Smart Structures and Systems, *Vol. 32, No. 3 (2023) 135-151*
https://doi.org/10.12989/sss.2023.32.3.135

A deep and multiscale network for pavement crack detection based on function-specific modules

Guolong Wang [1a], Kelvin C.P. Wang [1b], Allen A. Zhang[*2] and Guangwei Yang [1c]

[1] *Department of Civil and Environmental Engineering, Oklahoma State University, 207 Engineering South, Stillwater, Oklahoma, USA*
[2] *Department of Civil Engineering, Southwest Jiaotong University, No. 111, North 1st Section of Second Ring Road, Chengdu, China*

(*Received December 7, 2021, Revised May 20, 2022, Accepted August 14, 2023*)

Abstract. Using 3D asphalt pavement surface data, a deep and multiscale network named CrackNet-M is proposed in this paper for pixel-level crack detection for improvements in both accuracy and robustness. The CrackNet-M consists of four function-specific architectural modules: a central branch net (CBN), a crack map enhancement (CME) module, three pooling feature pyramids (PFP), and an output layer. The CBN maintains crack boundaries using no pooling reductions throughout all convolutional layers. The CME applies a pooling layer to enhance potential thin cracks for better continuity, consuming no data loss and attenuation when working jointly with CBN. The PFP modules implement direct down-sampling and pyramidal up-sampling with multiscale contexts specifically for the detection of thick cracks and exclusion of non-crack patterns. Finally, the output layer is optimized with a skip layer supervision technique proposed to further improve the network performance. Compared with traditional supervisions, the skip layer supervision brings about not only significant performance gains with respect to both accuracy and robustness but a faster convergence rate. CrackNet-M was trained on a total of 2,500 pixel-wise annotated 3D pavement images and finely scaled with another 200 images with full considerations on accuracy and efficiency. CrackNet-M can potentially achieve crack detection in real-time with a processing speed of 40 ms/image. The experimental results on 500 testing images demonstrate that CrackNet-M can effectively detect both thick and thin cracks from various pavement surfaces with a high level of Precision (94.28%), Recall (93.89%), and F-measure (94.04%). In addition, the proposed CrackNet-M compares favorably to other well-developed networks with respect to the detection of thin cracks as well as the removal of shoulder drop-offs.

Keywords: 3D pavement; crack detection; deep learning; functional-specific modules; real-time processing

1. Introduction

Autonomous crack diagnosis has long been pursued since pavement images in two-dimensional (2D) and three-dimensional (3D) formats were readily available from various automated data acquisition systems, with continuing efforts to improve detection accuracy, efficiency, and repeatability. In recent years, a number of attempts were made utilizing novel Deep-Learning (DL) based Artificial Intelligence (AI) techniques with substantial improvements (Maeda *et al*. 2018, Li *et al*. 2019b, Hsieh and Tsai 2020, Mohammed *et al*. 2020, Munawar *et al*. 2021, Asadi *et al*. 2021). This paper presents further improvements in cracking detection accuracy including detecting fine cracks and removing complex non-crack patterns, which remain challenges for current DL-based approaches.

Over the past decades, traditional methods and machine learning techniques were extensively studied for pavement crack detection (Lee and Lee 2004, Jahanshahi *et al*. 2013, Zalama *et al*. 2014, Daniel and Preeja 2014, Shi *et al*. 2016, Li *et al*. 2019a). However, these approaches were always associated with inconsistent performance in the field due to their inability or inadequate ability to learn from data. Recently, the DL-based techniques using deep convolutional neural networks (CNNs) have demonstrated a more powerful learning approach to pavement crack detection and achieved some success in both research tests and field applications. As per the precision of evaluation results, existing CNN applications on crack detection can be divided broadly into three categories: (1) region-based architectures that detect cracks at a coarse or blocky level (Cha *et al*. 2017, 2018, Ye *et al*. 2019, Hsieh and Tsai 2020); (2) Object detection networks that use bounding boxes of varying sizes to identify the areas consisting of cracks (Maeda *et al*. 2018, Mohammed *et al*. 2020, Du *et al*. 2020); and (3) the architectures for semantic segmentation to detect cracks at a pixel level (Alipour *et al*. 2019, Sathya *et al*. 2020, Munawar *et al*. 2021).

Region-based networks typically implement image classification to inspect whether the whole pavement image is cracked or apply patch classification to search potential crack regions through classifying the background and cracks as a whole target (Ma *et al*. 2017, Cha *et al*. 2017, 2018, Xu *et al*. 2019). Although the classification accuracy

*Corresponding author, Ph.D., Associate Professor,
E-mail: aaanzhang@gmail.com
[a] Ph.D. Student, E-mail: guolong.wang@okstate.edu
[b] Ph.D., Professor, E-mail: kelvin.wang@okstate.edu
[c] Ph.D., Senior Research Engineer,
E-mail: guangwy@okstate.edu

ISSN: 1738-1584 (Print), 1738-1991 (Online)

can be reached at a very high level, the geometric information of cracks with regard to type, length, and width are lost in the classification results. Compared to region classifications, object detection networks provide more flexible classifications. For instance, the faster-RCNN (Gou *et al*. 2019, Mohammed *et al*. 2020), SSD (Maeda *et al*. 2018, Feng *et al*. 2020), YOLO networks (Doshi and Yasin 2020, Du *et al*. 2020) and their variants have been widely used for crack detection and type classification. In these applications, pavement cracks are identified through varying bounding boxes and simultaneously classified into different types (e.g., transverse cracks, longitudinal cracks, block cracks, and alligators, etc.) based on the knowledge learned from manual data. However, the width of cracks was still unknown in the prediction outcomes due to the settings of learning objectives. In general, pixel-level crack detection is preferable in engineering practices with the benefits of estimating the width of cracks, which is a critical indicator for rating pavement conditions.

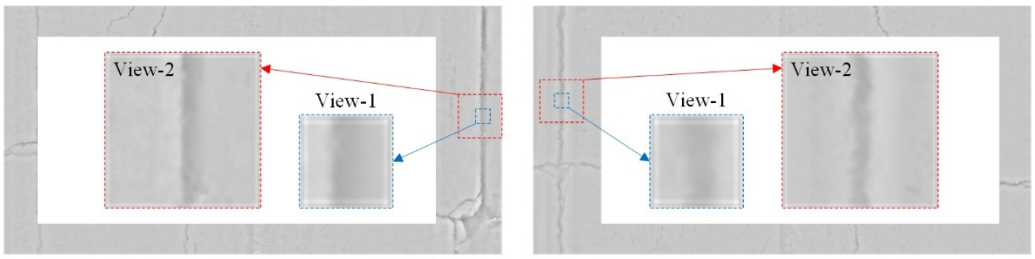
In very recent years, semantic segmentation based on Fully Convolutional Neural Network (FCNN) has become the main approach to achieving crack detection (Hsieh and Tsai 2020, Munawar *et al*. 2021). Unlike CNNs, the FCNNs abandon fully connected layers to conduct per-pixel classification, allowing the width of cracks to be measured more accurately. Based on FCNNs, Crack-Net and its improved version were innovatively proposed for crack semantic segmentation (Zhang *et al*. 2017, 2018), which contains no pooling layers in consideration of data loss. CrackNet consistently demonstrated good precision and bias levels in crack detection, using a set of large filters with a size of 50 × 50 for global context in order to robustly retrieve thick cracks, which, however, causes difficulty in detecting thin cracks due to the attenuation of their weak signals in the forward passes (Zhang *et al*. 2017). Although the improved Crack-Net can better retrieve thin cracks with a sequence of small filters (Zhang *et al*. 2018), some thick cracks are vanished in the final prediction maps due to a lack of valid global context. Such similar observation was also verified by another study that intended to design the net's receptive fields based on the same rationales (Fei *et al*. 2019).

Subsequently, many other FCNN-based networks were also propounded for crack segmentation. For example, using transposed convolutions, Alipour *et al*. (2019) modified VGG16 into a FCNN for the sake of robust pixel-level crack detection. U-net and its variants were proposed as typical FCNNs to detect pavement cracks at a pixel level from images (Escalona *et al*. 2019, Sizyakin *et al*. 2020). Such FCNNs as encoder-decoders were implemented for crack detection from black-box pavement images (Bang *et al*. 2019) and structured for detecting multiple types of impairments on concrete surfaces with the strict objective of pixel-level performance (Li *et al*. 2019b). By means of a patch-based training procedure, König *et al*. (2019) combined the attention gating mechanisms into a U-Net to achieve pixel-level crack detection under the condition of small training datasets. Also, the generative adversarial networks that adopt the FCNN and CNN as the backbone networks were applied to improve the resolution of pavement image for more accurate segmentation (Sathya *et al*. 2020) or to address the deficiency and imbalance inherent in the crack training data (Gao *et al*. 2020).

Although CNNs or FCNNs may show good precision and bias levels using private data, the imperfection of initial design in CNNs is neglected in current studies. Normally, pooling layers are applied in these FCNNs to increase the receptive fields at the sacrifice of losing spatial relations of initially indexed pixels. To localize cracks precisely, the lost signals in pooling layers are compensated at each scale with additional operations, such as skip connections, up-sampling using encoder pooling indices, and the concatenation of early layers to late layers. However, at least one of the three major imperfections are exposed that may affect final task performance. First, the abstraction level at input image resolution is shallow for those thin cracks of weak signals that are also faced with the problem of attenuations, causing difficulties in detecting them. Second, large-scale learning occurring at the bottleneck of the encoder-decoder is too inefficient with redundant parameters in the encoder to distinguish cracks desirably from other complex non-crack patterns, such as discussed shoulder drop-offs. Lastly, predicting crack maps using the highest-level features of the last layer alone or along with the lowest-level features of the first layer would lead to suboptimal performance (Hwang and Liu 2015).

Further, the longitudinal shoulder drop-offs, which indicate the elevation changes between a travel lane and its adjacent shoulder and should not be cracks, were frequently misclassified by previous algorithms due to the similarities to longitudinal cracks (Zhang *et al*. 2017, 2018). In a typical CNN, the size of the receptive field referring to the region in the input data space plays a crucial role in distinguishing one specific input feature from other ones. Fig. 1 shows that the longitudinal pavement crack pattern is barely distinguishable from the longitudinal pavement shoulder drop-off at a small receptive field (View-1). However, noticeable distinctions could be observed under a larger receptive field (View-2) where the edges of the shoulder drop-off are straighter than those of the ragged longitudinal cracks. In addition, the two patterns may also be different at the depths. One of the biggest challenges for crack detection based on FCNNs is not only to detect both thin and thick cracks accurately but to suppress any other non-crack patterns. The former task calls for small or medium size of receptive fields to avoid the attenuation of crack signals in deep transforms, whereas the latter mission requires large ones to reduce possible ambiguities among cracks and various noise patterns. To this end, this paper was also conducted to remove these longitudinal shoulder drop-offs at multiple scales using special architectural strategies.

Therefore, a novel robust framework based on FCNN is developed for automatic crack detection on 3D asphalt pavement surfaces, with explicit considerations on detecting thin and thick cracks as well as the removal of shoulder drop-offs which were weaknesses of prior studies. The framework termed CrackNet-M is configured with four genres of architectural modules, which function individually and collectively to ameliorate the three



Shoulder drop-off at right side of the pavement image Longitudinal crack at left side of the pavement image

Fig. 1 Illustration of visual effects of a noise pattern and a similar crack under different perception views

imperfections. The innovations of this paper include a new Deep-Learning network with four primary components or modules:

- A central branch net (CBN) module is organized to deepen the levels of feature delineation at the original dimensions for maintaining the accurate boundaries of cracks. In addition, CBN also serves as the backbone to associate the other three modules together.
- A crack map enhancement (CME) module is constituted with a pooling layer for enhancing the continuity of thin cracks. The CME module in conjunction with CBN can ensure the pixel-perfect accuracy of the framework particularly for the detection of thin cracks.
- Three pooling feature pyramids (PFP) are established as efficient encoder-decoders for the detection of thick cracks and exclusion of non-crack patterns based on efficient multiscale perceptions that reduce possible ambiguities within a local context.
- A skip layer supervision technique is proposed to further improve the network performance in respect of accuracy and robustness by connecting the output layer to the intermediate layers for per-pixel supervision based on the ground truths.

In view of network efficiency, only small filters with a size of 3 × 3 or 1 × 1 are adopted through all convolutional layers in spite of data scales. Finally, the four modules except the output layer are scaled to use the minimum number of parameters to produce the best outcomes, which achieves a real-time pavement distress survey in the field.

2. Data bank

The research team of the paper prepared a rich data bank specifically for the CrackNet-M development. At present, more than 6,500 3D pavement images collected from various types of pavements and corresponding ground truths have been continuously added to the data library in the past six years. All the raw pavement images in the data bank are 1-mm 3D data acquired by the PaveVision3D system of WayLink Systems Corporation, which scans the full 4-m-wide pavement lane at the highway collection speed up to 60 MPH (Zhang *et al*. 2017). Although the latest iteration of WayLink's 3D laser imaging technology has evolved to be able to collect the 0.5-mm (8K) data (Wang *et al*. 2022), the CrackNet-M trained with 1-mm resolution samples is still applicable as long as the input images are resized to the same dimensions. The ground-truth images were carefully annotated based on the raw images using a one-pixel-width pen and further inspected by multiple groups of technicians with strict considerations on pixel-perfect accuracy.

The 3D pavement images are saved with a dimension of 2,048 mm in length and 4,096 mm in width. For the network to be trainable on limited hardware resources, the input images are downsized from the original resolution to the dimensions of 256 × 512 by a min-pooling technique, which outputs the minimum value within an 8 × 8 sliding window. The min-pooling approach is assumed to retain any likely crack pixel that takes a lower elevation compared to its surrounding regions. Despite the significant reduction of original image resolution, the downsized image does not cause visible blur or discontinuity on both thick and thin cracks in contrast to the original image. The richness, diversity, and balance of the training image pairs, which are beneficial for improving the network performance, are considered in a way that selects no more than 100 non-overlap images from the same pavement section. Finally, a total of 3,200 3D images that contain various types of cracks (including thin, thick, horizontal, longitudinal, and intersected cracks) and non-crack patterns (including shoulder drop-offs) are selected as the data bank for CrackNet-M development. Then, the 3,200 3D images are randomly separated into three sets: 2,500, 200, and 500 image pairs for supervised training, validation, and testing, respectively. Fig. 2 displays several representative 3D pavement images and corresponding binary ground truths from the data bank.

3. Architecture

The overall scheme of the CrackNet-M for crack detection is illustrated in Fig. 3. It consists of four major architectural modules that aim at tackling challenges from specific perspectives and in an organized manner: the central branch net (CBN), the crack map enhancement (CME) module, the pooling feature pyramids (PFP), and the prediction layer. The CBN module undertakes the paramount role of coordinating the other three modules to work in synergy with each other so as to address different

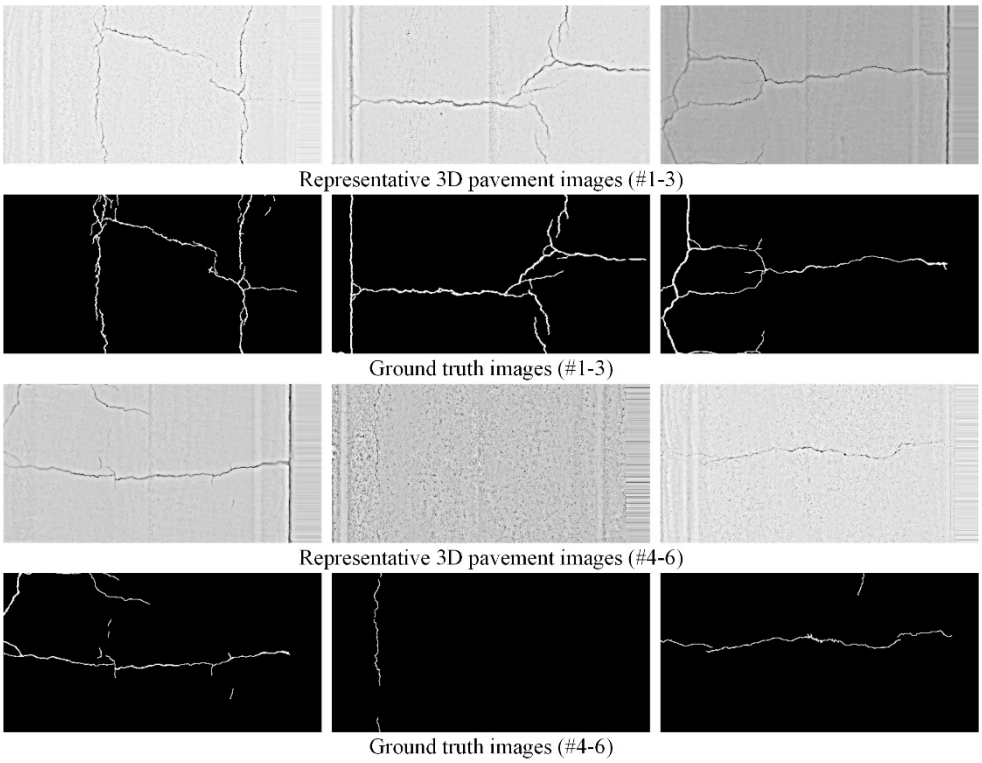


Fig. 2 Representative 3D images and ground truths from the data bank

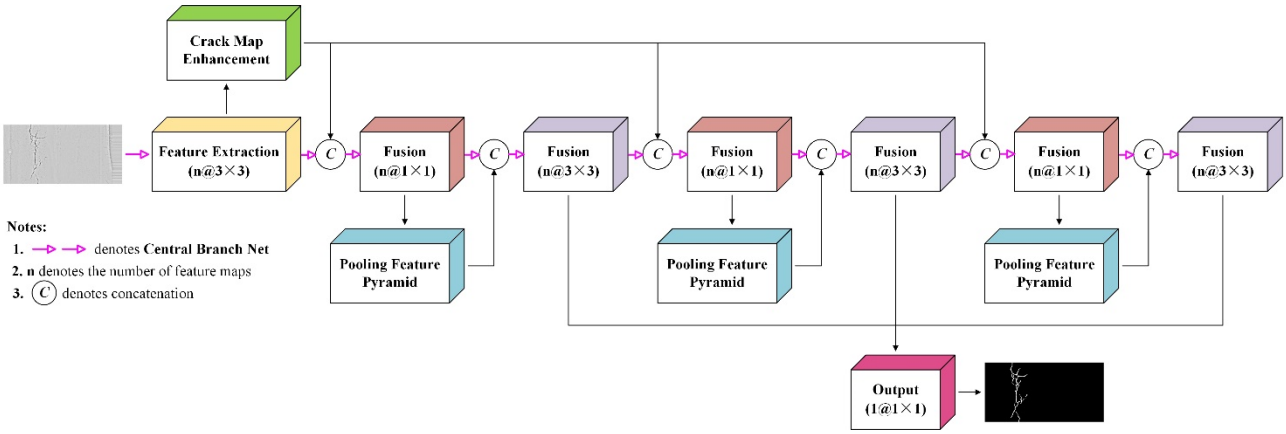


Fig. 3 The architecture of CrackNet-M

problems simultaneously. The other three modules individually with different contributions to the prediction, and there is no direct connection among them. The CBN, CME, and the prediction layer are arranged mainly for detecting hairline cracks, whereas the multiscale PFP in combination with CBN and the prediction layer aims for the segmentation of thick cracks and the erasion of non-crack patterns. By training these modules together, the final probability map for cracks of various topology shapes can be significantly enhanced with pixel-perfect accuracy.

3.1 Crack map enhancement

The 3D pavement data are constructed by collecting a massive number of deformed laser lines on the pavement surface, where the optical cameras in the 3D sensors are angled to capture the variations or deformed lines of the laser. Those thin cracks collected are always weak in relation to both depth and width, which poses a huge challenge for the network to detect them without special treatments, especially when the network learning is substantially dominated by thick cracks. Therefore, increasing either the depth or width of thin cracks is able to enhance their detection accuracy.

In this paper, width enhancement for thin cracks is conducted by using a max-pooling layer due to its eminent compatibility and efficiency for network learning. In many CNN-based classification tasks, the max-pooling layer is a standard component for dimensionality reduction while extracting dominant features that are rotation- and position-

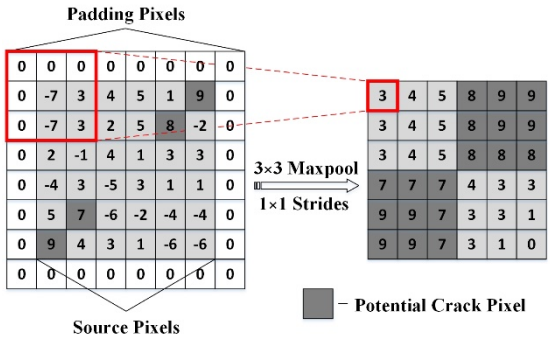


Fig. 4 Illustration of max-pooling enhancement in CME module

invariant when the pooling unit is sliding with a large stride. However, pixel-precision crack classification requires the exact location of each pixel to be strictly preserved without any alterations. Motivated by GoogLeNet (Szegedy *et al.* 2014), a 3×3 max-pooling layer with a stride of {1} is adopted to adjust the widths of cracks on the original image resolution, which reports the maximum depth within a rectangle neighborhood.

It is shown in Fig. 4 that the max-pooling function can enhance the thin cracks in two aspects without downsizing the input data. First, it dilates all the potential crack patterns in width with two pixels. Second, it connects the ends of the two nearest crack patterns by filling the gaps between them with at most two pixels. To ensure the functionality of max-pooling, the pavement rutting or slopes that possibly exist in the input image are firstly removed based on the average filtering technique and then standardized before it is fed into the network. However, the standardization or Z-score used in this study is the negative of the normal one (Carre *et al.* 2020). The negative standardization converts the negative values of crack pixels in the rectified image into positives. Therefore, it is highly likely that the outputs of these crack pixels are still positive after convolutions of normalized zero-mean weights, which makes the ReLu activation function and the CME module work properly. Thus, the standardization function used in this study can be written as

$$\hat{x} = -1 \times \frac{x - \bar{x}}{\max\left(\sigma(x), \frac{1}{\sqrt{n}}\right)} \tag{1}$$

where x, $\bar{x}$, and $\hat{x}$ are the input image, mean of the input image, and standardized image, respectively. $\sigma(x)$ is the standard deviation of the input image. n is the number of pixels of the input image.

The max-pooling layer is trailed behind the feature extraction layer instead of behind the input image to increase the adaptability of enhancement. The same enhanced feature maps will be fed along with three different feature maps into three 1 × 1 convolutional layers in CBN for feature fusions and corrections. As shown in Fig. 3, the first 1 × 1 fusion layer connects with the feature extraction layer while the other two link to the 3 × 3 fusion layers. The concatenations and fusions of enhanced feature maps to the three intermediate maps can avoid the distortion by the max-pooling layer, and also circumvent the attenuation of enhanced crack signals in deep transforms, by which the detection of thin cracks is finally improved.

3.2 Pooling feature pyramid

The pooling feature pyramid (PFP) is well-elaborated to particularly recognize thick cracks via conducting feature learning at multiscale receptive fields. Unlike traditional encoder-decoder, the PFP module implements direct multi-stream down-sampling and pyramidal up-sampling for multiscale feature formations. The PFP module is rationalized by the pyramid scene parsing network (PSPNet) (Zhao *et al.* 2017) and the feature pyramid network (FPN) (Lin *et al.* 2016, Guo *et al.* 2019). Analogous to the pyramid pooling module used in PSPNet, the proposed PFP employs multi-parallel pooling layers with different strides identical to corresponding pooling size to exploit global scene category clues. As shown in Fig. 5, the same input data are divided into five contextual maps

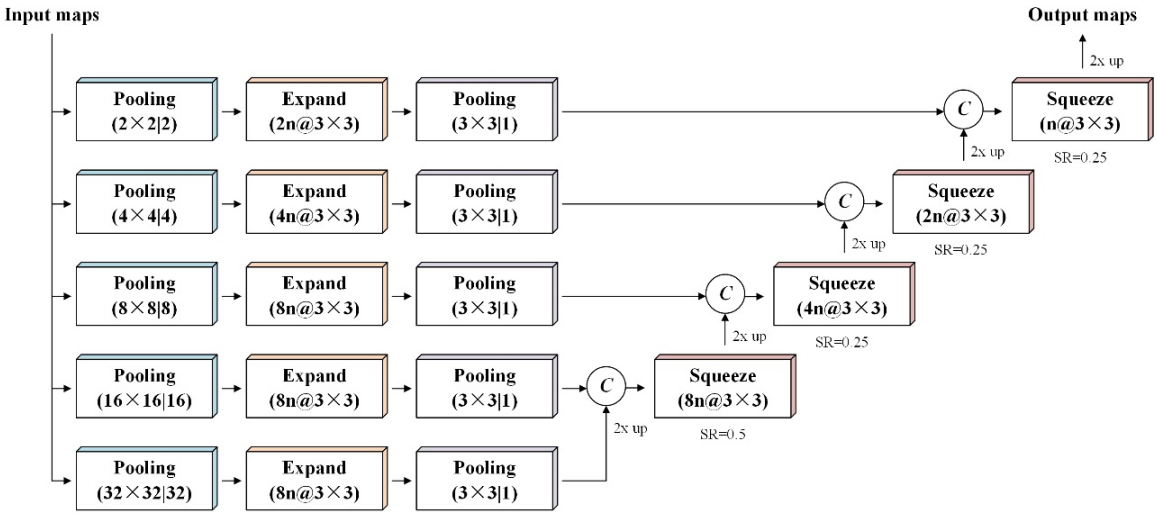


Fig. 5 The architecture of pooling feature pyramid (PFP)

using five parallel pooling layers with a stride of {2, 4, 8, 16, 32} pixels from top to bottom. Compared with the encoder-decoders that progressively downsize the data dimension with extensive parameters for global context, the proposed reduction technique is more straightforward and efficient since there are no extra parameters to be learned in a pooling layer. On the other hand, shrinking the data from the very start of the resolution may also produce more salient contexts, preventing severe motif attenuation that occurs typically in a deep encoder.

Average pooling that equally weights the elevation values within the rectangle region is adopted as the reduction technique for noise suppression and information preservation. The pooled feature maps of all five levels are then expanded using 3 × 3 filters in order to capture diverse crack patterns (Iandola *et al*. 2016). The expansion ratio of each level is shown in Fig. 5. With the usage of 3 × 3 filters, multiscale receptive fields are established with sizes of 96 × 96, 48 × 48, 24 × 24, 12 × 12, and 6 × 6 from the bottommost level to the top level. The extension of 96 × 96 receptive fields at the bottommost layer should be large enough for a network to distinguish cracks from textures or other complex non-crack patterns. Considering the effect of CME module may be limited on the finest scale, the 3 × 3 dimension-invariant max-pooling is also adopted after all the expand layers to further enhance the crack patterns at each level of coarser scale. The signal losses caused by pooling layers are trivial for the PFP as it intends to develop crack patterns rather than boundaries.

Hierarchical feature pyramid is adopted in PFP for pyramidal or step-wise up-sampling to fuse local and global features from different contextual streams at the same pixel location. The feature pyramid is a top-down architecture with lateral concatenations between two nearest levels where the predictions can be made on the finest level. The concatenations between two levels are squeezed or merged through the small filters in size of 3 × 3 for feature fusions or corrections, and at the same time to reduce the number of parameters (Iandola *et al*. 2016). The squeeze ratios (SR) for every two nearest levels are specified to be 0.5, 0.25, 0.25, and 0.25 from bottom to top, which are shown in Fig. 5. In a broad sense, the PFP module can be regarded as an efficient encoder-decoder with purpose-specific functions. The number of PFP encoder-decoders is designated to be three in this study for the sake of multiscale semantics at low, mid-, and high levels. For computation efficiency, the low-dimension data maps are up-sampled at each horizontal level via the simplest nearest-neighbor interpolation method. Using a combination of techniques, including direct down-sampling, expansions, enhancement, squeezes, and incremental up-sampling, a global-to-local or coarse-to-fine semantic context is formulated to be favorable for the detection of thick cracks and removal of non-crack patterns.

3.3 Central branch net

The CBN module connected by the horizontal hollow arrows comprises seven conservative convolutional layers. To ensure the spatial relation at each pixel location to be the same as that in the input image, no strided convolution or pooling reduction is executed in this module. Thus, each point response across all the channels and levels is definitely generated at the same location, which signifies a unique feature aspect of each pixel. Different from PFP or CME, the CBN module is developed to emphasize accurate edges. Given that the weak features of thin or hairline cracks are difficult to be recognized by large filters, only those filters with a receptive field size of 3 × 3 or 1 × 1 are applied through all the fusion layers, which also induces efficient computation. Three 3 × 3 convolutional layers fuse the local-scale features from CBN and the multi-large-scale features from PFPs, whereas the 1 × 1 convolutional layers fuse the enhanced features from CME and the lowest-level CBN features or the multi-large-scale PFP features. The fusions of the CBN and the other two modules are for two purposes. First, the CME and PFP modules are supposed to correct crack misclassifications caused by CBN via introducing pattern features of high confidence as well as suppressing the noises in the background. Second, the crack patterns distorted by CME or PFP could be intelligently trimmed through the parameters of fusion layers. After CBN finalizes all the fusions, the original topology of the image at each pixel is described partially by a sequence of point values located at each channel and each layer level.

3.4 Output layer

Output layer is intended to estimate the probability of each pixel being an element of cracks in reliance on the local features of the nearest layer to the output layer. In general, a layer connected to the output layer contributes directly to the performance of crack detection. Inspired by the principle of ResNet (He *et al*. 2015), a skip layer supervision technique is developed in the paper to achieve a high confidence level of crack class predictions. The skip layer supervision is established among the output layer and intermediate layers and rather than only the last layer such that each intermediate layer has a direct contribution in predicting crack probabilities. Compared with traditional networks, the CrackNet-M trained with intermediate layers exhibits at least two benefits. First, it can yield better prediction results as different levels and scales of crack features at intermediate layers are thoroughly investigated by the output layer with learnable parameters. Second, it has a faster convergence rate during back-propagation due to the skip connections of early intermediate layers to the output layer, by which a portion of gradients is transmitted directly to the beginning. Furthermore, the updated parameters in the early layers also facilitate the learning of late layers under the supervision of the remaining portions of gradients.

As is sketched in Fig. 3, the output layer is naturally attached to the 3 × 3 fusion layers that contain enhanced local and global features. The feature maps of these fusion layers are identical to the input image and ground-truth image in terms of width and height. Consequently, the precise pixel-to-pixel correspondence is built using the 1 × 1 convolution that weights the point features of each pixel. Here, the three feature maps produced from the 3 × 3 fusion layers are denoted as $x = (x_1, x_2, x_3)$, which corresponds

to the output layer weights $w = (w_1, w_2, w_3, b)$. Therefore, the probability map obtained from the output layer can be expressed as:

$$p = sigmoid(\sum_{i=1}^{3} w_i x_i + b) \quad (2)$$

where p represents the predicted probability map. w_i is the vector with the same number of weighting values as the channels of i_{th} fusion maps. The scalar b is the bias of the output layer.

4. Training

4.1 Loss function

In a typical cracked pavement image, the distribution of crack pixels (i.e., hard positive examples) is highly biased compared to background pixels (i.e., easy negative examples), in which the vast majority of the ground-truth pixels are easy negatives. This severe class imbalance results in two problems: (1) training is inefficient as only a few locations are positives that contribute little useful learning signal; (2) the network trained with unbalanced samples is always associated with degraded performance due to loss of hard positives. To mitigate the adverse impacts of class imbalance on detection accuracy, a focal loss function is adopted in place of traditional binary cross-entropy on a per-pixel term basis to measure the dissimilarities between predictions and ground truths (Lin *et al*. 2017). The focal loss function modifies the cross-entropy by scaling it with a factor that adaptively down-weights the importance of easy examples based on currently predicted probabilities. As a result, the contribution of easy examples will be small even though their number is large. According to the definition of focal loss, an α-balanced variant of the focal loss function is used in the article as

$$FL = -\alpha \sum_{i=1}^{N} \left(1 - \hat{p}(i)\right)^{\gamma} \log\left(\hat{p}(i)\right) \quad (3)$$

where the $\hat{p}(i)$ equals to either the predicted probability $p(i)$ at the pixel i from the probability map p if the target label is "1" or the $1 - p(i)$ if the label is "0". $\alpha = 0.25$ is a class-balancing factor. N is the total number of samples or pixels in a training batch set. $\gamma = 2.0$ is the focusing parameter that controls the rate at which the easy examples are down-weighted.

4.2 Training method

The objective of training CrackNet-M is to minimize the loss error all the way back to the beginning in accordance with an appropriate back-propagation method. The Adam optimizer (Kingma and Ba 2015), a variant of Stochastic Gradient Descent (SGD) for training neural networks, is implemented along with mini-batch training to optimize the network parameters. In comparison with SGD method, the Adam optimizer that combines the ideas of RMSProp and Momentum can update the parameters more smoothly with a sufficient convergence speed, regardless of noisy or sparse gradients.

4.3 Training configuration

To reduce training difficulties, the parameters of convolutional layers are all initialized by Glorot Normal Initializer (Glorot and Bengio 2010), and the Rectified Linear Unit (ReLu) function is adopted after each convolutional layer for nonlinear transitions. The number "n" for counting the feature maps in the overall architecture (Fig. 3) is optimally determined by the Scaling Method that allows for both accuracies and efficiencies (Tan and Le 2019). The architecture configured with n = 2 is used as the basis for optimal number searching by adjusting the network width (i.e., channel) scaling factor w. The process of scaling CrackNet-M can be formulated as

$$w^* = \text{argmax}(\text{F}\left(\bigotimes_{i=1,2,\ldots,s} f_i\left(X_{<\hat{H}_i, \hat{W}_i, w, \hat{C}_i>}\right)\right)) \quad (4)$$

subject to

$$\sum\nolimits_{\Theta} \left(\bigotimes_{i=1,2,\ldots,s} f_i\left(X_{<\hat{H}_i, \hat{W}_i, w, \hat{C}_i>}\right)\right) \leq 1M \quad (5)$$

where $< \hat{H}_i, \hat{W}_i, \hat{C}_i >$ denotes the shape of input tensor X at the layer i for the base architecture (n = 2). $\otimes f$ denotes the architecture stacked with a total of s layers. F(·) denotes the F-measure evaluation. w^* signifies the optimal width scaling factor that yields the best F-measure. $\sum_{\Theta} \otimes f$ signifies the total number of parameters of an architecture, which is fixed to be less than 1 million.

To manifest the effectiveness of skip layer supervision with focal loss, CrackNet-M is considered in three other training scenarios. The three composite scenarios are: (a) single layer supervision using cross-entropy, which is denoted as CrackNet-M-SGL-CE; (b) single layer supervision using focal loss, which is denoted as CrackNet-M-SGL-FL; (c) skip layer supervision using cross-entropy, which is denoted as CrackNet-M-SKP-CE. Additionally, CrackNet-M shall be partially altered and retrained for a comparison with the original one to verify the advantage of the PFP module in detecting cracks over the PSP scheme implemented in PSPNet. A count of 2,500 image pairs is randomly selected from the prepared 3,200 pavement images to develop models. Each input image is flipped horizontally, vertically, or kept the same with an equal chance to further combat overfitting risks in addition to the L1 penalty.

Due to the existence of scaling factors, the focal loss error in magnitude is much less than cross-entropy. To achieve similar convergence rates, the initial learning rate for the models trained with focal loss is specified to be 0.003, while the models supervised with cross-entropy have an initial learning rate of 0.001. Afterwards, the learning rate will decrease exponentially with a factor of 0.98 if the

validation accuracy is no longer improved after 400 iterations of learning. The parameters of models are updated by the average mini-batch gradients computed from the loss errors of 16 pavement images at each iteration. The learned parameters are saved for every 20 iterations, and meanwhile, 200 additional pavement images that are not involved in the training process are evaluated by these parameters for overfitting inspection.

Precision, Recall, and F-measure are used as the metrics for performance evaluation, which together can provide an objective and comprehensive comparison among different models (Fawcett 2006). Precision denotes the percentage of True Positives with respect to the total predicted Positives, which is a good measure when cracks detected erroneously with high False Positives are more intolerant. Recall signifies the percentage of True Positives with reference to the total True Positives, which is a good metric when cracks detected correctly with low False Negatives are more important. While a high F-measure that is the harmonic mean of Precision and Recall requires them to be both high and should be a primary metric for evaluating network performance.

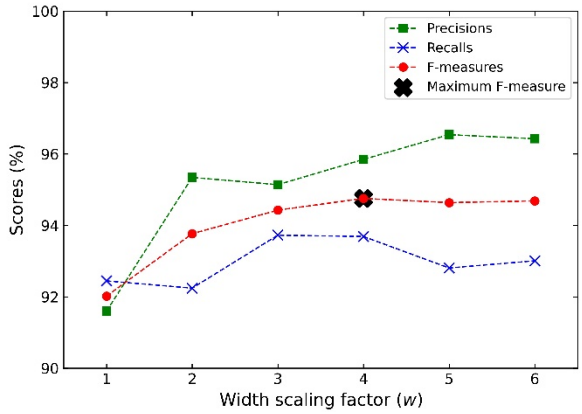


Fig. 6 Validation results of architectures using different width scaling factors

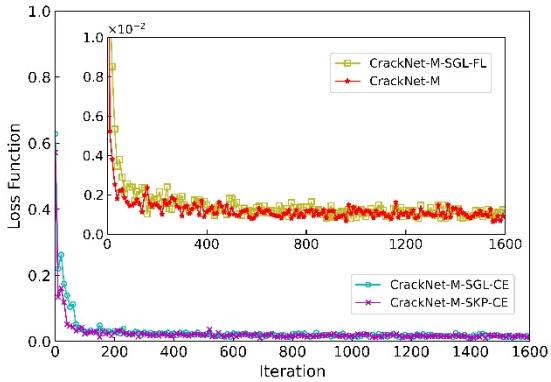
4.4 Training results

All the described models are implemented with Python and TensorFlow in the same workstation configured with a high-core count AMD Ryzen Threadripper CPU and a Nvidia GeForce GTX 1080 Ti GPU. The training for all scenarios is terminated after 1600 iterations without any human interference during this period, which approximately amounts to 10 epochs of data learning. The training of each scenario took no more than half an hour due to the GPU acceleration.

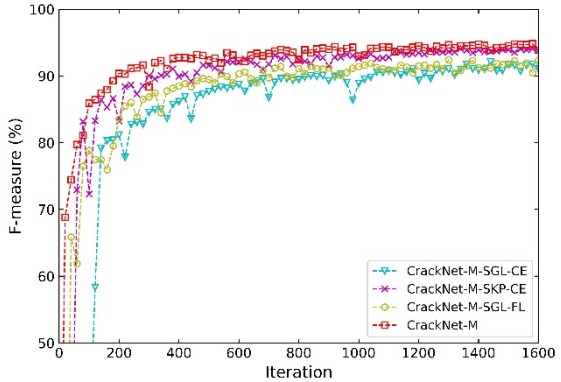
According to Eqs. (4)-(5), six architectures with different width scaling factors are trained and evaluated by means of Precisions, Recalls, and F-measures on validation images, which are shown in Fig. 6. It can be seen that the F-measure grows appreciably as the scaling factor increases and then reaches the performance bottleneck. The highest F-measure is observed at $w = 4$ with a score of 94.75%. Therefore, the architecture with n = 8 and the parameters of the best validation F-measure is finally saved as the CrackNet-M. Scaling Method enables CrackNet-M to produce the greatest results in an efficient way with the minimum number of parameters.

Fig. 7 illustrates the training results of three comparative scenarios and CrackNet-M with respect to the losses, and F-measures. The two types of loss function values over the iterations are shown in two separate windows for better views. It is obvious that the loss values for all scenarios are reduced significantly after 1000 iterations and oscillate steadily within small ranges for both of the two loss functions, implying that the four architectures have been trained successfully. The validation results of F-measures suggest that the architectures trained with focal loss exhibit comparably higher F-measures, in comparison with the architectures trained by cross-entropy loss. Such finding indicates that focal loss can indeed drive the network to learn more signals of hard positive examples with faster speeds because of the adaptive scaling factor. In other words, a high level of detection accuracy without much crack loss can be achieved by learning a focal loss function, which is crucial to prevent the underestimation of the distress severity of a pavement.

The skip layer supervision significantly elevates the



(a) Loss function walues



(b) Overall F-measures for Validation Images

Fig. 7 Illustration of training details under different scenarios

performance of CrackNet-M to a new high level irrespective of which type of loss function is used, according to the evaluation results of F-measures. Moreover, those architectures trained using skip layer supervision have a faster convergence rate and more smooth prediction curves in terms of F-measures, revealing the robustness of predictions by these architectures. CrackNet-M that is trained with skip layer supervision and focal loss demonstrates the best F-measures. The two noticeable performance gains of CrackNet-M with respect to the accuracy and robustness can be substantially attributed to the skip connections of intermediate layers to the output layer. Skip connections allow an early intermediate layer to directly contribute to the predictions by skipping its after-intermediate layers. In a broad sense, a single last layer represents only the highest levels of crack features on which the prediction is often biased. This explains why the inferior performances are observed for the scenarios of single layer supervision even though they look no significant difference in the loss values with the counterparts.

However, the each intermediate 3 × 3 fusion layer in CBN is the fusion of local-scale maps from CBN and multi-large-scale maps from PFP modules. The accuracies of crack detection can thus be improved by fusing these intermediate layers at low, mid-, and high levels. The decent robustness of predictions is also a product of this fusion process since the misclassifications occurring at a specific intermediate layer can be corrected through other normal intermediate layers with self-regulation benefits. In addition, skip connections create shortcut paths for gradients to propagate back to the beginning, which thereby results in a fast convergence rate. Table 1 summarizes the number of parameters of CrackNet-M in different modules. The CrackNet-M has a total number of 423,249 parameters, with which the average Precision, Recall, and F-measure evaluated on the 2,500 training images are 96.00%, 93.09%, and 94.52%, respectively.

Table 1 The number of parameters of CrackNet-M

Module	Number of parameters (including biases)
CBN	3,968
CME	0
PFPs	139,752× 3 = 419,256
Output layer	25
Total	423,249

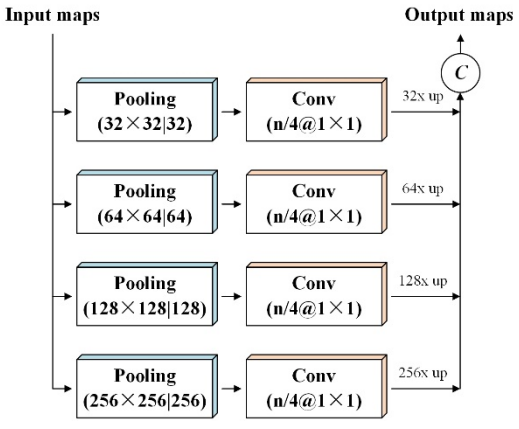


Fig. 8 Illustration of the Pyramid Pooling Module

4.5 PFP validation

An alternative architecture, denoted as PSPNet-V, is developed in the paper to compare with the proposed CrackNet-M. PSPNet-V is a variant of PSPNet that implements a pyramid pooling module for global scene parsing. PSPNet-V shares the same sub-architectures with CrackNet-M, and the only difference between them is that the PSPNet-V replaces the PFP modules used in CrackNet-M with pyramid pooling modules correspondingly. Fig. 8 displays the organizations of a pyramid pooling module that significantly shrinks the input data using four parallel pooling layers in strides of {32, 64, 128, 256}. The channels of pooled feature maps at each level are reduced to a quarter of the original ones using the 1 × 1 convolutions. Unlike the PFP that continuously upsizes data resolutions, the pyramid pooling module applies the nearest-neighbor interpolation to restore them directly. For comparisons, the PSPNet-V is trained under the supervision of skip layer focal loss with identical training hyper-parameters to CrackNet-M.

Table 2 gives the best validation F-measures of PSPNet-V and CrackNet-M during training. It is shown that the PSPNet-V with the best F-measure 91.32% is compared to CrackNet-M with 3.43% performance degradation. This indicates the pyramid pooling module is less competent than the PFP when it comes to improving the performance of crack detection. Up-sampling the downsized feature maps to the original resolution in one single stage is reasonable for PSPNet to segment extremely large objects in sizes of both length and width due to the substantial duplications of interpolation values around a local region. However, the feature maps directly up-sampled from low-resolution maps are so coarse that crack boundaries may be undermined and then cause much correction difficulty in subsequent CBN layers. Different from the pyramid pooling module, the proposed PFP module doubles only the data sizes at each parallel level. Then, the doubled data maps are concatenated with the near finer level for feature correction using the 3 × 3 convolutions, with explicit considerations on both global context acquisition and crack boundary reservation.

5. Testing and evaluation

Finally, the CrackNet-M is assessed on the 500 testing images unused previsouly for further verifications. In

Table 2 The overall Precisions, Recalls, and F-measures of PSPNet-V and CrackNet-M on validation images

Network	Precision (%)	Recall (%)	F-measure (%)
PSPNet-V	94.41	88.43	91.32
CrackNet-M	95.85	93.69	94.75

Table 3 The overall testing results of the described networks in terms of Precisions, Recalls, F-measures, the number of parameters, and forward propagation speed

Network	Precision (%)	Recall (%)	F-measure (%)	Number of Parameters (M)	Forward-prop time per image (sec.)
SegNet	93.56	81.80	87.12	29.46	0.06
CrackNet-V	93.38	86.04	89.42	0.06	0.09
FCN	91.91	87.35	89.45	141.76	0.11
CrackNet-II	94.00	89.72	91.72	0.04	0.07
CrackNet-M	94.28	93.89	94.04	0.42	0.04

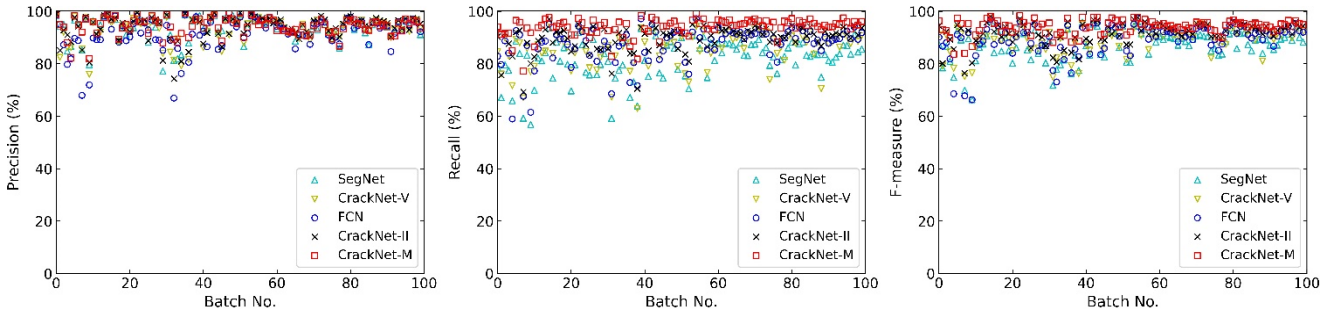


Fig. 9 Illustration of Precisions, Recalls, and F-measures of SegNet, CrackNet-V, FCN, CrackNet-II, and CrackNet-M on the testing images

addition, the CrackNet-M is also compared with other state-of-the-art deep networks, including SegNet (Badrinarayanan *et al*. 2015), FCN (Yang *et al*. 2018), CrackNet-II (Zhang *et al*. 2018), and CrackNet-V (Fei *et al*. 2019), by retraining them using the same image data. Both the SegNet and FCN are structured as encoder-decoder for deep multiscale learning. However, the SegNet is a typical FCNN that adopts pooling indices produced in the encoder to restore data dimensions for object semantic segmentation. While, FCN resumes data signals with the encoder pooling layers that are added directly to deconvolution layers of the same dimensions. Both CrackNet-V and CrackNet-II achieve multiscale learning particularly for pixel-level crack detection simply through the dimension-invariant convolutions stacked continuously with different sizes of filters. In fact, the original four architectures except CrackNet-V receive different sizes of input data from our 3D image data. Considering computing constraints of GPU, the SegNet and FCN are reorganized to implement the same strategies for upsampling as in the original architectures. CrackNet-II is also retrained with the same set of parameters on the images in sizes of 256 × 512 instead of the ones in sizes of 512 × 1024.

Table 3 lists the overall Precisions, Recalls, and F-measures evaluated by the four comparative networks and the proposed CrackNet-M using the 500 testing images. Also, the Precisions, Recalls, and F-measures of these testing images are evaluated for a batch size of 5, as shown in Fig. 9. It is indicated from Table 3 and Fig. 9 that SegNet yields the worst F-measures due to the lowest Recall rates. CrackNet-V demonstrates similarly lesser performance to FCN in terms of F-measures. Although CrackNet-II produces the competitive F-measures compared to SegNet, FCN, and CrackNet-V, the proposed CrackNet-M achieves the highest scores in both Precisions and Recalls. All the four comparative methods can arrive at a high level of Precisions but low Recalls, which means the events of missing cracks occur more frequently than those of incorrectly detecting cracks by these methods. However, CrackNet-M has the highest Precision 94.28% and Recall 93.89%, resulting in the best F-measure up to 94.04%. The performance of CrackNet-M on testing images approximates that on training images, implying that the overfitting problem is successfully circumvented by the L1 regulation and Random Flipping.

The forward propagation time of a single image for the networks and corresponding number of parameters are also listed in Table 3. Note that the forward propagation time is not necessarily positively proportional to the number of parameters. The computation overhead of a network primarily depends on three factors: the shape of data maps, the number of layers, and the count of parameters of each layer. It can be seen that the proposed CrackNet-M has the most efficient processing speed down to 0.04 seconds/image, although its number of parameters is not the least compared to CrackNet-II and Crack-Net-V. This is because more than 99% of the parameters of CrackNet-M is set up in PFP modules that operate these parameters only on the data maps of low dimensions. Architecturally, the CrackNet-M is fully considered in both accuracy and efficiency by Scaling Method instead of empirical architectural design as in most networks. Finally, CrackNet-M can efficiently and effectively extract cracks from 3D pavement surfaces.

To further examine the performance difference among the five methods, three typical detection scenarios on

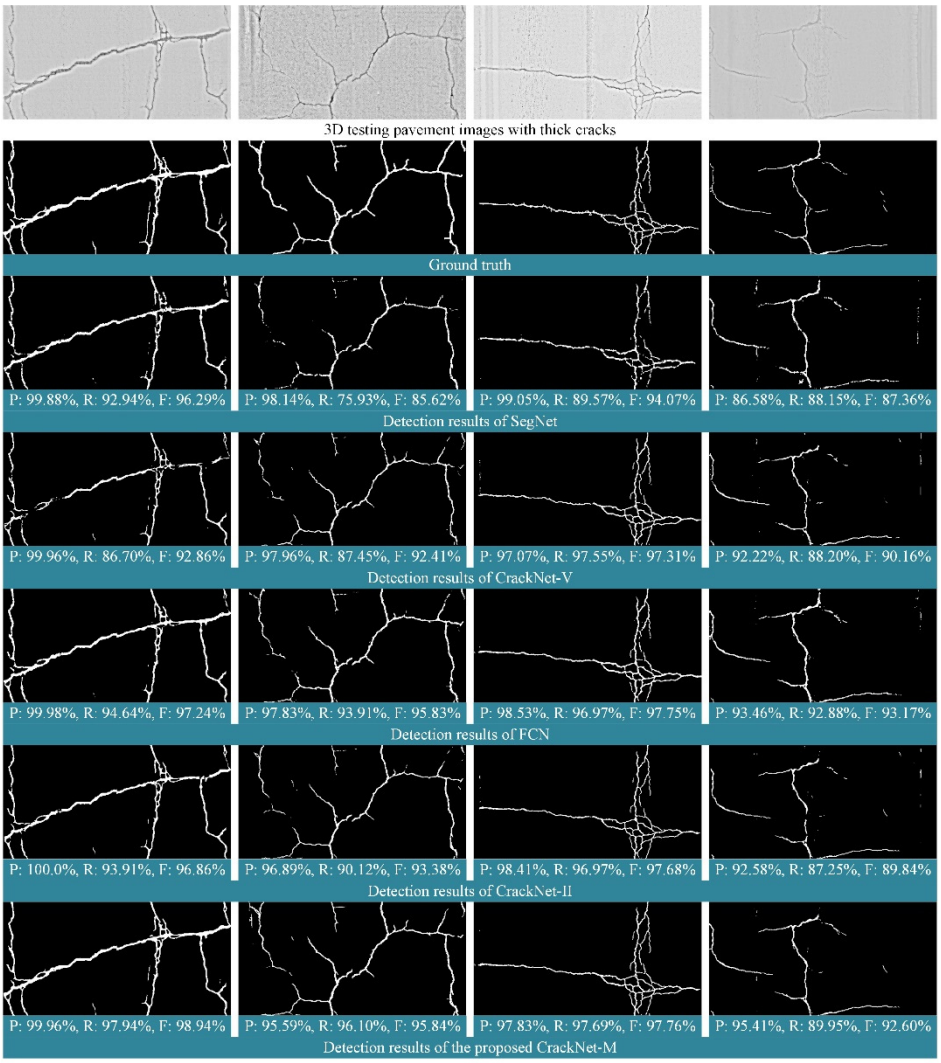


Fig. 10 Illustration of the performance of SegNet, CrackNet-V, FCN, CrackNet-II, and CrackNet-M on the detection of thick cracks

different pavement surfaces are considered in the following tests, which are detection of thick cracks, detection of thin cracks, and removal of pavement shoulder drop-offs.

5.1 Thick cracks

Fig. 10 illustrates four representative testing images that contain various types of thick cracks and corresponding prediction results by the five networks. It clearly shows that the vast majority of thick cracks can be detected via the five methods due to their distinct fracture features, indicating little difference compared with the ground truths. However, it can be found that SegNet and CrackNet-V introduced discontinued cracks at some locations where the fracture signals are not either so salient or thick on the crack width. In addition, all of the four comparative networks would misclassify some parts of longitudinal scratches on the fourth image due to a lack of efficient global context. In contrast to other networks, CrackNet-M can robustly detect thick cracks and suppress global noises simultaneously due to the synergy of the four well-balanced modules.

5.2 Thin cracks

Four typical testing images that include horizontal, longitudinal, and intersected cracks with a small crack width along with the ground truths picked to compare with these networks' predictions are shown in Fig. 11. It can be observed that SegNet is barely able to recognize thin cracks regardless of cracking shapes. This is because the max-pooling indices used in SegNet to resume data can only preserve the locations of dominant features that are essentially excluded in those thin cracks. Although FCN performs better than SegNet in detecting thin cracks due to the contribution of encoder pooling layers to decoder layers for signal retainment, it is still missing finer cracks in the third and fourth images. Dimension-invariant learning without signal loss appears to be beneficial for CrackNet-V and CrackNet-II to detect thin cracks. CrackNet-II yields relatively better outcomes than CrackNet-V due to the use of smaller filters. However, the signals of some finer cracks with respect to both width and depth are so weak that the four contrastive networks cannot detect them well using

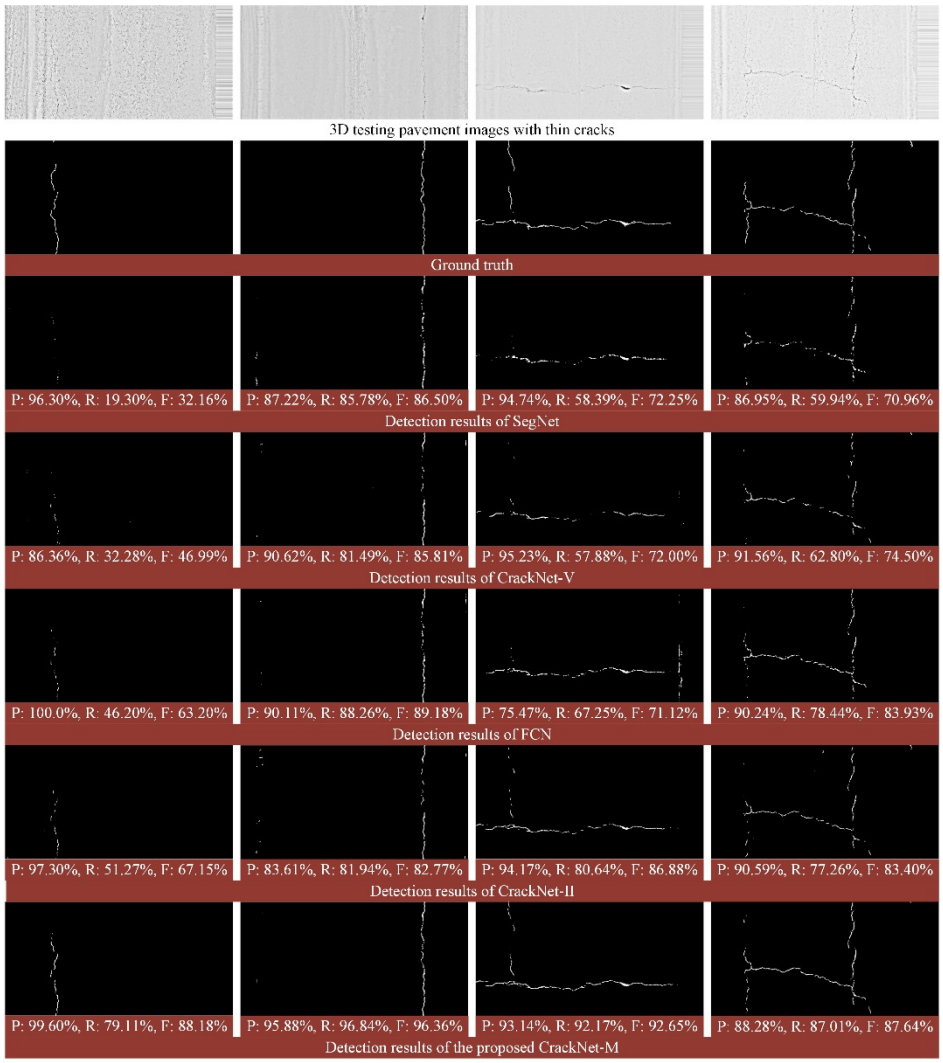


Fig. 11 Illustration of the performance of SegNet, CrackNet-V, FCN, CrackNet-II, and CrackNet-M on the detection of thin cracks

those average architectural designs that would attenuate the signals in the deep passes. Instead, CrackNet-M implements a multi-connected CME module that can effectively enhance the signals of thin cracks without any signal loss and attenuation when working jointly with the CBN module. After enhancement, the majority of thin cracks, including finer cracks, are finally identified in most occasions as compared to the ground truths.

5.3 Shoulder drop-offs

To identify whether the networks can suppress shoulder drop-offs and meanwhile do not skip the longitudinal cracks that are similar to shoulder drop-off patterns, two testing images with longitudinal cracks and two ones with shoulder drop-offs are selected for comparisons, which are shown at the left and right sides of Fig. 12, respectively. It is shown that, CrackNet-V exhibits better performance in removing thicker shoulder drop-off yet at the cost of neglecting thicker longitudinal cracks, which can be attributed to the employment of Leaky Rectified Tanh as the activation function. The other three contrastive networks can retrieve most of longitudinal cracks but misclassify the shoulder drop-offs as longitudinal cracks without the ability to distinguish between them. The primary reason is that these networks lack efficient global context and accurate local context for the unique representation of different patterns. Nevertheless, CrackNet-M can handle well the detection of longitudinal cracks and removal of shoulder drop-off simultaneously with a desirably balanced accuracy without regard to the widths of different patterns. Architecturally, CrackNet-M employs three PFP modules for higher levels and larger scales of perceptions, and meanwhile, as well as a CBN module for higher levels and finer scales of learning as opposed to other networks. The fusions of crack features in both coarse and fine scales are of high confidence. Further integration of these fusions of different levels fundamentally ensures the capability of CrackNet-M to robustly separate various cracks from complex non-crack patterns, which discloses the performance consistency in the testing examples.

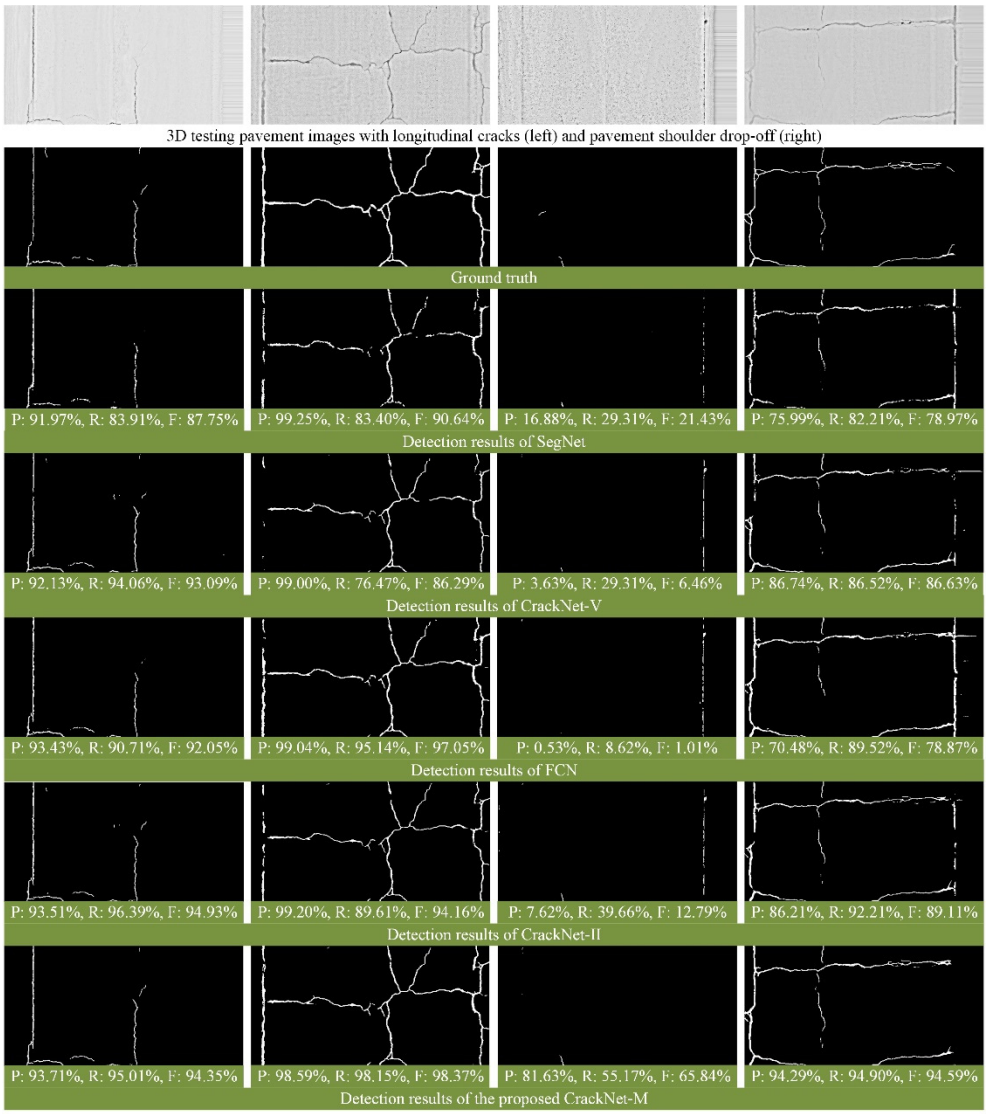


Fig. 12 Illustration of the performance of SegNet, CrackNet-V, FCN, CrackNet-II, and CrackNet-M on the removal of shoulder drop-offs

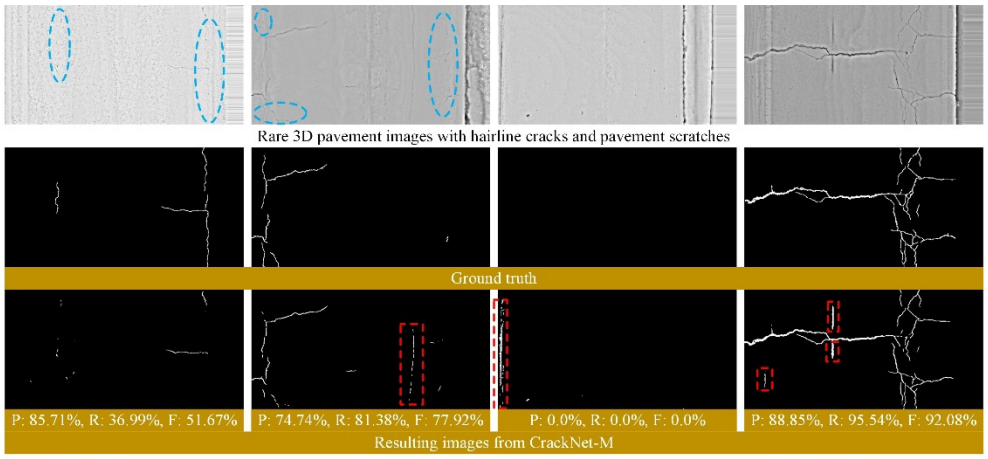


Fig. 13 Typical errors of CrackNet-M on the testing images

6. Discussion

Fig. 13 displays several typical errors caused by CrackNet-M when processing rare pavement surfaces. The false-negative errors are highlighted in dashed circles on the original images, whereas the false-positive errors are marked out by dashed rectangles on the processed images. The false-negative errors are the results of missing hairline cracks that are nearly invisible to human eyes. There are two leading reasons to account for this event. First, hairline cracks are substantially low at the depth, which contributes less information in the CME module to the enhancement of cracks, even though the widths of hairline cracks are augmented based on their depths. Second, hairline cracks are so scarce with regard to the number of pixels or samples that the CrackNet-M has finally learned to neglect them in the presence of vast thick cracks. To improve the performance of CrackNet-M for detecting hairline cracks, one of the possible solutions is to enhance the contrast of hairline cracks to the background using preprocessing techniques. The second solution might be to employ edge detectors to pre-extract edge features for a network to learn subsequently. Alternatively, more training examples with hairline cracks could be added to current data library for balanced learning.

There can be two common types of false-positive errors generated by CrackNet-M. It is shown in the second image in Fig. 13 that the first type of false-positive errors is the misclassification of long curved scratches. The second type of false-positive errors, such as those marked on the third and fourth images, mainly results from short longitudinal scratches. In the field, the two types of pavement scratches are normally the surface defects caused by the construction equipment during or after pavement construction, showing similar physical characteristics to cracks in shape and width, except the depth. As one of the structural damages, cracks generally bear deeper depths than scratches. However, in the obtained 3D pavement images, the scratches and cracks instead present similar pixel depths as the laser of PaveVision3D system cannot reach deep inside the cracks of irregular surfaces when photographing the cracks. This explains why these scratches are easily mistaken by the algorithm as cracks. For further improvements, the long curved scratches are likely to be eliminated subsequently by analyzing their subtle curvature changes. While, the misrecognition errors on short longitudinal scratches may be solved by post-processing techniques such as connected-component-based shape and area analysis that remove longitudinal patterns whose area is lower than a predefined threshold.

Actually, the false-negative errors on hairline cracks and false-positive errors on scratches are not exclusive to the proposed CrackNet-M, which also impact the other four comparative algorithms presented in this study. On the average, the errors tied to CrackNet-M are rare occurrences and can scarcely lower the overall performance in terms of F-measures. Moreover, the last three testing images with shoulder drop-offs further indicate that the proposed CrackNet-M is capable of robustly erasing shoulder drop-offs without adverse impacts on the concurrent detection of thick and thin cracks.

7. Conclusions

An efficient and specifically-armed architecture termed CrackNet-M is introduced in the paper for pixel-level crack detection on 3D asphalt pavement surfaces. The CrackNet-M is composed of four architectural modules: a central branch net (CBN), a crack map enhancement (CME) module, three pooling feature pyramids (PFP), and an output layer. Specifically, the CBN module conducts scale-invariant feature learning by excluding pooling layers to retain accurate spatial relations for pixel-to-pixel supervision. Compared with encoder-decoder networks of scale-invariant learning at early and end layers, the abstraction level formed by CBN is much deeper due to the representation of all the seven levels of scale-invariant layers. The CME module implements a max-pooling function to enhance the continuity of thin cracks. Different from traditional pooling reductions, the max-pooling layer used in CME does not alter the dimensions of features by virtue of the minimal stride. The accuracy in retrieving thin cracks can thus be improved by CME when working with the CBN that recompenses the signal loss or spatial distortion caused by the max-pooling operation. The attenuation problem with enhanced crack signals is also prevented through the skip concatenations and fusions of CME and CBN. The PFP module enforces two major procedures, down-sampling and up-sampling with multiscale semantics, for the detection of thick cracks and exclusion of non-crack patterns. Five parallel average pooling layers are arranged in the down-sampling process to build a global feature pyramid by directly downsizing the same data in a series of exponential strides. The expansion layer and max-pooling layer are combined to adaptively enhance the down-sampled features of each level. After these dispositions, the attenuation of semantic motif, which often occurs at a deep encoder, can be sufficiently avoided. To finely extract crack features from the global feature pyramid of five levels, a stepwise feature pyramid is effectively constituted by four squeeze layers in the up-sampling process. The PFP module enables small filters to yield large-scale features in the high-resolution maps (i.e., fine maps) with self-corrections. Three duplications of the PFP module can endow deeper abstractions on complex patterns compared to those usual encoder-decoders.

CrackNet-M is trained using the focal loss skip layer supervision proposed to further advance the network performance. Skip layer supervision is accomplished through the fusions of point features from three fused dimension-invariant intermediate layers in CBN. It is demonstrated that this new training tactic exposes three advantages over traditional single layer supervision with respect to the accuracy, robustness, and convergence rate, which are all enhanced due to the skip connections of early intermediate layers to the output layer.

Using training techniques, including Mini-batch, Adam Optimizer, Skip Layer Supervision, Focal Loss, Scaling Method, Normalized Initialization, Random Flipping, and L1 Regulation, the optimization of CrackNet-M is successfully completed after 1600 iterations of running over a total of 2,500 pavement images. CrackNet-M achieves a testing Precision, Recall, and F-measure of 94.28%,

93.89%, and 94.04%, respectively. It is shown that CrackNet-M exceeds other well-developed deep networks in many aspects. By comparison with SegNet, CrackNet-V, FCN, and CrackNet-II, the proposed CrackNet-M can not only detect thick cracks better but exhibit desirable accuracy in retrieving thin and hairline cracks. In addition, the removal of pavement shoulder drop-offs, which has been a difficult problem by far for most networks and algorithms, is also accomplished with CrackNet-M with notable and concurrent accuracies and robustness.

A total number of 423,249 parameters are trained in CrackNet-M, with which the average forward time for processing a single image is roughly 40 ms on a single 1080 Ti GPU. In other words, the CrackNet-M is able to scan a pavement surface at a speed of 110 mph, which means a real-time pavement distress survey desired in engineering practice can be achieved using the proposed network. The training and testing images used for CrackNet-M are all 3D surface data in sizes of 256 × 512 downsized from original data with a resolution of 2048 × 4096. However, the CrackNet-M is also suitable for the 4096 × 8192 (8K) data collected from the latest PaveVision3D system as long as the data are resized to 256 × 512 based on the min-pooling technique. Conversely, the 8K data essentially contain more accurate information in terms of surface characteristics and are likely to produce better results in the real world.

The CrackNet-M experiences some difficulties in extracting extreme hairline cracks and inabilities to differentiate crack-like scratches from real pavement cracks at times. Technically speaking, it is quite challenging for any network to accomplish these two missions desirably with the current image library that has limited examples with hairline cracks and scratches. However, the improvements of CrackNet-M are proceeding with efforts in two directions. Preparing more complex and diverse pavement images with rare condition scenarios, such as hairline cracks, scratches, potholes and patches, is the initial step for future enhancement. Although manual preparation of ground truths at the pixel level for these scenarios demands resources such as labor and time, it is indispensable for a network to be trained pervasively and successfully on various pavement conditions. Based on the newly constructed image library, the current CrackNet-M shall be improved with comprehensive considerations on accuracy, robustness, and efficiency by modifying the current architecture or adding other purpose-specific functional modules.

Acknowledgments

The study presented in this article was partially supported by the National Natural Science Foundation of China (Grant No. 51208419) and the Fundamental Research Funds for Central Universities of China (Grant No. 2682021CX009).

References

Alipour, M., Harris, D.K. and Miller, G.R. (2019), "Robust pixel-level crack detection using deep fully convolutional neural networks", *J. Comput. Civil Eng.*, **33**(6), 04019040. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000854

Asadi, P., Mehrabi, H., Asadi, A. and Ahmadi, M. (2021), "Deep convolutional neural networks for pavement crack detection using an inexpensive global shutter RGB-D sensor and ARM-based single-board computer", *Transport. Res. Record*, **2675**(9) 03611981211004974. https://doi.org/10.1177/03611981211004974

Badrinarayanan, V., Handa, A. and Cipolla, R. (2015), "SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling", arXiv preprint arXiv:1505.07293. https://doi.org/10.48550/arXiv.1505.07293

Bang, S., Park, S., Kim, H. and Kim, H. (2019), "Encoder–decoder network for pixel-level road crack detection in black-box images", *Comput.-Aided Civil Infrastr. Eng.*, **34**(8), 713-727. https://doi.org/10.1111/mice.12440

Carré, A., Klausner, G., Edjlali, M., Lerousseau, M., Briend-Diop, J., Sun, R., Ammari, S., Reuzé, S., Alvarez Andres, E., Estienne, T. and Niyoteka, S. (2020), "Standardization of brain MR images across machines and protocols: bridging the gap for MRI-based radiomics", *Scientific Reports*, **10**(1), 12340.

Cha, Y.J., Choi, W. and Büyüköztürk, O. (2017), "Deep learning-based crack damage detection using convolutional neural networks", *Comput.-Aided Civil Infrastr. Eng.*, **32**(5), 361-378. https://doi.org/10.1111/mice.12263

Cha, Y.J., Choi, W., Suh, G., Mahmoudkhani, S. and Büyüköztürk, O. (2018), "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types", *Comput.-Aided Civil Infrastr. Eng.*, **33**(9), 731-747. https://doi.org/10.1111/mice.12334

Daniel, A. and Preeja, V. (2014), "A novel technique for automatic road distress detection and analysis", *Int. J. Comput. Applicat.*, **101**(10), 18-23.

Doshi, K. and Yasin, Y. (2020), "Road damage detection using deep ensemble learning", In: *2020 IEEE International Conference on Big Data (Big Data*), Atlanta, GA, USA, December.

Du, Y., Pan, N., Xu, Z., Deng, F., Shen, Y. and Kang, H. (2020), "Pavement distress detection and classification based on YOLO network", *Int. J. Pave. Eng.*, **22**(13), 1659-1672. https://doi.org/10.1080/10298436.2020.1714047

Escalona, U., Arce, F., Zamora, E. and Sossa, H. (2019), "Fully convolutional networks for automatic pavement crack segmentation", *Computación y Sistemas*, **23**(2), 451-460. https://doi.org/10.13053/cys-23-2-3047

Fawcett, T. (2006), "An introduction to ROC analysis", *Pattern Recogn. Lett.*, **27**(8), 861-874. https://doi.org/10.1016/j.patrec.2005.10.010

Fei, Y., Wang, K.C.P., Zhang, A., Chen, C., Li, J.Q., Liu, Y., Yang, G. and Li, B. (2019), "Pixel-level cracking detection on 3D asphalt pavement images through deep-learning-based CrackNet-V", *IEEE Transact. Intell. Transport. Syst.*, **21**(1), 273-284. https://ieeexplore.ieee.org/document/8620557

Feng, X., Xiao, L., Li, W., Pei, L., Sun, Z., Ma, Z., Shen, H. and Ju, H. (2020), "Pavement crack detection and segmentation method based on improved deep learning fusion model", *Mathe. Problems Eng.*, 2020. https://doi.org/10.1155/2020/8515213

Gao, Y., Zhai, P. and Mosalam, K.M. (2021), "Balanced semisupervised generative adversarial network for damage assessment from low-data imbalanced-class regime", *Comput.-Aided Civil Infrastr. Eng.*, **36**(9), 1094-1113. https://doi.org/10.1111/mice.12741

Glorot, X. and Bengio, Y. (2010), "Understanding the difficulty of

training deep feedforward neural networks", *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, Sardinia, Italy, May.

Gou, C., Peng, B., Li, T. and Gao, Z. (2019), "Pavement Crack Detection Based on the Improved Faster-RCNN", *Proceedings of 2019 IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Dalian, China, November.

Guo, C., Fan, B., Zhang, Q., Xiang, S. and Pan, C. (2019), "AugFPN: improving multi-scale feature learning for object detection", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CA, USA, June.

He, K., Zhang, X., Ren, S. and Sun, J. (2015), "Deep residual learning for image recognition", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, June.

Hsieh, Y.A. and Tsai, Y.J. (2020), "Machine learning for crack detection: review and model performance comparison", *J. Comput. Civil Eng.*, **34**(5), 04020038. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000918

Hwang, J.J. and Liu, T.L. (2015), "Pixel-wise deep learning for contour detection", arXiv preprint arXiv:1504.01989. https://doi.org/10.48550/arXiv.1504.01989

Iandola, F.N., Moskewicz, M.W., Ashraf, K., Han, S., Dally, W.J. and Keutzer, K. (2016), "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 MB model size", arXiv preprint arXiv:1602.07360. https://doi.org/10.48550/arXiv.1602.07360

Jahanshahi, M.R., Jazizadeh, F., Masri, S.F. and Becerik-Gerber, B. (2013), "Unsupervised approach for autonomous pavement-defect detection and quantification using an inexpensive depth sensor", *J. Comput. Civil Eng.*, **27**(6), 743-754. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000245

Kingma, D.P. and Ba, J.L. (2015), "Adam: a method for stochastic optimization", arXiv preprint arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980

König, J., Jenkins, M.D., Barrie, P., Mannion, M. and Morison, G. (2019), "A convolutional neural network for pavement surface crack segmentation using residual connections and attention gating", In: *2019 IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, September.

Lee, B.J. and Lee, H.D. (2004), "Position-invariant neural network for digital pavement crack analysis", *Comput.-Aided Civil Infrastr. Eng.*, **19**(2), 105-118. https://doi.org/10.1111/j.1467-8667.2004.00341.x

Li, B., Wang, K.C.P., Zhang, A., Yue, F. and Giuseppe, S. (2019a), "Automatic segmentation and enhancement of pavement cracks based on 3D pavement images", *J. Adv. Transport.*, 2019, 1813763. https://doi.org/10.1155/2019/1813763

Li, S., Zhao, X. and Zhou, G. (2019b), "Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network", *Comput.-Aided Civil Infrastr. Eng.*, **34**, 616-634. https://doi.org/10.1111/mice.12433

Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. (2016), "Feature pyramid networks for object detection", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, AZ, USA, June.

Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P. (2017), "Focal loss for dense object detection", *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, October.

Ma, K., Hoai, M. and Samaras, D. (2017), "Large-scale continual road inspection: visual infrastructure assessment in the wild", In: BMVC. https://dx.doi.org/10.5244/C.31.151

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T. and Omata, H. (2018), "Road damage detection using deep neural networks with images captured through a smartphone", arXiv preprint arXiv:1801.09454. https://doi.org/10.48550/arXiv.1801.09454

Mohammed Y., Uddin, N., Tan, C. and Shi, Z. (2020), "Crack detection using faster R-CNN and point feature matching", *Civil Eng. Res. J.*, **10**(3), 555790.

Munawar, H.S., Hammad, A.W., Haddad, A., Soares, C.A.P. and Waller, S.T. (2021), "Image-based crack detection methods: A review", *Infrastructures*, **6**(8), 115. https://doi.org/10.3390/infrastructures6080115

Sathya, K., Sangavi, D., Sridharshini, P., Manobharathi, M. and Jayapriya, G. (2020), "Improved image based super resolution and concrete crack prediction using pre-trained deep learning models", *J. Soft Comput. Civil Eng.*, **4**(3), 34-44. https://doi.org/10.22115/SCCE.2020.229355.1219

Shi, Y., Cui, L., Qi, Z., Meng, F. and Chen, Z. (2016), "Automatic road crack detection using random structured forests", *IEEE Transact. Intell. Transport. Syst.*, **17**(12), 3434-3445. https://ieeexplore.ieee.org/document/7471507

Sizyakin, R., Voronin, V., Gapon, N. and Pižurica, A. (2020), "A deep learning approach to crack detection on road surfaces", In: *Artificial Intelligence and Machine Learning in Defense Applications II*, **11543**(115430P), 1-7. https://doi.org/10.1117/12.2574131

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2014), "Going deeper with convolutions", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, June.

Tan, M. and Le, Q. (2019), "Efficientnet: Rethinking model scaling for convolutional neural networks", In: *International Conference on Machine Learning*, CA, USA, June.

Wang, G., Wang, K.C.P., Yang, G., Liu, Y., Li, J.Q. and Peters, W. (2022), "Nondestructive bridge deck evaluation using sub-mm 3D laser imaging technology at highway speeds", *J. Bridge Eng.*, **27**(6), 04022045. https://doi.org/10.1061/(ASCE)BE.1943-5592.0001888

Xu, H., Su, X., Wang, Y., Cai, H., Cui, K. and Chen, X. (2019), "Automatic bridge crack detection using a convolutional neural network", *Appl. Sci.*, **9**(14), 2867. https://doi.org/10.3390/app9142867

Yang, X., Li, H., Yu, Y., Luo, X. and Huang, T. (2018), "Automatic pixel-level crack detection and measurement using fully convolutional network", *Comput.-Aided Civil Infrastr. Eng.*, **33**(12), 1090-1109. https://doi.org/10.1111/mice.12412

Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X. and Ling, H. (2019), "Feature pyramid and hierarchical boosting network for pavement crack detection", *IEEE Transact. Intell. Transport. Syst.*, **21**(4), 1525-1535. https://doi.org/10.48550/arXiv.1901.06340

Ye, X.W., Jin, T. and Yun, C.B. (2019), "A review on deep learning-based structural health monitoring of civil infrastructures", *Smart Struct. Syst., Int. J.*, **24**(5), 567-586. https://doi.org/10.12989/sss.2019.24.5.567

Zalama, E., Gomez-Garcia-Bermejo, J., Medina, R. and Llamas, J. (2014), "Road crack detection using visual features extracted by Gabor filters", *Comput.-Aided Civil Infrastr. Eng.*, **29**(5), 342-358. https://doi.org/10.1111/mice.12042

Zhang, A., Wang, K.C.P., Li, B., Yang, E., Dai, X., Peng, Y., Fei, Y., Liu, Y., Li, J Q. and Chen, C. (2017), "Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network", *Comput.-Aided Civil Infrastr. Eng.*, **32**(10), 805-819. https://doi.org/10.1111/mice.12297

Zhang, A., Wang, K. C. P., Fei, Y., Liu, Y., Tao, S., Chen, C., Li, J. Q. and Li, B. (2018), "Deep learning-based fully automated pavement crack detection on 3D asphalt surfaces with an improved CrackNet", *J. Comput. Civil Eng.*, **32**(5), 04018041. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000775

Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J. (2017), "Pyramid

scene parsing network", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July.

BS