of current methods.



(a) Original image.

(b) Ground truth.

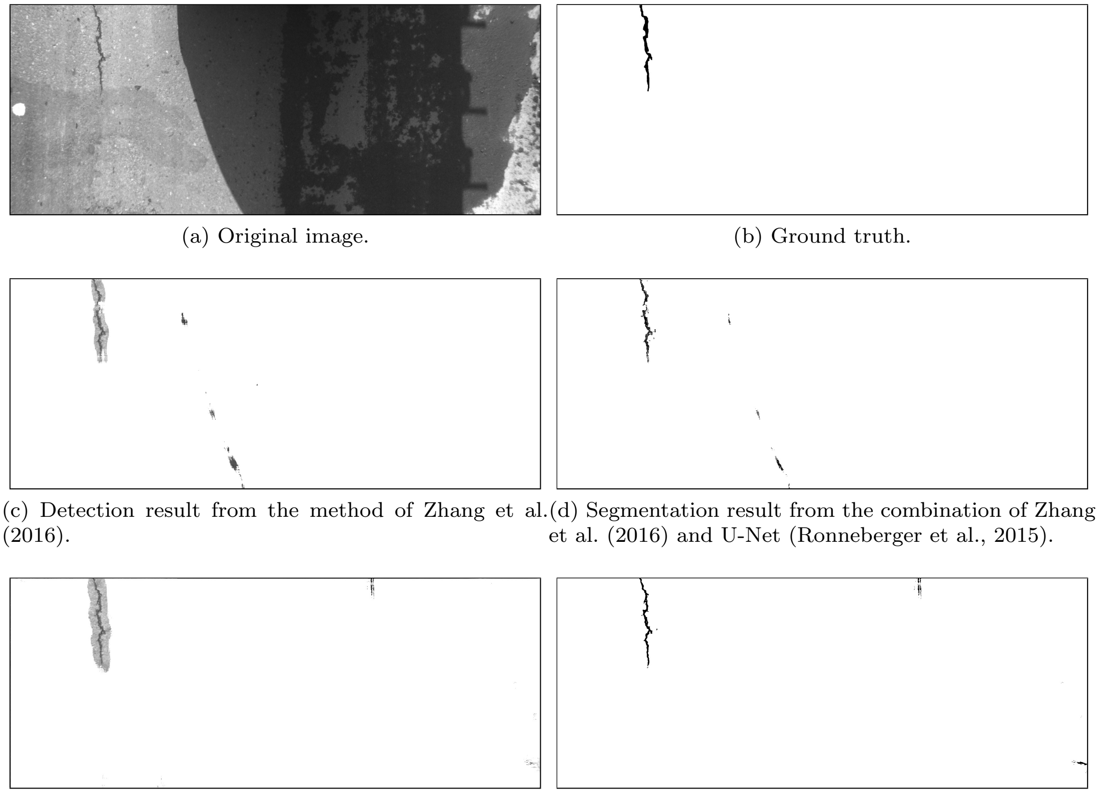
(c) Detection result from the method of Zhang et al. (2016)

(d) Segmentation result from the combination of Zhang et al. (2016) and U-Net (Ronneberger et al., 2015).

(e) Detection result from the first stage of the proposed method.

(f) Final segmentation result from the two-stage model.

Figure 6: Experiment on an image with a long, single crack and a large shadow.



(a) Original image.

(b) Ground truth.

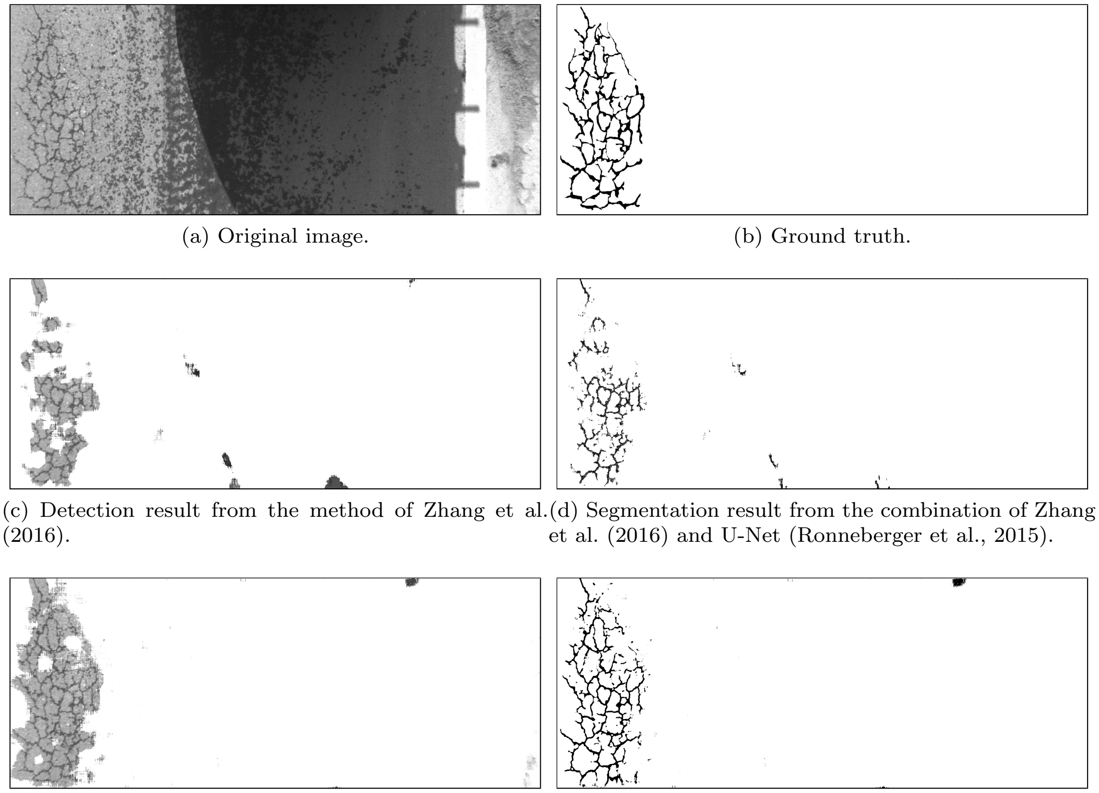
(c) Detection result from the method of Zhang et al. (2016).

(d) Segmentation result from the combination of Zhang et al. (2016) and U-Net (Ronneberger et al., 2015).

(e) Detection result from the first stage of proposed method.

(f) Final segmentation result from the two-stage model.

Figure 7: Experiment on an image with a short, single crack and a large shadow.



(a) Original image.

(b) Ground truth.

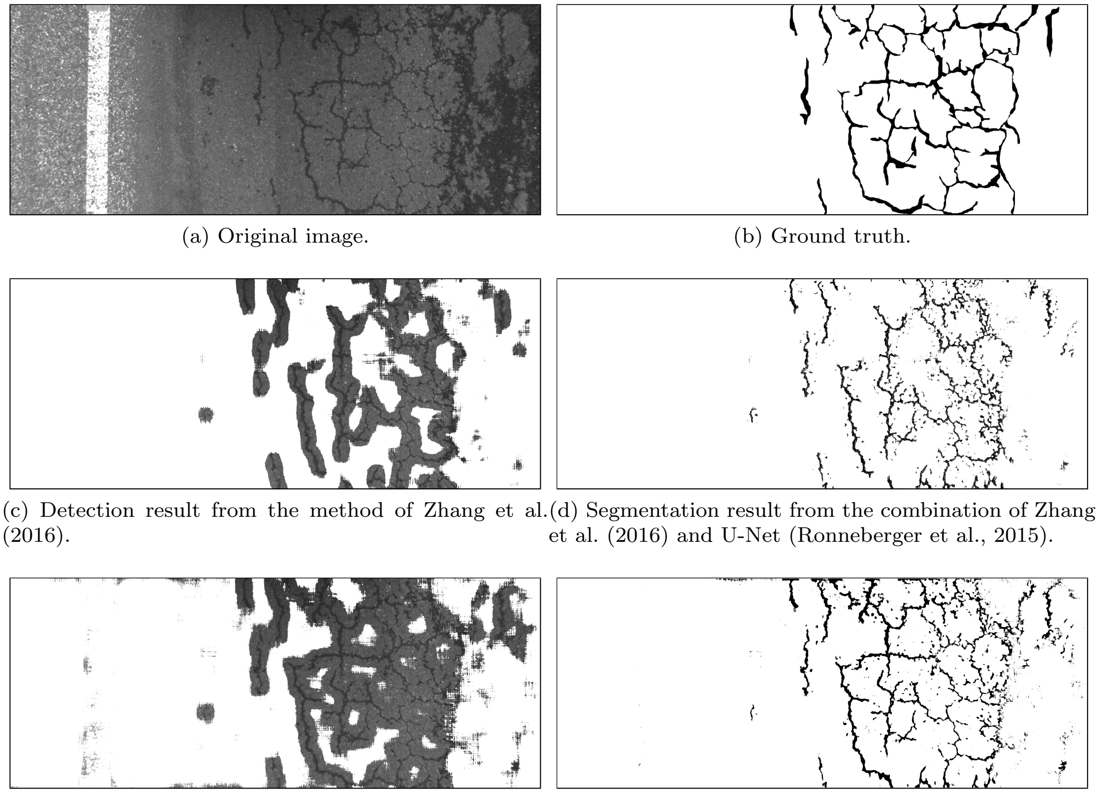
(c) Detection result from the method of Zhang et al. (2016).

(d) Segmentation result from the combination of Zhang et al. (2016) and U-Net (Ronneberger et al., 2015).

(e) Detection result from the first stage of proposed method.

(f) Final segmentation result from the two-stage model.

Figure 8: Experiment on an image with a connected crack on a wet surface with dotty noise.



(a) Original image.

(b) Ground truth.

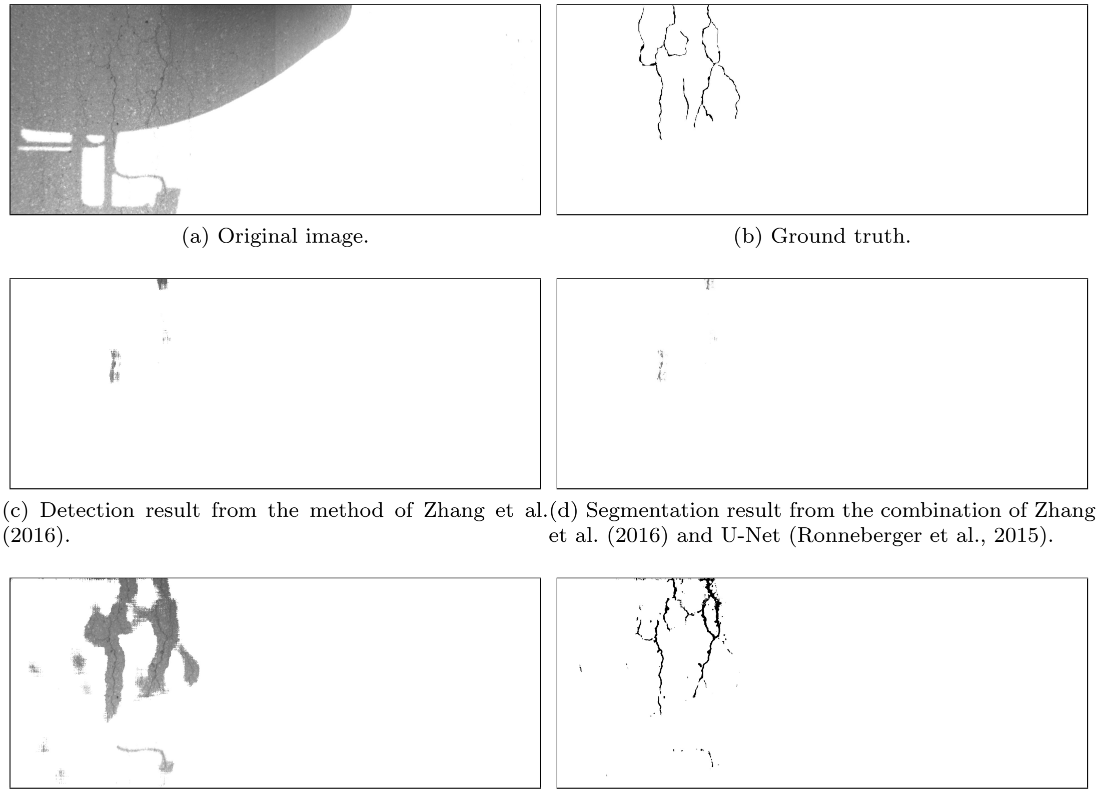
(c) Detection result from the method of Zhang et al. (2016).

(d) Segmentation result from the combination of Zhang et al. (2016) and U-Net (Ronneberger et al., 2015).

(e) Detection result from the first stage of proposed method.

(f) Final segmentation result from the two-stage model.

Figure 9: Experiment on connected, wet cracks captured under weak light conditions.



(a) Original image.

(b) Ground truth.

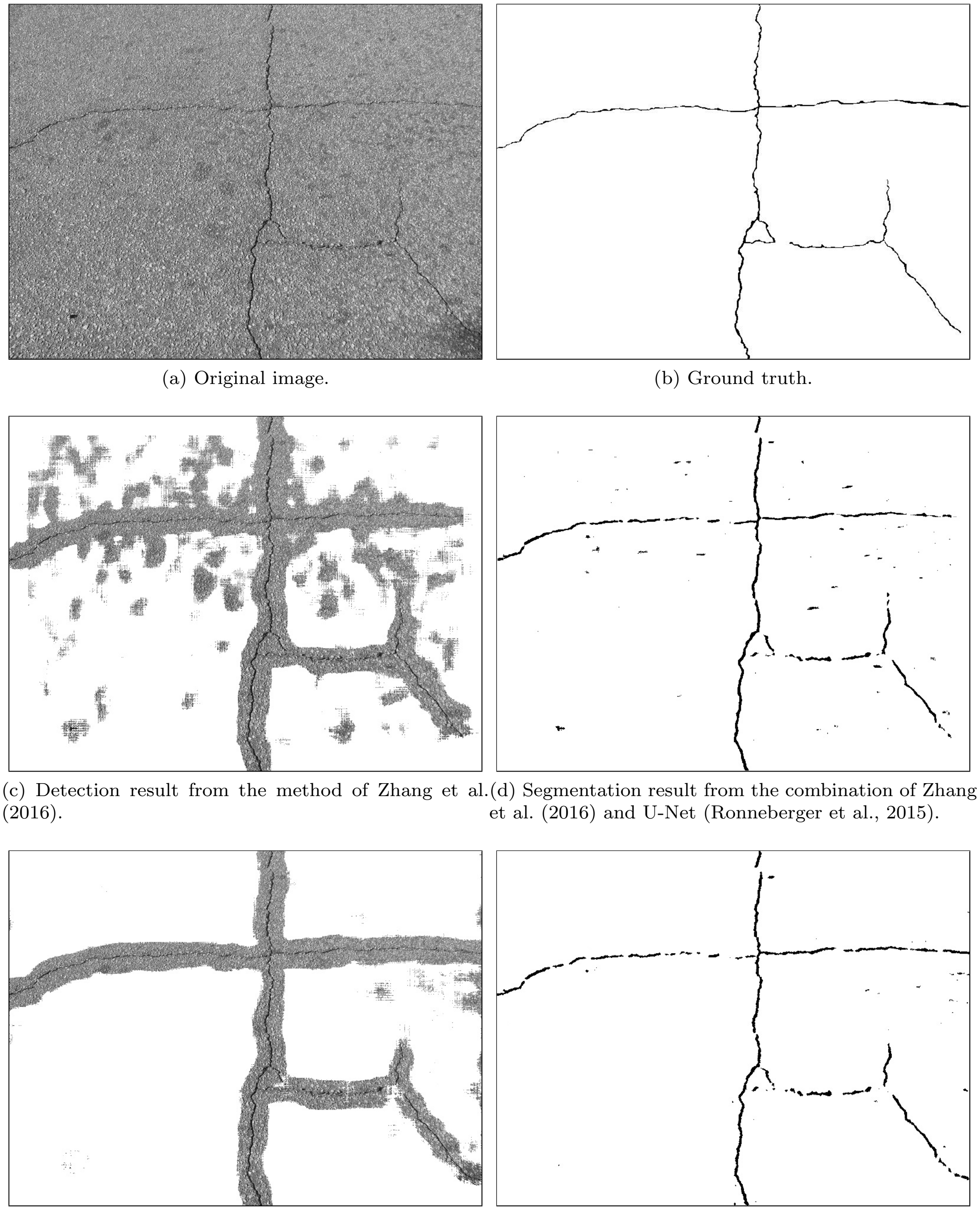
(c) Detection result from the method of Zhang et al. (2016).

(d) Segmentation result from the combination of Zhang et al. (2016) and U-Net (Ronneberger et al., 2015).

(e) Detection result from the first stage of proposed method.

(f) Final segmentation result from the two-stage model.

Figure 10: Experiment on a crack under a shadow.



(a) Original image.

(b) Ground truth.

(c) Detection result from the method of Zhang et al. (2016).

(d) Segmentation result from the combination of Zhang et al. (2016) and U-Net (Ronneberger et al., 2015).

(e) Detection result from the first stage of proposed method.

(f) Final segmentation result from the two-stage model.

Figure 11: Experiment on thin cracks in the CrackIT dataset.

References

L. Ale, N. Zhang, and L. Li. Road damage detection using retinanet. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 5197–5200. IEEE, 2018.

V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

K. Bhoi and D. K. Solanki. *Texture Segmentation Using Optimal Gabor Filter*. PhD thesis, 2011.

D. Bradley and G. Roth. Adaptive thresholding using the integral image. *Journal of graphics tools*, 12(2):13–21, 2007.

J. Bullock, C. Cuesta-Lázaro, and A. Quera-Bofarull. Xnet: a convolutional neural network (cnn) implementation for medical x-ray image segmentation suitable for small datasets. In *Medical Imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging*, volume 10953, page 109531Z. International Society for Optics and Photonics, 2019.

J. Canny. A computational approach to edge detection. In *Readings in Computer Vision*, pages 184–203. Elsevier, 1987.

S. Chambon and J.-M. Moliard. Automatic road pavement assessment with image processing: Review and comparison. *International Journal of Geophysics*, 2011, 2011.

D. Chicco and G. Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):1–13, 2020.

R. Fan, M. J. Bocus, Y. Zhu, J. Jiao, L. Wang, F. Ma, S. Cheng, and M. Liu. Road crack detection using deep convolutional neural network and adaptive thresholding. *arXiv preprint arXiv:1904.08582*, 2019.

Z. Fan, Y. Wu, J. Lu, and W. Li. Automatic pavement crack detection based on structured prediction with the convolutional neural network. *arXiv preprint arXiv:1802.02208*, 2018.

H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1397–1409, 2012.

W. K. Kong, D. Zhang, and W. Li. Palmprint feature extraction using 2-d gabor filters. *Pattern recognition*, 36(10):2339–2347, 2003.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338:139–153, 2019.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

L. Ma, Y. Wang, T. Tan, et al. Iris recognition based on multichannel gabor filtering. In *Proc. Fifth Asian Conf. Computer Vision*, volume 1, pages 279–283, 2002.

H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata. Road damage detection using deep neural networks with images captured through a smartphone. *arXiv preprint arXiv:1801.09454*, 2018.

V. Mandal, L. Uong, and Y. Adu-Gyamfi. Automated road crack detection using deep convolutional neural networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 5212–5215. IEEE, 2018.

B. W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

R. Medina, J. Llamas, E. Zalama, and J. Gómez-García-Bermejo. Enhanced automatic detection of road surface cracks by combining 2d/3d image processing techniques. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 778–782. IEEE, 2014.

A. Mohan and S. Poobal. Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal*, 2017.

J. R. Movellan. Tutorial on gabor filters. *Open Source Document*, 2002.

H. Nguyen, T. Kam, and P. Cheng. Automatic crack detection from 2d images using a crack measure-based b-spline level set model. *Multidimensional Systems and Signal Processing*, 29(1):213–244, 2018a.

N. T. H. Nguyen, T. H. Le, S. Perry, and T. T. Nguyen. Pavement crack detection using convolutional neural network. In *Proceedings of the Ninth International Symposium on Information and Communication Technology*, pages 251–256. ACM, 2018b.

F. Ni, J. Zhang, and Z. Chen. Zernike-moment measurement of thin-crack width in images enabled by dual-scale deep learning. *Computer-Aided Civil and Infrastructure Engineering*, 34(5):367–384, 2019.

H. J. M. Oliveira. *Crack Detection and Characterization in Flexible Road Pavements using Digital Image Processing.* PhD thesis, PhD thesis, Universidade de Lisboa-Instituto Superior Técnico, 2013.

N. Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

S. Park, S. Bang, H. Kim, and H. Kim. Patch-based crack detection in black box images using convolutional neural networks. *Journal of Computing in Civil Engineering*, 33(3):04019017, 2019.

L. E. Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

M. Salman, S. Mathavan, K. Kamal, and M. Rahman. Pavemet crack detection using the gabor filter. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 2039–2044. IEEE, 2013.

M. Y. Shahin and S. D. Kohn. Development of a pavement condition rating procedure for roads, streets, and parking lots. volume i. conditions rating procedure. Technical report, CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL, 1979.

Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

B. Sobol, A. Soloviev, P. Vasiliev, and L. Podkolzina. Deep convolution neural network model in problem of crack segmentation on asphalt images. *Vestnik of Don State Technical University*, 19(1), 2019.

I. The MathWorks. *Image Segmenter, Image Processing and Computer Visions.* Natick, Massachusetts, United State, 2019. URL `https://www.mathworks.com/help/symbolic/`.

H. Zakeri, F. M. Nejad, and A. Fahimifar. Image based techniques for crack detection, classification and quantification in asphalt pavement: a review. *Archives of Computational Methods in Engineering*, 24(4):935–977, 2017.

E. Zalama, J. Gómez-García-Bermejo, R. Medina, and J. Llamas. Road crack detection using visual features extracted by gabor filters. *Computer-Aided Civil and Infrastructure Engineering*, 29(5):342–358, 2014.

L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu. Road crack detection using deep convolutional neural network. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3708–3712. IEEE, 2016.