Multimedia Tools and Applications (2023) 82:4387–4404
https://doi.org/10.1007/s11042-022-13422-w

Cracks identification using mask region-based denoised deformable convolutional network

Kia Wei Kee[1] · King Hann Lim[1] · Chin Hong Lim[2] · Wen Loong Lim[2] · Huei Ee Yap[3]

Received: 8 January 2021 / Revised: 15 March 2022 / Accepted: 2 July 2022 /
Published online: 26 July 2022
© The Author(s) 2022

Abstract

Cracks are one of the critical structural defects in building assessment to determine the integrity of civil structure. Structural surveying process using computer vision is required to automatically identify cracks. The application of Convolutional Neural Networks (CNNs) is limited by its fixed geometric kernels to extract the irregular shape of cracks. In this paper, a mask Region-based Denoised Deformable Convolutional Network (R-DDCN) is proposed to detect cracks for accurate instance segmentation and image classification. Denoised deformable convolution is introduced to improve the modeling capability of convolution layer. It adopts the existing deformable convolution, with non-local means as a denoising mechanism to optimize the augmentation of spatial sampling locations with filtered offsets. Experimental results show that the proposed mask R-DDCN has lower validation loss and improved mean accuracy precision of mAP_{75} from 66.7% to 76.7% as compared to the mask R-CNN. Mask R-DDCN can perform better modeling capability in cracks identification.

Keywords Cracks identification · Mask R-CNN · Denoised deformable Convolutional network

1 Introduction

Cracks are one of the critical structural defects to determine an early possible structural failure in a building integrity inspection [17]. Building surveyors conduct structural inspection by collecting visual data to identify various defects due to environmental exposure during the service life of the structure (such as cracks, loss of material, rusting of metal bindings, etc). Visual inspection can provide preliminary information that may lead to positive identification of the cause of observed distress. However, its effectiveness depends on the knowledge and experience of the surveyors and is prone to human error [36]. Additionally,

✉ Kia Wei Kee
keekw97@hotmail.com

[1] Curtin University Malaysia, CDT 250, 98000 Miri, Malaysia

[2] SafeT5 Sdn. Bhd., Sungai Buloh, Selangor, Malaysia

[3] LP Research Inc, Tokyo, Japan

inspection of mega structures such as dams, bridges and tall buildings can be prohibitively risky and difficult due to hard-to-reach facets [36]. Therefore, automated building surveying tools [29] are highly required to reduce the task complexity and to prevent the occurrence of catastrophic event.

The use of deep learning neural networks in automated cracks detection for building surveying has drawn increasingly attention in the civil and construction industry. The latest techniques reviewed in automated cracks identification [21, 26, 34, 37] apply Convolutional Neural Networks (CNNs) as the baseline to detect cracks as it is always in multi-orientation and irregular shape. The convolution layer in CNNs possess the limitation of geometric variation in feature extraction due to the fixed structure of the convolution kernels [6]. The size of the receptive field in CNNs is always fixed, which is highly undesirable for the deeper layer that serves to extract high-level feature from an image by encoding semantics over spatial locations [49]. Multiple efforts are suggested to solve this challenge, including some notable works such as scale-invariant feature transform [15, 25], deformable part-based models [9], transformer networks [16, 38], active convolution [19], moment-based local feature extraction [48], and graph-based networks [47]. However, in the civil and construction industry, the capability of modelling unknown geometric transformation or variation is highly required in cracks identification.

Besides that, there are several recent development on anchor-free instance segmentation networks such as PolarMask and DeepSnake. PolarMask network [43] is used to perform single shot anchor-box free instance segmentation for bounding box detectors and instance-wise recognition tasks. It formulates the instance segmentation problem as predicting contour of instance through instance center classification and dense distance regression in a polar coordinate. On the other hand, DeepSnake network [28] applies the classic contour-based approach to deform an initial contour to match the object boundary using deep snake algorithm. It exploits the cycle-graph structure of a contour using circular convolution for real-time instance segmentation. However, these approaches have the assumption of circular closed-loop contour around the segmented instances to detect instance segmentation. As a result, these approaches are not suitable to be used in cracks segmentation, which has irregular shape and pattern.

In this paper, a novel mask Region-based Denoised Deformable Convolutional network (R-DDCN) is proposed in Fig. 1 to handle the variant of receptive field of CNN during convolution layer. The proposed Mask R-DDCN uses pixel-wise deformable convolution to optimize the augmentation of the sampling location of the convolution kernel with filtered offset. It can extract accurate semantic segmentation within each bounding box with an improved mAP. The basic architecture of R-DDCN is inherited from mask R-CNN [12]

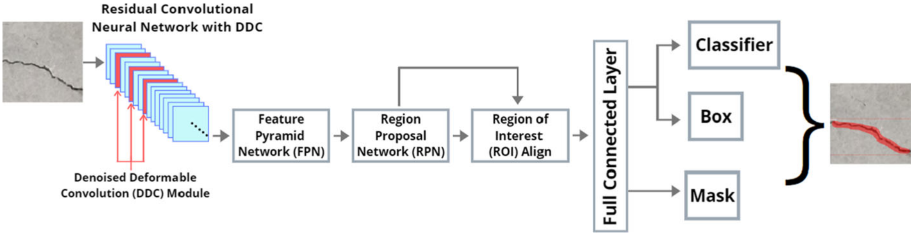
Residual Convolutional Neural Network with DDC
Denoised Deformable Convolution (DDC) Module
Feature Pyramid Network (FPN)
Region Proposal Network (RPN)
Region of Interest (ROI) Align
Full Connected Layer
Classifier
Box
Mask

Fig. 1 Block Diagram of the proposed mask Region-based Denoised Deformable Convolutional Network (R-DDCN) in detecting cracks. The use of DDC modules are used in the convolutional layers to remove noise occurred in the feature maps

with the following modifications, i.e. (a) denoised mechanism removing distortion using non-local means, (b) deformable convolution incorporated with 2D learnable offset into the regular convolution through the preceding feature maps resulting in deformation of the standard sampling grid. This hybrid module is known as Denoised Deformable Convolution (DDC) and it is integrated into mask R-CNN to improve the geometric transformation for cracks identification.

2 Related work

Crack is defined as a line on the surface of a concrete without completely breaking apart due to the drastic changing conditions of environment and a limited lifespan of the structures. Crack inspections are generally divided into destructive testing (DT) and non-destructive testing (NDT). NDT is the inspection assessment in detecting defects and flaws in accordance to a certain standards without altering or harming the object being testing, and vice versa to DT [7]. Automated crack detection using camera as proposed is a type of NDT because it is more effective than manual crack detection without altering or harming the inspected object. Other options of NDT methods [1] includes ultrasonic testing, X-ray and Gamma-ray testing, and laser testing. The general steps of computer vision in the crack identification consist of: (a) Image of cracks are captured using camera, (b) Collected images are pre-processed to make it more efficient in image analysis steps. (c) Statistical techniques are applied to analyse cracks property from the pre-processed images, (d) Crack detection is performed to extract crack features where parameters such as size and direction of crack can be measured. The rapid advancement of computer vision techniques have been applied in the field of civil and construction. They can be split into two categories [10], i.e. (a) Image processing techniques and (b) Deep learning based techniques. Prior research works are further analysed and discussed to highlight an overview of latest advancements with the benefits and limitations in each category.

2.1 Image processing techniques

Conventional image processing techniques use the basic statistical approach to directly process image data to obtain cracks properties such as edges, shapes and other salient features that can be defined using mathematical models. The information needed for crack detection can be purely derived from pixel-based or pre-processed data, which is possibly assisted by other hardware and measuring tools. Zou et al. [52] proposed a fully automatic cracks detection method called CrackTree. This method first uses a new geodesic shadow-removal algorithm to remove the shadow without affecting the cracks. This algorithm offers the benefit of an accurate modeling in large penumbra areas with well preservation of strong particle textures. Subsequently, tensor voting is applied to construct a crack probability map. This probability map is used to construct minimum spanning tree of the graph model and then used to conduct recursive edge pruning to identify the final crack curve. The shadow removal performed as pre-processing process is essential to produce accurate result as the experiment conducted recorded a lower performance without shadow removal.

Zhang et al. [50] proposed a six-stage integrated crack detection and classification methodology for subway tunnels which utilizes the high-speed line scan camera to capture low cost-high quality image. The input image is pre-processed before passing through the black top-hat transform in order to detect possible cracks. Following by that, the output representation is then gone through crack segmentation and classification. This technique obtains an average error rate of 6 %. However, the disadvantage is that the digital image is obtained by laser scanning, leading to some of the cracks not presented as dark pixels as they are illuminated by laser lights, resulting in decreased accuracy. Shan et al. [35] proposed a stereo vision-based crack width detection method to quantitatively analyse the width of concrete cracks. This method uses cameras to capture 3D coordinate of crack edge and do not require any conversion of measurement unit of the captured data. A novel Canny-Zernike combination algorithm is implemented to crack edge coordinate which can achieve up to 0.02 sub-pixel precision to obtain the 3D coordinate of crack edge. The width is computed through the minimum distance between two sides of crack edge. The proposed method can measure crack width as accurate as using a vernier caliper and hence, it is applicable for engineering application.

Lee et al. [20] proposed crack detection technique with base image. This system uses a camera as an input device, which then convert the input image to gray scaled images for fast processing and a wiener filtering is applied to filter our noises. After that, a Sobel mask is applied to detect edge line in an image. Lastly, local image amplitude mapping process is applied to the base and test image to reduce false indication and detect the cracks. This crack detection technique with base image achieves a fairly decent and accuracy experimental result, even though this technique has the tendency to obtain a false positive error with a large window size. Hoang et al. [13] proposed an image processing model that utilizes min-max gray level discrimination as a pre-processing step to enhance the otsu binarization approach before performing shape analyses for refining the performance of crack detection. The proposed model is effective in detecting crack objects and analysing their characteristic such as height, width, and orientation. However, the limitation to this approach is the need to fine-tune the ratio and margin parameters as well as the high failure rate of detecting thin crack objects. On the other hand, Qu et al. [31] proposed an ultra-efficient crack detection algorithm (CrackHHP) and an improved pre-extraction and second percolation process. CrackHHP can improve the percolation speed whereas the second percolation can detect small cracks and fractures. Each pixel is first designated a weight value depending on the pixel brightness and the candidate dark pixel then extracted. The dot noise is removed before percolating the dark pixels and the neighbouring pixels to connect the tiny fracture to detect cracks. This method display accurate detection and fast computation time.

2.2 Deep learning-based techniques

Deep learning-based techniques adopt neural networks to learn patterns from input image directly with the guidance of dataset and training mechanism. The current popular learning architecture in crack identification is Convolutional Neural Networks (CNNs). Fan et al. [22] proposed a crack detection solution based on a deep ConvNet, which is trained on square image patches using the provided ground truth information to classify positive and negative patch. The objective of the solution is to identify whether if a specific pixel is part of a crack. Hence, the patch is considered as positive patch if it is a crack pixel or is within close vicinity of one. Otherwise, the patch is considered as negative patch. This proposed solution shows superior performance in identifying positive patch from background. Zhao et al. [51] proposed to use a deep CNN to build classifier which can recognize and detect cracks from input image directly using smartphone. The mentioned CNN classifier needs to be trained using a large set of training data in order to detect and classify cracks effectively.

Prasanna et al. [30] proposed an automated crack detection algorithm, called spatially-tuned robust multi feature (STRUM) classifier which is used on a robotic scanning system. STRUM classifier is used to obtain high accuracy image by performing robust curve fitting to spatially localize potential crack regions without affected by the existing noise and distraction, before computing the visual feature using support vector machine, adaboost and random forest. After that, a crack density map is computed to provide a global view of the spatial patterns of cracks. Hu et al. [14] proposed a non-destructive crack detection system which integrates both time and spatial pattern mining for crack information with Faster R-CNN. This system uses thermal video sequence as an input, which then is compressed by the spatial-transient pattern separation. After that, cracks are detected through a trained Faster R-CNN and visualized with the bounding box.

Janpreet et al. [18] proposed the usage of Mask R-CNN using smartphone captured image as input to detect road damages such as cracks. Xu et al. [45] proposed to use CNNs based crack detection model, taking advantage of atrous convolution, Atrous Spatial Pyramid Pooling (ASPP) module and depth-wise separable convolution. Atrous convolution can exponentially expand the receptive field without reducing the resolution, resulting in a denser feature map. The ASPP module enables the network to extract multi-scale image feature information whereas depth wise separable convolution reduces the computational complexity of the model. Cho et al. [5] proposed the application of deep learning-based technique for crack assessment on civil structure. The proposed Mask R-CNN yielded an average accuracy of 88.7% in crack detection on real structures such as bridges, tunnel and concrete pavement.

Lee et al. [21] proposed the use of Sobel Filter as the edge agreement head to improve the average precision of 63.3% in mask Region-based Convolutional Neural Network (R-CNN). The proposed enhanced Mask R-CNN solution shows superior performance compared to the original Mask R-CNN in detecting crack on complex background. Ryu et al. [34] performed cracks study on the fire-damaged beams using a two-streamed CNN, i.e. weighting feature network and low-level feature network to extract both abstract and primitive information in cracks detection. Convolution-deconvolution structure was incorporated with the application of element-wise multiplication at the end of the network to combine two feature maps for precise cracks location. Mei et al. [26] proposed the use of densely connected convolutional layers in a feed-forward manner with multiple-level features fusion for cracks detection on road pavements. On the other hand, Song et al. [37] proposed an addition of multiscale dilated convolutional module in the deep convolutional neural network, which is known as CrackSeg to automate the detection of road pavement cracks. The multiscale convolutional layers are incorporated in the CNN architecture to learn rich deep convolutional features, allowing more detection of crack feature on a complex background.

3 Denoised deformable convolution

A denoised deformable convolution block is introduced in Fig. 2 by incorporating a denoising mechanism into the deformable convolution. Denoising mechanism is necessary to remove image noises presented during the image acquisition or processing as their appearance disturbs the original information in the input image, resulting in distortion in the feature map [3]. Distortion in the feature map causes inaccurate or redundant offset, leading to unnecessary activation at the output feature map. In the current practices, there are multiple approaches for image denoising, such as non-local means [4], bilateral filter [40], mean filter [39] and median filter [46]. Removing noises using non-local means with embedded Gaussian function as its feature-dependent weighting function yields better performance compared to other methods due to its similarity generalization from the non-local region [44].

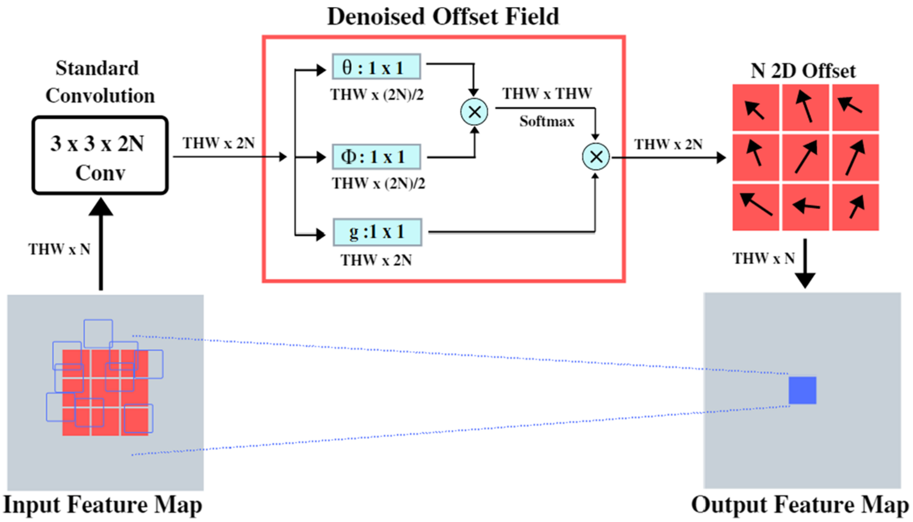

Fig. 2 Denoised Deformable Convolution (DDC) generates a denoised offset field using non-local means and produced a deformable kernel. The operation in the denoised block uses 1 × 1 convolution and the softmax operation is denoted in the pipeline before doing the 2D offsetting. The feature map is denoted as tensor shape (eg: THW × N) where N is the dimension of kernel, T is the batch size, H is the height and W is the width of feature map. The "⊗" represents the matrix multiplication

Non-local means [44] is a denoising mechanism that takes a weighted mean of features in all spatial location (L) in the input feature in order to obtain the denoised output feature map using,

$$m_i(n) = \frac{1}{C(n)} \sum_{\forall j \in L} f(n_i, n_j) \cdot g(n_j), \tag{1}$$

where i is the index of an output position, j is the index that enumerates all possible positions, $g(\cdot)$ is an unary function that computes a representation of an input signal at the position j and linear embedding is considered in this context, where $g(n_j) = W_g \cdot n_j$ and W_g is the learnable weight matrix. $C(n)$ is a normalization factor of $f(n_i, n_j)$, where $C(n) = \sum_{\forall j} f(n_i, n_j)$. This operation is a feature-dependent weighting (pairwise) function, which in this case is embedded Gaussian function defined as follows [41],

$$f(n_i, n_j) = e^{\frac{1}{\sqrt{d}} \theta(n_i)^T \varphi(n_j)}, \tag{2}$$

where the $\theta(n)$ and $\varphi(n)$ refer to the embedded component of n which is obtained through two regular convolutional layers with convolution kernel of 1×1 and the variable d is the channel dimension. The term of non-local behavior takes all positions in the averaging operation instead of just looking into the local neighborhood, which only considers the group of pixels that is surrounding the target pixel. A non-local operation is a flexible building block and can be easily used together with convolution layers [41]. This allows to build a richer hierarchy that combines both non-local and local information by producing a higher post-filtering clarity and lesser loss of detail in the image denoising mechanism.

A random sampling of the deformed location is applied to an input feature map using convolution kernel (K) which is augmented with offsets $\{\Delta a_n | n = 1, \ldots, N\}$, where $N = |K|$ is the total number of sampling grid of the convolution kernel in the deformable convolution within DDC module. K is the sampling grid of convolution kernel with a receptive field size of 3 × 3 and dilation of one, where $K = \{(-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-1), (1,0), (1,1)\}$ and $N = 9$ in this basic setup having nine elements in this module. All sampled value are weighted by w for each pixel location a_0 on the the output feature map y as follows [6],

$$y(a_o) = \sum_{a_n \in R} w(a_n) \cdot x(a_o + a_n + \Delta a_n) \tag{3}$$

where a_n enumerates the locations in K and Δa_n represents the offset value. Figure 3 demonstrates the sampling on the offset locations $a_n + \Delta a_n$. As the offset Δa_n is typically fractional, (3) is implemented through bilinear interpolation as,

$$x(a) = \sum_{b} H(b, a) \cdot x(b), \tag{4}$$

where a denotes an arbitrary (fractional) location where $a = a_0 + a_n + \Delta a_n$, b enumerates all integral spatial locations in the feature map x, and $H(\cdot)$ is the bilinear interpolation function. The offset (Δa_n) is obtained through a regular convolutional kernel with the exact spatial resolution and dilation as the input feature map, with the exception of outputting channel dimension of $2N$ to correspond with the N two-dimensional offsets. The output feature and offsets are learned simultaneously during training phase at the regular convolutional layer using back propagation of gradient.

In the operation of the proposed DDC block, a regular convolutional layer is first applied over the input feature map x to obtain an offset field Δa_n by performing the denoised mechanism to filter and suppress noises as follows,

$$m_i(\Delta a_n) = \frac{1}{C(\Delta a_n)} \sum_{\forall j \in L} f(\Delta a_{n_i}, \Delta a_{n_j}) \cdot g(\Delta a_{n_j}), \tag{5}$$

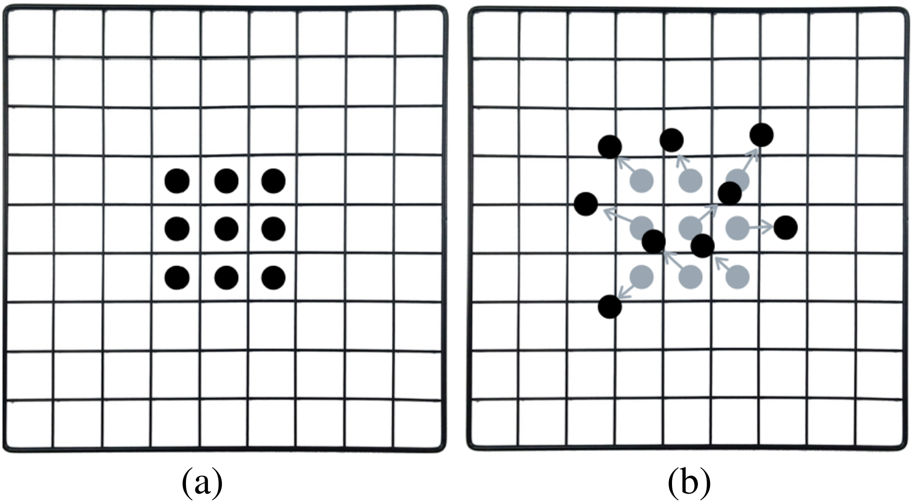

Fig. 3 Illustration of the sampling locations in 3 × 3 standard and deformable convolution. (a) regular fixed 3 × 3 sampling grid (black dots) of standard convolution. (b) deformed 3 × 3 sampling locations with offsets (grey arrows) in deformable convolution with the sequence formation started at the bottom left corner with position (-1,-1). The top right corner's position is (1,1) in the convolutional kernel (K)

where Δa_{n_i} represents the corresponding output position and Δa_{n_j} represents the possible position enumerated around the Δa_{n_i}.

The denoised offset field $m_i(\Delta a_n)$ is subsequently used to augment the sampling location using bilinear interpolation, resulting in an output feature map y as follows,

$$y(a_o) = \sum_{a_n \in R} w(a_n) \cdot x(a_o + a_n + m_i(\Delta a_n)). \tag{6}$$

This DDC module is very important to generate the denoised offset field before being augmented into the output feature map because the existence of noise in the input feature map could cause false positive offset, leading to unnecessary activation in the output feature map. Hence, denoised offset field provides the benefit of optimizing the augmentation of the sampling grid inside the convolution kernel while preventing unnecessary activation, resulting in a more accurate ROI.

4 Mask region-based Denoised deformable convolutional network

DDC module is proposed to improve the capability of CNNs to model unknown transformations or variations, by optimizing the augmentation of sampling location of convolution kernel using denoised offset field. The fixed receptive field size of convolution kernel is undesirable in high level CNN layer that encode semantics over spatial locations. Hence, DDC module is incorporated into stage 5 of Residual Convolution Neural Network (ResNet-101) as demonstrated in Fig. 4. Stage 5 is the high level feature representation where the local region is grouped for task segmentation. The incorporation of DDC into ResNet-101 produces a more accurate, cleaner and representative high level feature map of the actual crack, as compared to the original mask R-CNN. However, this incorporation will also leads to higher computational requirement. The feature map of all levels from stage 1 to stage 5 are passed to Feature Pyramid Network (FPN) to generate multiple feature map layers, containing better quality semantic information for object detection. FPN [24] is a feature extractor using a top-down architecture with lateral connection to build high-level semantics feature map at all scales.

The resulting high-level semantics feature map are passed on to Region Proposal Network (RPN) in order to propose candidate object bounding boxes. RPN [33] is an algorithm that takes in an image of any size as inputs and outputs a set of rectangular object proposals, each with a specific objectness score. The object proposals are generated based on the multiple scale and aspect ratio parameter specified in the network, where a total number

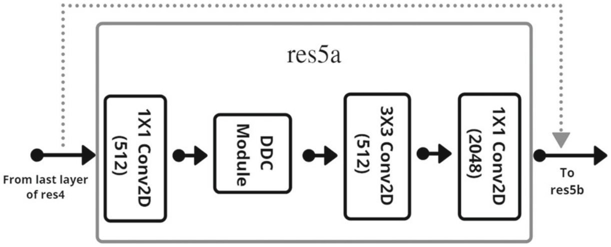

Fig. 4 Denoised Deformable Convolution module is incorporated in the stage 5 of ResNet-101 (res5a, res5b and res5c)

of proposals are generated for every pixel in an input image. The region proposals [2] generated by RPN are passed on to ROI align in order to map the proposals onto the feature diagram to get the Region of Interest (ROI). ROI align is a ROI mapping algorithm used to solve the misalignment problem faced by its predecessor, ROI Pooling, resulting in quantization loss that has a very negative effect in pixel-accurate tasks. In ROI align, bilinear interpolation is used to obtain the value of image on their exact pixel point whose coordinates is a floating number to solve the misalignment problem, resulting a more accurate mapping of ROI to the input image.

These ROIs are passed through fully connected layers in order to generate the ROI vectors. The ROI vectors are then passed through a predictor of two branches, each with a fully connected layer. One branch is for predicting the bounding box regression values whereas the other branch is for predicting the object class. In parallel with the existing branch for bounding box regression and classification, there is another branch for mask generation. The mask branch containing a fully convolutional network (FCN) which applies to every ROI, predicting a mask in pixel-specific manner [12]. Hence, the mask R-DDCN can be used to perform crack identification to locate the position, and classify the type of cracks.

5 Experimental setup

The proposed method is evaluated using cracks dataset, which contains of the crack images recorded from the Middle East Technical University Campus Buildings and published at the Mendeley Library [27]. This dataset contains two kind of images, i.e. image containing the crack data and images without any cracks under multiple surface finishing and illumination conditions without data augmentation in terms of random rotation or flipping. The sample images are illustrated in Fig. 5. The images containing crack consist of three types of cracks, i.e. longitudinal, transverse and crocodile cracks. Each class has 250 images, where 200 images are used for training and 50 images are used for validation. In total, there are 600 training images and 150 validation images for the three classes.

The measurement metrics that we used in this experiment is mean average precision (mAP), which is similar to PASCAL Visual Object Classes (VOC) 2007 challenge [8]. Mean average precision (mAP) is calculated as follows:

$$\text{mAP} = \frac{\sum_{q=1}^{Q} P_{avg}(q)}{Q} \times 100\%, \tag{7}$$

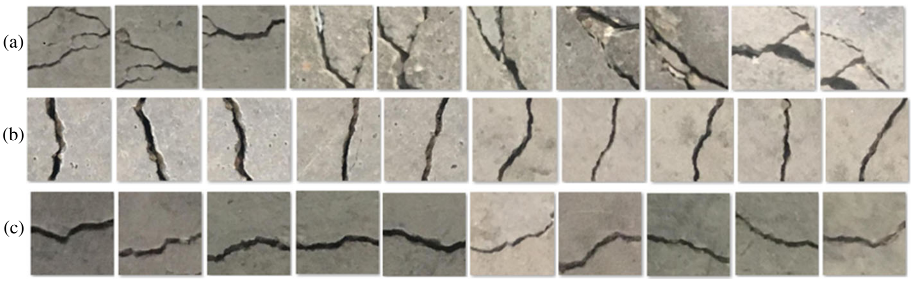

Fig. 5 Mendeley Library Dataset [27] for (a) Crocodile cracks, (b) Longitudinal cracks (c) Transverse cracks

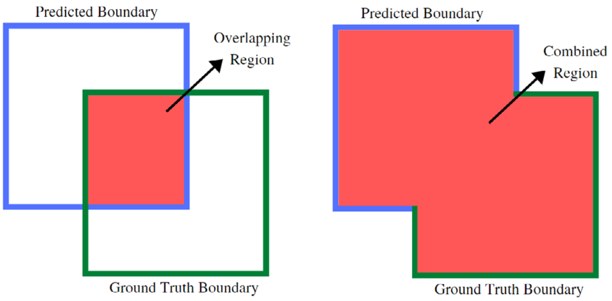

Fig. 6 Illustration of overlapping region and combined region for Intersection over Union (IOU)

where Q is the number of queries in the set, P_{avg} is the average precision (AP) of a given query (q) with an intersection over union (IOU) threshold value. mAP is only obtained after dividing the P_{avg} sum of all queries with the total number of queries in the test. The AP calculation is set based on the intersection over union (IOU) threshold value as follows:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}, \tag{8}$$

where the area of overlap represents the regions that overlaps between the predicted region and ground-truth region, and area of union represents the total combined region of predicted region and ground-truth region. The illustration of overlapping region and union region are shown in Fig. 6. This threshold is measured with the mean AP over three crack classes, which is longitudinal, transverse and crocodile crack, along with the background class. The classification rate can be measured as follows,

$$Acc = \frac{TP}{S_{test}} \times 100\%, \tag{9}$$

where TP is the True Positive where the type of cracks is predicted correctly and S_{test} is the total number of testing samples.

A pre-trained weight from the Common Objects in context (COCO) [23] is used to initialize the weights on ResNet feature extractor. ResNet-101 is implemented as the backbone for R-CNN network developed based on [12] to perform image classification and semantics segmentation on the cracks in the dataset. To modify configuration of mask R-CNN, the DDC module is inserted into all block of stage 5 after the activation on 1×1 convolution and before the regular 3×3 convolution layer on stage 5, as illustrated in Table 1. In Table 1, the sequential building layers are listed in the brackets, with the number of stacked blocks indicating outside the brackets. The output size is down-sampled along conv3_1, conv4_1 and conv5_1 with DDC blocks using a stride of 2.

As for the parameter setup, the hyper-parameters are set following the latest version of mask R-CNN [12]. During the training and inference, the input images are resized to "square" mode, which means the input images are resized and padded with zeros in order to get a square image of size 1024 pixels on each side of the resized square image. As for Region Proposal Network, the anchors setting is configured to 5 different scales (32, 64, 128, 256, 512) and 3 aspect ratios (0.5, 1, 2), generating 2k and 1k region proposals at non-maximum suppression threshold of 0.7 at training and inference respectively. The optimization is set as Stochastic Gradient Descent (SGD) with learning rate, weight decay and learning momentum as 0.001, 0.0001 and 0.9, respectively. The channel dimension, N

Table 1 Network Architecture in the modified ResNet-101

Layer Name	Output Size	ResNet-101
conv1	112×112	7×7, 64, stride=2
max pooling	56×56	3×3, stride=2
conv2_x	56×56	$\begin{bmatrix} 1\times 1, 64 \\ 3\times 3, 64 \\ 1\times 1, 256 \end{bmatrix}\times 3$
conv3_x	28×28	$\begin{bmatrix} 1\times 1, 128 \\ 3\times 3, 128 \\ 1\times 1, 512 \end{bmatrix}\times 4$
conv4_x	14×14	$\begin{bmatrix} 1\times 1, 256 \\ 3\times 3, 256 \\ 1\times 1, 1024 \end{bmatrix}\times 23$
conv5_x with DDC blocks	7×7	$\begin{bmatrix} 1\times 1, 512 \\ 1\times 1, \mathrm{N} \\ 3\times 3, 512 \\ 1\times 1, 2048 \end{bmatrix}\times 3$

in the DDC module is set as 512. In our experiment, the network is trained for 120 epochs with 100 iterations per epoch.

6 Result and discussion

The proposed DDC module is inserted into the ResNet-101 as one of the backbone of the mask R-DDCN to extract an irregular shape of cracks. For quantitative analysis, the performance of the proposed framework, mask R-DDCN is evaluated against the original mask R-CNN and the modified R-CNN with only deformable convlution (R-DCNN) in term of its training loss, validation loss, and mAP evaluation metrics with three IoU thresholding values, i.e. 0.5, 0.7 and 0.75. Figure 7 displays the training loss for mask R-CNN, R-DCNN and R-DDCN. In Fig. 7, the proposed mask R-DDCN demonstrated a similar decreasing trend in training loss along 120 epoch with the training loss of 0.3035. It is almost similar to the training loss of original mask R-CNN and R-DCNN, which is 0.2883 and 0.3045 respectively. Hence, this result showed that the loss incurred during the training process of the 600 training dataset for mask R-DDCN is acceptable, if not equal to the original and deformable convolution counterpart.

In Fig. 8, the proposed method demonstrated the lowest validation loss as compared to its original and deformable convolution counterpart. Mask R-DDCN can achieve validation loss of 0.4795 whereas original mask R-CNN and R-DCNN achieved validation loss of 0.4888 and 0.5060 respectively after 120 epochs. This implies that the mask R-DDCN is able to detect cracks more accurately for the validation dataset. As reported in [11], a model can be said to “overfit” to the trained dataset when the training loss is low but validation loss is high, whereas a model can be said to “underfit” when both training and validation loss is low. By comparing Figs. 7 and 8, it can be observed that the validation loss of original mask R-CNN and mask R-DCNN is higher than mask R-DDCN whereas the training loss for all method is almost identical. Hence, both mask R-CNN and R-DCNN model has a

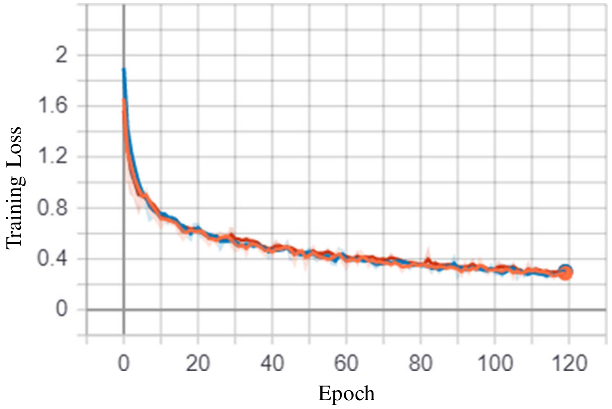

Fig. 7 Training loss for original mask R-CNN (orange), mask R-DCNN (blue) and mask R-DDCN (red) for 120 epoch

higher tendency of overfitting to the training dataset. In other word, the proposed mask R-DDCN model has a higher generalization ability, which is expected due to the DDC module that serves to improve the capability of the proposed method in modeling unknown transformation.

In Table 2, YOLOv3, original mask R-CNN, mask R-DCNN and mask R-DDCN were evaluated in term of mAP$_{50}$. Both YOLOv3 and Mask R-DDCN are different in term of their backbone of deep neural networks. YOLOv3 uses a variant of DarkNet [32] whereas Mask R-DDCN uses ResNet-101 as its backbone. As in mAP$_{50}$, mask R-DDCN could achieve 87.5% using the testing dataset. This indicates that proposed mask R-DDCN achieves a higher accuracy as compared to original Mask R-CNN, deformable convolution methods and YOLOv3 when the IoU threshold is 0.5, which means that the prediction boundary and ground truth boundary are overlapping more than 50%. The performance of proposed method improved the crack segmentation from 63.3% to 87.5% as compared to [21]. However, mAP$_{50}$ is insufficient to evaluate the accuracy as a higher IoU threshold should be

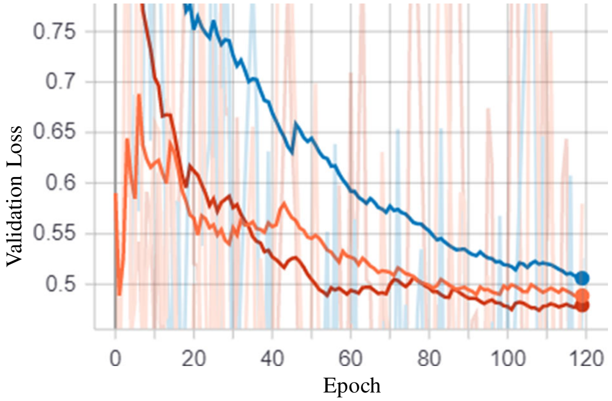

Fig. 8 Validation loss for original mask R-CNN (orange), mask R-DCNN (blue) and mask R-DDCN (red) for 120 epoch

Table 2 Number of trained parameters, percentage of mean average precision and classification rate comparison between YOLOv3, mask R-CNN with ResNet-50 and ResNet-101, R-DCNN and R-DDCN after 120 epoch

Method	#params	mAP_{50}	mAP_{70}	mAP_{75}	Acc
YOLOv3 [21]	40.55M	28.7	–	–	–
Mask R-CNN w/ ResNet-50 [21]	44.00M	63.3	–	–	–
Mask R-CNN w/ ResNet-101	63.73M	83.3	80.0	66.7	95.33
Mask R-DCNN	63.74M	83.3	73.3	66.7	89.33
Mask R-DDCN	63.74M	87.5	83.3	76.7	96.67

used for accuracy sensitive task like crack identification. As for the higher level evaluation metrics, which is mAP_{70}, original mask R-CNN achieved 80.0%, mask R-DCNN achieved 73.3% whereas mask R-DDCN achieved 83.3%. This result implies that the proposed method is able to predict more accurately than its original and deformable convolution counterpart as more prediction boundary and ground truth boundary are overlapping more than 70%. In addition, both mask R-CNN and mask R-DCNN achieved mAP_{75} of 66.7% whereas mask R-DDCN achieved 76.7%. This indicates that the proposed method can predict precise boundary than its original and deformable convolution counterpart as more prediction boundary and ground truth boundary are overlapping more than 75%. In term of classification rate, mask R-DDCN can achieve classification rate of 96.67%, which is higher than mask R-CNN and mask R-DCNN to identify the correct types of cracks. In term of number of trained parameters, the proposed mask R-DDCN has 63.74 million parameters which has only extra 0.01 million parameters as compared to mask R-CNN model. As a result, the proposed mask R-DDCN with the combination of denoised mechanism and deformable kernel can detect accurate crack boundary regardless of the shape and orientation of the cracks.

Figure 9 shows the comparison of validation result, which is the red region on crocodile, longitudinal and transverse crack between original mask R-CNN, mask R-DCNN and mask R-DDCN. Crocodile cracks refer to cracks with the appearance of a distinctive irregular shaped pattern of small cracks, which resemble the hide of crocodiles. Longitudinal cracks are cracks parallel to pavement's or concrete center line while transverse cracks are cracks at approximately right angles to the pavement's or concrete's center line. The red regions indicated in Figs. 9 and 10 are pixel-wise outputs generated as part of the output from Mask R-CNN, Mask R-DCNN and the proposed Mask R-DDCN. A detection bounding box is drawn to indicate the classification of correct cracks. As for the validation result on crocodile cracks, original mask R-CNN and mask R-DCNN demonstrated the signs of missing segmentation on the crack but it showed better crack segmentation using the proposed mask R-DDCN. On the other hand, the detection result on longitudinal crack displayed accurate segmentation by the proposed mask R-DDCN whereas the original and deformable counterpart shown signs of over-segmentation and missing segmentation respectively. Furthermore, the detection result on transverse crack displayed wrong classification of crack by original mask R-CNN and mask R-DCNN whereas the proposed mask R-DDCN showed good instance segmentation with correct crack classification. As a result of visual comparison with all three methods, Mask R-DDCN shows the most convincing results in crack segmentation and detection. Due to the deformation of DDC module is introduced in stage 5 of Mask R-DDCN by turning the fixed kernel into sampling location, higher computation is required to train the model with the ResNet-101 architecture as the backbone. However, the

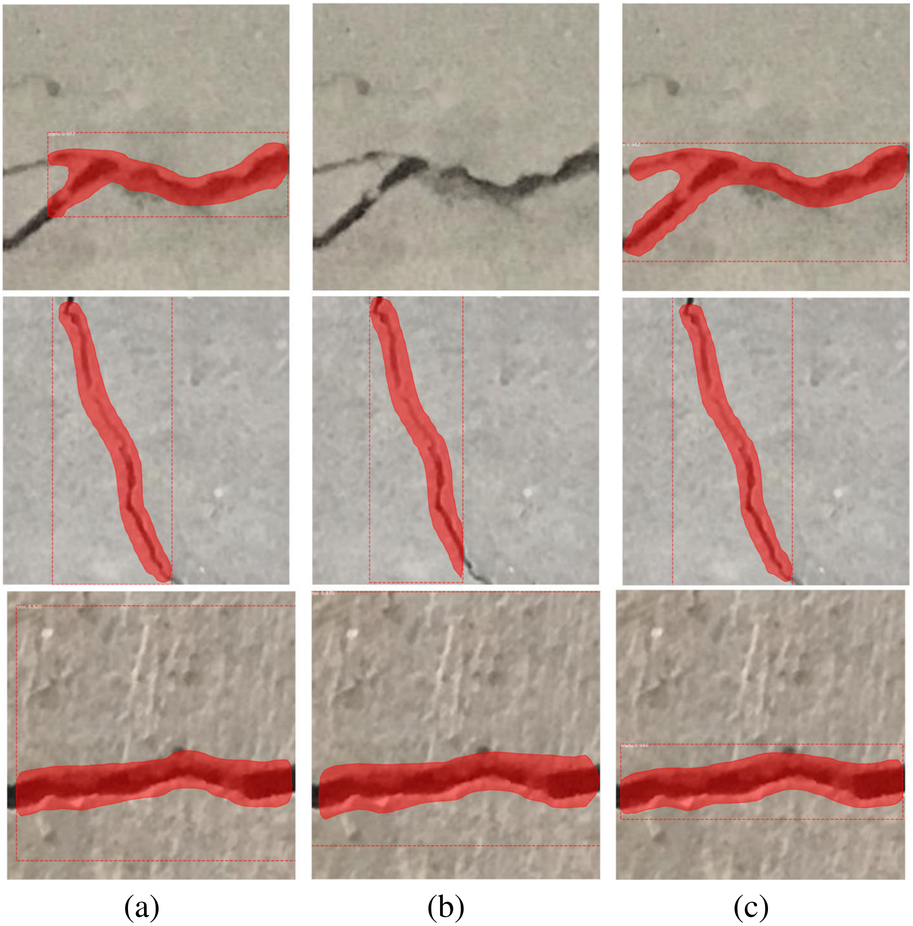

Fig. 9 Detection result of validation dataset on crocodile cracks, longitudinal cracks and transverse crack using (a) mask R-CNN, (b) mask R-DCNN and, (c) the proposed mask R-DDCN

computational cost could be addressed by improving hardware limitation such as graphical processing unit.

Mask R-DDCN has a higher capability in modeling unknown geometric transformation or variation compared to the original mask R-CNN due to the incorporation of DDC module in the backbone which augments the sampling points of convolution kernel with denoised offset field. The original mask R-CNN, mask R-DCNN and mask R-DDCN are trained using the crack dataset [27] and, the modeling capability of all methods are evaluated using random image from real world situation that contain geometric variation or transformation to a certain degree. In Fig. 10, the real-life cracks dataset is used to determine a more pragmatic approach evaluation and to obtain a more realistic comparison results with the multiple techniques, including our proposed R-DDCN model. The results from three images showing on the (a) and (b) gave incomplete and inaccurate crack detection and segmentation in the real-life images using mask R-CNN and mask R-DCNN respectively. As shown in Fig. 10c, mask R-DDCN could improve the crack detection and segmentation

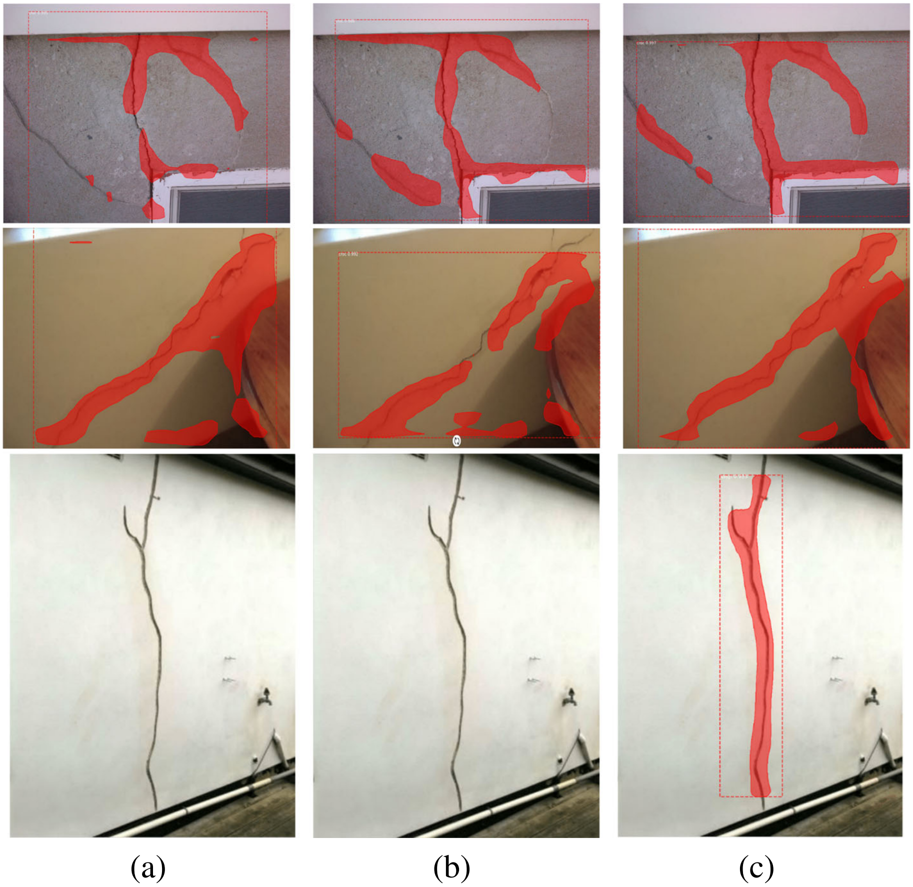

Fig. 10 Detection result of real application dataset on crocodile cracks and longitudinal crack using (a) mask R-CNN, (b) mask R-DCNN and (c) the proposed mask R-DDCN

due to the proposed method have a superior capability in modeling unknown variation or transformation over the original and deformable convolution counterpart.

7 Conclusion

Mask R-DDCN is proposed in this paper to extract irregular shape of crack identification in real time application. The denoised deformable convolution is introduced to optimize the augmentation of the sampling location of the convolution kernel with filtered offset. It is capable of improving capability of the CNNs in modeling unknown geometric transformation and improving the accuracy of the model with higher computational requirement. Experimental results shown that the proposed mask R-DDCN has lower validation loss and also improved the accuracy of mAP$_{75}$ from 66.7% to 76.7%. Mask R-DDCN can classify three type of cracks, i.e. crocodile, longitudinal and transverse cracks with 96.67% classification rate. In future work, more cracks with different geometric transformation or variation can be added to improve the capability of model in modeling unknown transformation. A more powerful backbone such as ResNeXt [42] could be adopted together with DDC for the improved performance in cracks identification.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Aire GE, Chimezie HN (2016) Comparison of non-destructive and destructive examinations in today inspection practices, 19th World Conference on Non-Destructive Testing 2016
2. Bai T, Pang Y, Wang J, Han K, Luo J, Wang H, Lin J, Wu J, Zhang H (2020) An optimized faster r-cnn method based on drnet and roi align for building detection in remote sensing images. Remote Sens 12(5):762
3. Boyat AK, Joshi BK (2015) A review paper: noise models in digital image processing, arXiv:1505.03489
4. Buades A, Coll B, Morel J-M (2005) A non-local algorithm for image denoising. In: 2005 IEEE Computer society conference on computer vision and pattern recognition (CVPR'05), vol 2. IEEE pp 60–65
5. Cho JS, Kim BH, Kim GS (2019) Application of deep learning-based crack assessment technique to civil structures, SMAR 2019 - Fifth Conference on Smart Monitoring Assessment and Rehabilitation of Civil Structures
6. Dai J, Qi H, Xiong Y, Li Y, Zhang G, Hu H, Wei Y (2017) Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 764–773
7. Dwivedi S, Vishwakarma M, Soni A (2018) Advances and researches on non destructive testing: A review. In: Materials today: Proceedings, vol 5. 1, pp 3691
8. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338
9. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2009) Object detection with discriminatively trained part-based models. IEEE Trans Pattern Anal Mach Intell 32(9):1627–1645
10. Geethalakshmi S (2018) A survey on crack detection using image processing techniques and deep learning algorithms. Int J Pure Appl Math 118(8):215–220
11. Ghasemian A, Hosseinmardi H, Clauset A (2019) Evaluating overfit and underfit in models of network community structure, IEEE Transactions on Knowledge and Data Engineering
12. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
13. Hoang ND (2018) Detection of surface crack in building structures using image processing technique with an improved otsu method for image thresholding, 4
14. Hu J, Wang Y, Chen J (2018) Pattern deep region learning for crack detection in thermography diagnosis system, Open Access Metallurgy Journal, vol 8, 8
15. Hu X, Yang K, Fei L (2019) Acnet: Attention based network to exploit complementary features for rgbd semantic segmentation. In: 2019 IEEE International conference on image processing (ICIP). IEEE pp 1440–1444
16. Jaderberg M, Simonyan K, Zisserman A et al (2015) Spatial transformer networks. In: Advances in neural information processing systems, pp 2017–2025

17. Jamaluddin N, Ayop S, Ibrahim MW, Boon K, Yeoh D, Shahidan S, Mohamad N, Chik TT, Ghafar NA, Ghani AA et al (2017) Forensic building: Deterioration and defect in concrete structures. In: MATEC web of conferences, vol 103. EDP Sciences, pp 02016
18. Janpreet S, Shashank S (2018) Road damage detection and classification in smartphone captured images using mask r-cnn, 11
19. Jeon Y, Kim J (2017) Active convolution: Learning the shape of convolution for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4201–4209
20. Lee CW, Jung H, Park G (2016) Automatic crack detection on pressed panels using camera image processing with local amplitude mapping, 1
21. Lee WQ, Lim KH, Lim CH, Lim WL, Yap HE (2020) Automated building crack identification using enhanced mask r-cnn. ASM Sci J, ICSCC2019 13(2):74–82
22. Lei Z, Fan Y (2017) Road crack detection using deep convolutional neural network, 10
23. Lin M, Belongie T-Y, Maire S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: European conference on computer vision. Springer, pp 740–755
24. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2117–2125
25. Lowe DG (1999) Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision, vol 2. Ieee, pp 1150–1157
26. Mei Q, Gül M, Azim MR (2020) Densely connected deep neural network considering connectivity of pixels for automatic crack detection. Autom Constr 110:103018
27. Özgenel ÇF, Sorguç AG (2018) Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In: ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction, vol 35. IAARC Publications, pp 1–8
28. Peng S, Jiang W, Pi H, Li X, Bao H, Zhou X (2020) Deep snake for real-time instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8533–8542
29. Prasanna P, Dana KJ, Gucunski N, Basily BB, La HM, Lim RS, Parvardeh H (2014) Automated crack detection on concrete bridges. IEEE Trans Autom Sci Eng 13(2):591–599
30. Prasanna P, Dana KJ, Lim R (2014) Automated crack detection on concrete bridges, IEEE Trans Autom Sci Eng, 10
31. Qu Z, Ju FR, Chen K (2018) Concrete surface crack detection with the improved pre-extraction and the second percolation processing methods, 7
32. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement, arXiv:1804.02767
33. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
34. Ryu E, Kang J, Lee J, Shin Y, Kim H (2020) Automated detection of surface cracks and numerical correlation with thermal-structural behaviors of fire damaged concrete beams. Int J Concr Struct Mater 14:1–12
35. Shan BH, Zheng SJ, Ou JP (2015) A stereovision-based crack width detection approach for concrete surface assessment, KSCE J Civ Eng, 20(2), 4
36. Singh K, Guruvayurappan S, Anand A (2019) Surface crack detection using computer vision powered by intel ai technologies, Wipro Limited, Tech Rep
37. Song W, Jia G, Zhu H, Jia D, Gao L (2020) Automated pavement crack damage detection using deep multiscale convolutional features, J Adv Trans. vol 2020
38. Su W, Lu L, Li B, Wang X, Dai J (2020) Deformable detr: Deformable transformers for end-to-end object detection, arXiv:2010.04159
39. Sun T, Gabbouj M, Neuvo Y (1994) Center weighted median filters: some properties and their applications in image processing. Signal Process 35(3):213–229
40. Tomasi C, Manduchi R (1998) Bilateral filtering for gray and color images. In: Sixth international conference on computer vision (IEEE Cat. No. 98 CH 36271). IEEE, pp 839–846
41. Wang X, Girshick R, Gupta A, He K (2018) Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7794–7803
42. Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1492–1500
43. Xie E, Sun P, Song X, Wang W, Liu X, Liang D, Shen C, Luo P (2020) Polarmask: Single shot instance segmentation with polar representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12193–12202
44. Xie C, Wu Y, Maaten Lvd, Yuille AL, He K (2019) Feature denoising for improving adversarial robustness. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 501–509

45. Xu H, Su X, Chen X (2019) Automatic bridge crack detection using a convolutional neural network, 7
46. Yang R, Yin L, Gabbouj M, Astola J, Neuvo Y (1995) Optimal weighted median filtering under structural constraints. IEEE Trans Signal Process 43(3):591–604
47. Yu C, Liu Y, Gao C, Shen C, Sang N (2020) Representative graph neural network. In: European conference on computer vision. Springer, pp 379–396
48. Zhang H-Z, Kim D-W, Kang T-K, Lim M-T (2019) Mift: a moment-based local feature extraction algorithm. Appl Sci 9(7):1503
49. Zhang R, Tang S, Zhang Y, Li J, Yan S (2019) Perspective-adaptive convolutions for scene parsing. IEEE Trans Pattern Anal Mach Intell 42(4):909–924
50. Zhang WY, Zhang ZJ, Qi DP, Liu Y (2014) Automatic crack detection and classification method for subway tunnel safety monitoring, Sensors 2014, 10
51. Zhao XF, Li SY (2018) Convolutional neural networks-based crack detection for real concrete surface, 3
52. Zou Q, Cao Y, Wang S (2011) Cracktree: Automatic crack detection from pavement images, 11

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.