Smart Structures and Systems, *Vol. 29, No. 1 (2022) 153-166*
https://doi.org/10.12989/sss.2022.29.1.153

One-step deep learning-based method for pixel-level detection of fine cracks in steel girder images

Zhihang Li [a], Mengqi Huang [b], Pengxuan Ji [c], Huamei Zhu* and Qianbing Zhang [d]

Department of Civil Engineering, Monash University, VIC 3800, Australia

(*Received April 15, 2021, Revised August 6, 2021, Accepted August 7, 2021*)

Abstract. Identifying fine cracks in steel bridge facilities is a challenging task of structural health monitoring (SHM). This study proposed an end-to-end crack image segmentation framework based on a one-step Convolutional Neural Network (CNN) for pixel-level object recognition with high accuracy. To particularly address the challenges arising from small object detection in complex background, efforts were made in loss function selection aiming at sample imbalance and module modification in order to improve the generalization ability on complicated images. Specifically, loss functions were compared among alternatives including the Binary Cross Entropy (BCE), Focal, Tversky and Dice loss, with the last three specialized for biased sample distribution. Structural modifications with dilated convolution, Spatial Pyramid Pooling (SPP) and Feature Pyramid Network (FPN) were also performed to form a new backbone termed CrackDet. Models of various loss functions and feature extraction modules were trained on crack images and tested on full-scale images collected on steel box girders. The CNN model incorporated the classic U-Net as its backbone, and Dice loss as its loss function achieved the highest mean Intersection-over-Union (mIoU) of 0.7571 on full-scale pictures. In contrast, the best performance on cropped crack images was achieved by integrating CrackDet with Dice loss at a mIoU of 0.7670.

Keywords: CNN; crack detection; data imbalance; feature extraction; loss function

1. Introduction

Fatigue cracks are signs of compromised condition that may lead to structural deficiency and damage the proper functionality of infrastructure facilities. Therefore, early identification of cracks is important for evaluating the urgency of repairing, strengthening or retrofitting the infrastructure to extend its service life. The traditional crack inspection generally relies on trained human operators to manually locate and mark the cracks, which can be accompanied by exposing workers to dangerous environments, time-consuming process, and objectivity-lacking results. On these grounds, there is a rising need of using vision-based methods to automate this process.

Computer vision-based techniques have been applied to infrastructure inspection and monitoring as non-contact condition assessment alternatives (Koch *et al*. 2015, Bao *et al*. 2019, Spencer *et al*. 2019, Huang *et al*. 2021). Approaches involved on crack recognition can be broadly classified into traditional digital image processing-based and deep learning-based methods. Crack identification based on traditional computer vision engages low- to intermediate-level image processing (Hutchinson and Chen 2006, Yamaguchi and Hashimoto 2009, Zhao *et al*. 2010, Yeum and Dyke 2015, Nayyeri *et al*. 2018). Low-level image processing algorithms such as hand-crafted feature descriptors considering texture and morphology were used to recognize targets from backgrounds (Abdel-Qader *et al*. 2003). Whereas the advanced image analysis that involves recognizing and classifying objects and their varied instances can be facilitated by engaging machine learning algorithms, such as support vector machine (SVM), decision trees and artificial neural network (ANN) (Prasanna *et al*. 2012, Lattanzi and Miller 2014, Lei *et al*. 2020). On top of them, segmentation techniques such as thresholding, or sometimes referred to as binarization can be implemented to differentiate foreground and background (Li *et al*. 2018, Nayyeri *et al*. 2018). Although techniques have achieved significant results in identification and recognition tasks, methods based on traditional digital image processing are still highly dependent on the design of feature descriptors and are therefore difficult to be generalized.

Moreover, deep learning has showcased state-of-the-art efficiency to identify cracks and the independence from hand-crafted features. As a subfield of machine learning based on neural networks, deep learning follows a general learning approach that automatically extracts patterns by decomposing a complex mapping target into a nested sequence of simple mappings (Goodfellow *et al*. 2016). Although being proposed several decades ago, deep learning only starts to gain real popularity when Graphical Processing Units (GPUs) were used for parallel training of

*Corresponding author, Ph.D. Student,
E-mail: Huamei.Zhu@monash.edu
[a] Ph.D. Student
[b] Ph.D. Student
[c] Ph.D. Student
[d] Ph.D.

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

deep neural networks in 2009. Henceforth, considerable research has been directed to leverage deep learning algorithms for computer vision tasks, such as image classification and object recognition, and superior performances were delivered in comparison with those based on traditional image processing techniques (Nash *et al*. 2018). CNN, as a supervised learning algorithm, has been especially useful for automatic image interpretation and pattern recognition. Deep learning-based crack segmentation methods can be broadly classified as two-step or one-step. The first approach, based on two-step architecture, combines object detection, consisted of region proposal and CNN-based classification, with semantic segmentation systems (Cha *et al*. 2017, 2018, Kang *et al*. 2020). The second approach is end-to-end and pixel-to-pixel segmentation achieved by fully convolutional networks (FCN) (Long *et al*. 2015), based on a down-sampling path consisted of only convolutional and pooling layers for crack detection and an up-sampling path for spatial inference (Yang *et al*. 2018, Dung 2019, Ren *et al*. 2020). FCN model shares similar feature extraction backbone with CNN-based classification model, but replaces the fully connected layers in the latter with convolutional layers; hence, 2D spatial information could be retained in up-sampling step to achieve pixel-level classification at a higher resolution. U-Net (Ronneberger *et al*. 2015) is one of the earlier networks modified based on FCN that inherited the encoder-decoder architecture but considered image segmentation with smaller size of annotated samples and contacting objects of the same class.

Deep learning-based methods have provided promising results on crack semantic segmentation on asphalt and concrete infrastructures (Cha *et al*. 2017, Dung 2019, Kang *et al*. 2020, Ren *et al*. 2020). However, these images are often present with minimal occurrence of noises and morphologically distinct crack instances traversed the surface. In contrast, cracks appeared on steel structures are hairline fractures. The pixel-level crack identification task thus faces two main challenges. The first one is the severe imbalance of foreground and background samples. For example, the numbers of positive samples (crack) and that of negative samples (background) in Fig. 1 are 16,109 and 16 million, respectively. That is to say, the prediction accuracy of more than 90% would be obtained even if the model classifies all pixels as background without identifying any crack. The second challenge regards the complicated background with various interferences like handwritings, shadows, and crack-like edges (Fig. 1). The two challenges arise from the fact that the crack inspection task is conducted on full-scale images captured from the real world. Apart from subjecting to different illumination and camera angles, multiple scene elements are contained, such as markers, paints and defacement that overlap with thin-width crack instances (Xu *et al*. 2018). The noisy background containing various color, brightness and morphology increases the segmentation difficulty and requires the adaptions made to existed models which are often proposed based on relatively clean pictures.

Aiming at the problem identified as ‘small object detection with complicated backgrounds’ for Project 1 in the IPC-SHM 2020 competition (Bao *et al*. 2021), here we proposed a structurally modified FCN model to extract features from noisy backgrounds and conducted comparison experiments to evaluate multiple loss functions regarding their abilities to address data imbalance during the model training. The paper is organized as follows. Section 2 explains the dataset preparation process and tools. Section 3 introduces the framework, methods engaged, and training environment. Section 4 discusses the evaluation of candidate loss functions, and the results of crack segmentation based on the proposed framework. Section 5 concludes the paper with the main findings.

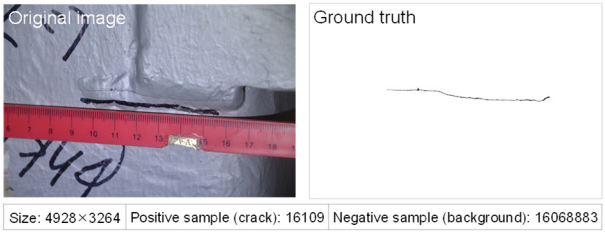


Fig. 1 Example image of fine crack on steel girder

2. Dataset preparation

A total of 200 images taken on bridge steel girders were provided by the competition committee, consisting of 120 image-label pairs and 80 raw images. Manual annotations of the additional 80 raw images were performed via LabelMe (Russell *et al*. 2008) to increase the data size. In the end, the dataset used in this study included 200 full-scale images with the size of 4,928 × 3,264 or 5,152 × 3,864 pixels, 160 of which were randomly selected to generate the training set, and the remaining 40 images were used for testing. When generating the training set, the size of full-scale images was adjusted to multipliers of 512 to be adapted to the proposed CNN model. The resized pictures were then cropped to create a total of 8,338 sub-images with the size of 512 × 512, including 928 crack images and 7,410 background-only images. Considering the significant imbalance of foreground and background samples, we intentionally removed the background-only images from the training set while all crack sub-images were retained. The full-scale images selected for testing got cropped in the similar way, which generated 274 crack and 1,832 background-only sub-images. In contrast to the training set, background-only sub-images retained for the test set in order to merge sub-image patches back into a full-scale one. In total, the training set was made up of 928 crack-containing images with size of 512 × 512, and the test set consisted of 274 crack-containing images plus 1,832 background-only patches with the same size. To further enrich the training dataset, ‘*Image Data Augmentation*’, a built-in function in Keras, was also applied to enlarge the original dataset by operations such as shifting, flipping, and zooming (Fig. 2).

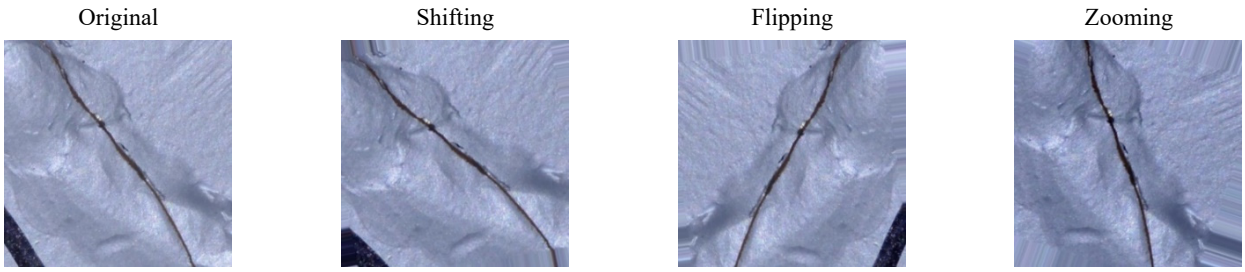


Fig. 2 Image data augmentation strategies used in this study

3. Methodologies

3.1 Overall framework of the proposed method

The proposed framework contains a trained CNN network as its core component to perform one-step pixel-level crack detection on steel girder images (Fig. 3). It receives full-scale images and produces predicted masks of the same size, with crack pixels marked as zero and background as one. The end-to-end workflow is summarized as followed.

(1) The original full-scale image I, with size of 4,928 × 3,264 or 5,152 × 3,864, is resized to image II with its width and height being multipliers of 512, namely with size of 4,608 × 3,072 or 5,120 × 3,584;
(2) The image II is cropped to a batch of 512 × 512 images;
(3) A trained segmentation CNN model takes the image batch III as inputs and produce predicted mask batch IV with crack labeled as 1 (white) and background labeled as 0 (black);
(4) The mask batch IV is merged and performed with color inversion to generate the mask V;
(5) The size of mask V is recovered to the same size as the original full-scale image to produce mask VI.

Intersection over union (IoU), the most commonly-used indicator to evaluate segmentation performance, is defined as Eq. (1).

$$IoU = \frac{A \cap B}{A \cup B} = \frac{A \cap B}{A \cap B + |A - B| + |B - A|} \tag{1}$$

where the overlapping degree of prediction set A and ground truth set B is measured.

Mean IoU (mIoU) was also calculated to average the prediction accuracies among different classes, as defined in Eq. (2).

$$mIoU = \frac{1}{k}\sum_{i=0}^{k-1} \frac{p_{ii}}{\sum_{j=0}^{k-1} p_{ij} + \sum_{j=0}^{k-1} p_{ji} - p_{ii}} \tag{2}$$

where k equals to 2 in this case, indicating the number of pixel class. Pixels can be classified as either 0 (background) or 1 (crack) in this crack segmentation task. The p_{ij} is the number of pixels belong to class i but are mis-predicted as class j, and p_{ii} and p_{ji} are also defined by that analogy. For example, p_{01} refers to the number of background pixels that are incorrectly identified as a crack.

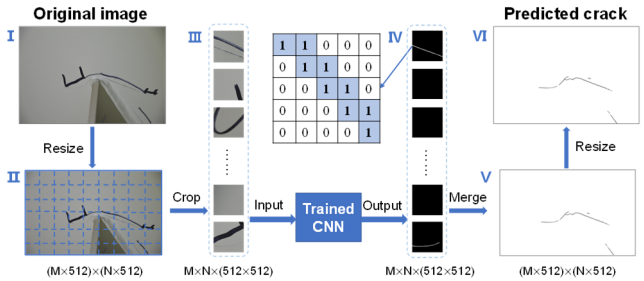


Fig. 3 Overall framework of the pixel-level crack detection method

3.2 Structure of the CNN model

The main structure of the model used to perform the task was a modified FCN. Instead of using fully connected layers for the object classification task, FCN-based networks replace them with additional convolutional layers and join with a deconvolutional network to restore the up-sampled image to the original input size. This study mainly made adaptations to this encoder-decoder structure for pixel-level and end-to-end image segmentation by introducing additional convolutions to the up-sampling path. More convolutions reduced the number of feature channels so that the produced feature map can concatenate with the same-level feature map from down-sampling. This ensures that the network can propagate with minimal loss of contextual information. Such modification was first adopted by U-Net (Fig. 4), with its U-shape architecture formed by having the up-sampling path almost mirrored the down-sampling path.

Inspired by U-Net, the CrackDet, with the FCN-based encoder-decoder architecture, was proposed (Fig. 5). The CrackDet, designed as an image segmentation network for very fine cracks, has its down-sampling path referencing the typical convolutional network VGG-16. This network consists of convolutional layers activated by ReLU, alternating with max pooling layers for feature extraction. However, instead of only relying on FPN, additional feature extraction modules are introduced, including the dilated convolutional network and SPP. This architectural modification has enhanced feature extraction performance and successfully increased the segmentation accuracy comparing to that achieved by adopting the original U-Net and FCN.

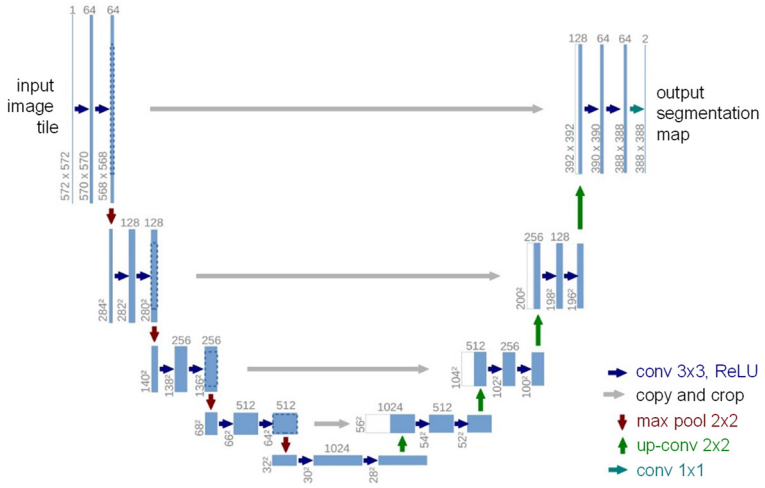


Fig. 4 U-Net consisted of symmetrical down-sampling and up-sampling architectures and connections, referenced from Ronneberger *et al*. (2015)

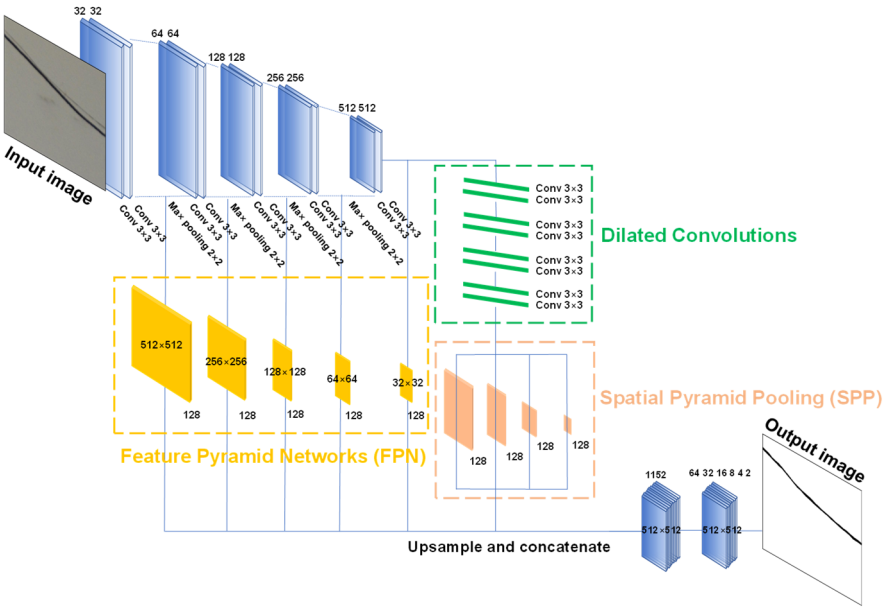


Fig. 5 CrackDet - the proposed crack segmentation network

As shown in Fig. 5, after the convolutional layer and max pooling layer, the features are input into eight dilated convolution layers with different dilated rates for additional feature extraction to achieve better feature extraction by deepening the network. Then the spatial pyramid pooling is applied to the decoder to obtain the representation of different sub-regions (He *et al*. 2014). Up-sampling and concatenation are carried out to form features that containing local and global information. The feature map after each two convolution layers in the encoder path is up-sampled after the spatial pyramid pooling, and connected with the feature map through feature pyramid networks (Lin *et al*. 2017). Finally, dense pixel-wise prediction is obtained by inputting the feature maps into seven convolutional layers. The following introduces the individual feature extraction modules and explains their incorporation into the overall model.

3.2.1 Dilated convolution (DC)

To be immune from the complicated backgrounds containing much crack-like chaos, a deeper backbone than that of the original U-Net was built to extract features in this study. Whereas, there was a trade-off between large receptive field and feature map resolution retaining, since the accumulation of features extracted by conventional convolution kernels, which are densely operated, may have overlaps between their receptive fields, causing the waste of computation and efficiency. Considering this problem, eight dilated convolution layers with different dilation rates were applied at the end of networks in the proposed CrackDet (Fig. 5). The dilated convolution technique uses the Atrous convolution kernel, which only has weights in certain positions, and fills other spaces in the kernel with zero (Yu and Koltun 2016). By geometrically increasing the dilation rate in continuous convolution layers, the receptive field

can be extended while ensuring the coverage. For instance, there are only nine weights (3 × 3) to be optimized in a convolution kernel of size 5 × 5, with a dilation ratio of 2. Compared with the conventional convolution method, the Atrous convolution kernel seems to be 'dilated', extracting features sparsely but still effectively.

3.2.2 Spatial pyramid pooling (SPP)

In a classic CNN structure, there is usually a full connection after the convolution layer at the end of the network. The number of features in a full connection layer is fixed; therefore, a fixed size of the input image is required. However, the input images are not always meeting the size requirement. Common transformation techniques, such as crop and warp, that change the ratio aspect of the image can result in image distortion. This problem has been tackled by the SPP layer introduced in He *et al*. (2014) on CNN model for object recognition (Fig. 6). An SPP layer is usually connected to the last convolutional layer of the network to generate a fixed size output regardless of the input image size. Attaching an SSP layer not only improves the scale invariance of the image but also reduces the over-fitting. Another advantage of SSP lies in the speed, which has been reflected during the training of our network. Since feature extraction by max-pooling is only performed on the output feature maps, the network takes less time to converge, compared with a conventional CNN model. Moreover, SPP is independent of CNN network designs and structures; thus, it could be easily placed at the end of the network by replacing the last pooling layer without affecting the remaining structure of the network.

3.2.3 Feature pyramid network (FPN)

In a CNN network, the shallow layers often focus on details, while the high-level layers pay more attention to semantic information that contributes directly to target detection. Therefore, the information delivery in a deep feature extraction network may cause detail omissions at the output layer. Especially in a small object detection task like this study, a conventional network may gradually lose its sensitivity to cracks that are too fine to be captured after several convolutional and pooling layers. In the proposed CrackDet, the FPN (Lin *et al*. 2017) was incorporated, which fuses the information of multi-scale feature maps and uses the output by the last layer to classify objects, so that high processing speed can be achieved while maintaining a low-level graphic memory usage.

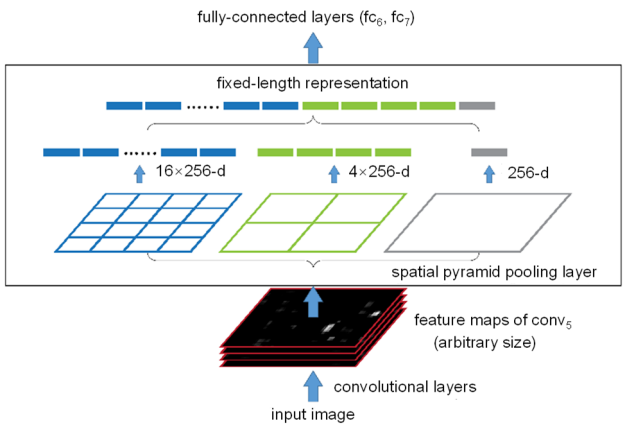


Fig. 6 The SPP layer that divides the feature maps into a fixed number of bins with sizes proportional to the image size, in order to achieve a fixed-length representation, referenced from (He *et al*. 2014)

3.3 Evaluation of loss functions

To explore the potentials of various loss functions to alleviate the imbalanced data distribution, here we evaluated loss functions specialized for small object detection, which are Focal, Tversky and Dice loss, in addition to the BCE loss used in the original U-Net. Loss is measured as the difference between predicted result and real value of a sample, and loss minimization is the driving force to optimize a CNN model. The four loss functions can be classified as cross entropy-based (BCE and Focal) and IoU-based (Tversky and Dice), according to their mathematical definitions. In this case, crack pixels (i.e., foreground) were set as positive samples (marked as '1') and the background pixels were set as negative samples (marked as '0') in loss calculation.

3.3.1 BCE loss

BCE loss is one of the most commonly used functions in image classification, detection and segmentation tasks, as defined with the formula in Eq. (3).

$$L = -\sum_{c=1}^{C} y_c \, log(p_c) \tag{3}$$

where C is the number of categories and p_c is the probability of predicted sample belongs to category C. The y_c is ground truth represented by a one-hot vector which has value of 1 and 0. The cross entropy is calculated by the average value of per-pixel loss, which is quantified separately and tends to make the model deviated to the category with a larger sample size.

3.3.2 Focal loss

In a case with unbalanced categories, the model will tend to only predict negative samples in BCE loss, so that the prediction probability of negative samples will be extremely high. The Focal Loss was proposed to improve the accuracy in the presence of overwhelming easily-classified background, by reshaping the BCE loss to Eq. (4) (Lin *et al*. 2018).

$$Fl = \begin{cases} -\alpha(1-p)^{\gamma}\log(p), & if\ y = 1 \\ -(1-\alpha)p^{\gamma}\log(1-p), & if\ y = 0 \end{cases} \tag{4}$$

By adding a probability-dependent parameter in front of the cross-entropy loss, the function down-weights the easy sample dynamically, thus strengthening the attention to hard samples. The best results were obtained at $\alpha = 0.25$ and $\gamma = 2.0$ (Lin *et al*. 2018), which were adopted in this study.

3.3.3 Dice loss

Rather than modifying the cross-entropy loss, some studies have also resorted to Intersection over Union (IoU)-

based loss functions, which adopt weighted term according to sample classes to address data imbalance. The Dice loss function calculated from the Dice coefficient was used by (Milletari *et al*. 2016) when performing 3D medical image segmentation. The Dice loss is calculated based on a generalized version of Dice coefficient and Jaccard coefficient (i.e., IoU coefficient), as defined in Eq. (5) (Tversky 1988).

$$D(A,B)=\frac{A\cap B+smooth}{A\cap B+\alpha|A-B|+\beta|B-A|+smooth} \quad (5)$$

where A is the predicted results and B is the ground truth. $|A\text{-}B|$ represents FP (false positive) set and $|B\text{-}A|$ represents FN (false negative) set. The balance between these two clusters can be controlled by adjusting α and β, thus affecting the recall rate and other performance indicators. Its format would be the same as the Jaccard coefficient when $\alpha=\beta=1.0$. Specifically, the Dice loss was calculated with Eq. (6) where $\alpha=\beta=0.5$ and $smooth=1.0$.

$$D=\frac{\sum_{i=1}^{N}p_{0i}g_{0i}+1.0}{\sum_{i=1}^{N}p_{0i}g_{0i}+0.5\sum_{i=1}^{N}p_{0i}g_{1i}+0.5\sum_{i=1}^{N}p_{1i}g_{0i}+1.0} \quad (6)$$

The p_{0i} and p_{1i} is the probability of pixel i either be the background or the crack, in the output of the softmax layer in this case. The g_{0i} and g_{1i} is 0 and 1 for a background and a crack pixel, respectively. When being used in deep learning model training, the parameter smooth is often added to the equation to accelerate the convergence.

3.3.4 Tversky loss

Similarly, the Tversky loss was designed to improve trade-off between precision and recall in segmenting highly unbalanced data by Salehi *et al*. (2017). It shares the IoU-based designing rule with the Dice loss, as defined in Eq. (5). Adjusting factors of $\alpha=0.3$ and $\beta=0.7$ performed best among parametric studies conducted by Salehi *et al*. (2017). The Tversky loss was calculated with Eq. (7) in this case.

$$T=\frac{\sum_{i=1}^{N}p_{0i}g_{0i}+1.0}{\sum_{i=1}^{N}p_{0i}g_{0i}+0.3\sum_{i=1}^{N}p_{0i}g_{1i}+0.7\sum_{i=1}^{N}p_{1i}g_{0i}+1.0} \quad (7)$$

It should be noted that Eq. (7) would exactly be the Dice loss function when setting both α and β as 0.5 (Sudre *et al*. 2017). By automatically increasing penalties of foreground pixels, certain kinds of loss functions, like the Dice loss (Sudre *et al*. 2017), have been proved to perform better on small object detection task than the most commonly-used BCE loss function. A summary of the parameter selections for the abovementioned loss functions can be found in Table 1. Comparison tests among these functions and against the BCE function used in the typical U-Net model are discussed in Section 4.

Table 1 Details of the loss function comparison test

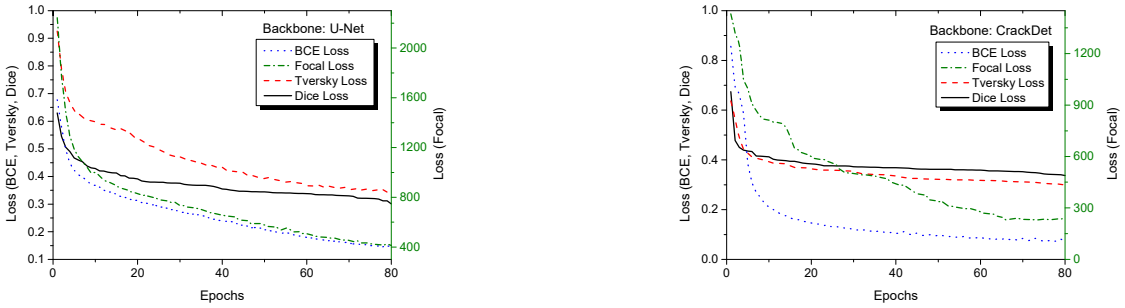
Loss function	Parameters
BCE	
Focal	$\alpha=0.25$, $\gamma=2.0$
Dice	$\alpha=\beta=0.5$, $smooth=1.0$
Tversky	$\alpha=0.3$, $\beta=0.7$, $smooth=1.0$

3.4 Deep learning model training details

Specifications of the deep learning platform used for crack identification framework proposed by this paper are Windows 10, Python 3.7.4, and Keras 2.13; platform hardware configuration is CPU Intel i7 9800X with 32GB of memory; configuration for graphics card includes one NVIDIA RTX2080Ti, 11GB of video memory, CUDA10.0, and NVIDIA cuDNN7.4.2 are used for GPU acceleration. Training details are shown in Table 2. Considering that graphical processing card (GPU) used by deep learning platform is not intentionally-designed for data-driven algorithm which consumes large number of data, the input structure of CrackDet, batch size and GPU memory allowance in training were adjusted to prevent the 'out of memory' error. Specifically, image patches with size of 512 × 512 were cropped from raw data, GPU memory allocation was set as growing adaptively with demand, and each deep learning model discussed in this study was trained for 2,000 steps and 80 epochs, with a batch size of one. For sample appearing frequency, namely steps and epochs, 2,000 steps were fixed for finding appropriate epochs based on learning loss. No significant loss reduction was observed after more than 70 epochs training for all models. Therefore, to benchmark the performance, training of all models stopped at 80 epochs. To mitigate overfitting, strategies including Adam optimizer and data augmentation were used in this study. Varying learning rate was used starting from 1e-5 and attenuated with epochs to about 5e-7. A total of 928 annotated crack images are used for training the model by a backward propagation algorithm. The trained models were tested with 2,434 images, including 274 crack images and 2,160 background images. Output patches were then merged into full size, in order to evaluate the model performance on full-scale images.

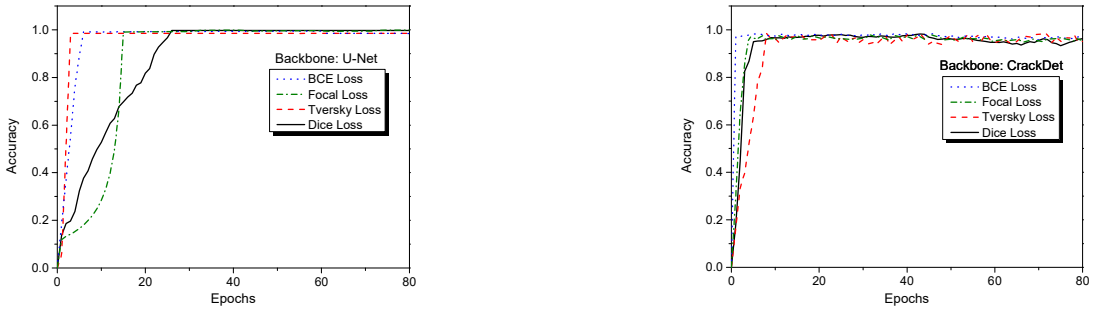
Table 2 Training details of deep learning models

Learning rate	1e-5~5e-7
Optimizer	Adam
Input size	512 × 512 × 3
Step	2000
Epoch	80
Batch size	1
Threshold	0.5



(a) U-Net (b) CrackDet

Fig. 7 Training loss curves of (a) U-Net; and (b) CrackDet with various loss functions



(a) U-Net (b) CrackDet

Fig. 8 Training accuracy curves of (a) U-Net and (b) CrackDet with various loss functions

4. Results and discussions

4.1 Overview of the comparison experiments

Aiming at the background interference, structural modifications were made to the backbone of U-Net, to improve the network's feature extraction ability. In addition, to address the data imbalance problem, multiple loss functions were evaluated regarding their potentials to train the network to recognize hard samples. A total of 8 models were trained and compared, which were backboned by different structures (the classical U-Net and the proposed CrackDet) and trained with various loss functions (BCE, Focal, Tversky, and Dice loss). The original U-Net, trained by BCE loss, here was taken as the baseline for comparison. The loss reduction and accuracy improvement curves during training process were depicted in Figs. 7 and 8 to ensure the convergence of all models. The IoU was used as main evaluation indicator, quantifying the proportion of pixels with the same label and position from the original and predicted mask to the total pixels. In this case, only the non-zero pixels would be counted during the IoU calculation since crack and background pixels were marked as 1 and 0 on binary images.

As illustrated in Fig. 3, the trained model firstly takes the cropped image with a resolution of 512 × 512 pixels as input, creates predicted masks with the same size where the crack is indicated by 1 and background is marked as 0. Then a batch of cropped masks is merged and rescaled to the original size. Therefore, here we examined the model performance based on two image clusters: the first cluster contains cropped images with cracks appeared on them; the second cluster contains the merged full-scale images output from the general framework (Table 3). The mIoU distributions of the 8 models are graphically demonstrated (Fig. 9) to reveal certain patterns, in which the improvements of loss functions are presented vertically while that of backbones can be observed horizontally. The baseline network (U-Net trained with BCE loss) obtained mIoU of 0.7206 and 0.7328 in cropped and full-scale images, respectively. When backboned by CrackDet, the model outperformed the original U-Net both in cropped and full-scale data with mIoU indicators achieving 0.7448 and 0.7459. Also, training with IoU-based loss functions (U-Net with Tversky or Dice loss) improved the network performance as well. Especially when using Dice loss, the

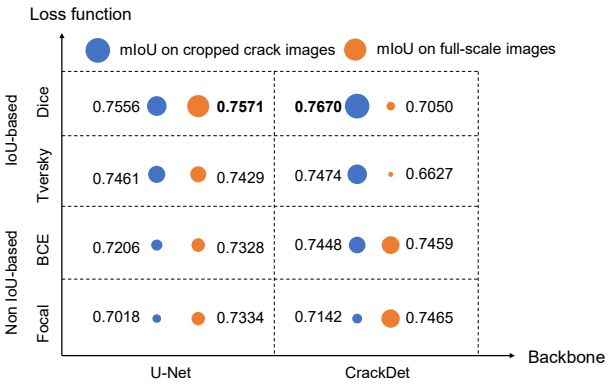


Fig. 9 The mIoU distribution of models with varied backbones and loss functions

Table 3 IoU of models with different loss functions and backbones

Loss function	Backbone	Cropped Crack images (512 × 512)			Original full-scale image		
		IoU		mIoU	IoU		mIoU
		1-class	0-class		1-class	0-class	
BCE	*U-Net	0.4537	0.9876	0.7206	0.4668	0.9987	0.7328
	CrackDet	0.5012	0.9885	0.7448	0.4930	0.9988	0.7459
Focal	U-Net	0.4165	0.9872	0.7018	0.4679	0.9989	0.7334
	CrackDet	0.4407	0.9878	0.7142	0.4941	0.9990	0.7465
Tversky	U-Net	0.5045	0.9877	0.7461	0.4872	0.9987	0.7429
	CrackDet	0.5082	0.9866	0.7474	0.3286	0.9969	0.6627
Dice	U-Net	0.5229	0.9883	0.7556	**0.5154**	**0.9988**	**0.7571**
	CrackDet	**0.5454**	**0.9885**	**0.7670**	0.4119	0.9981	0.7050

*Baseline: the original U-Net

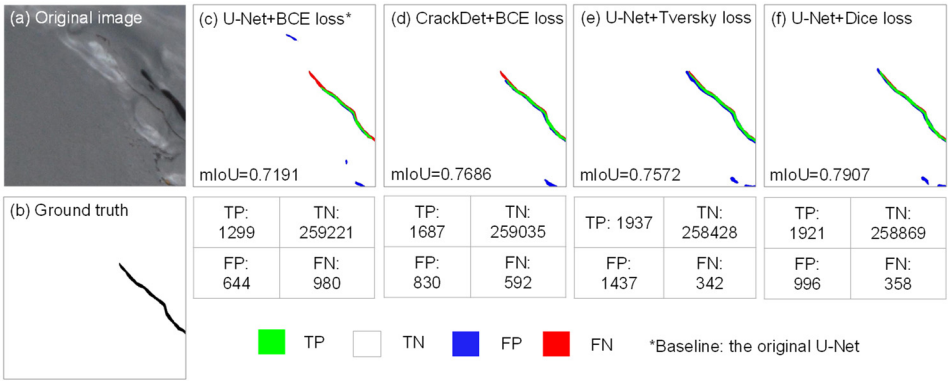


Fig. 10 Crack segmentation on crack image (512 × 512) by (c) original U-Net network; (d) backbone-improved U-Net network; and (e)-(f) loss function-improved U-Net networks

mIoU of U-Net was improved to 0.7556 and 0.7571 on crack and full-scale images, respectively.

To specify the improvements contributed by backbone and loss function modifications, pixel-wise predictions of a single crack image were evaluated, as shown in Fig. 10. Under the low illumination and handwriting interference (Fig. 10(a)), 1,299 true crack pixels were correctly recognized and the mIoU of 0.7191 was achieved by the original U-Net (Fig. 10(c)). Replacing the original backbone with the proposed CrackDet enabled the network to predict a higher number of true crack samples and improved the mIoU to 0.7686. Classic BCE loss was compared with loss functions (Focal, Tversky, and Dice loss) which were proposed for datasets with imbalanced sample distribution. Training the original U-Net with IoU-based loss functions enhanced the network's ability to detect more TP samples, such as 1,937 and 1,921 TP predictions with Tversky loss (Fig. 10(e)) and Dice loss (Fig. 10(f)). Although more FP samples were predicted as well, the mIoU values were improved to 0.7572 and 0.7907 by using Tversky and Dice loss, respectively. Although optimizing models with specialized loss functions cannot balance data distribution directly, this strategy improved model performance in an implicit way by enhancing penalty term of hard samples and therefore making more TP predictions, as can be seen in Fig. 10(c) and Figs. 10(e)-(f).

Confusion matrix results were averaged for all the 40 full-scale images to further demonstrate the improvement principle of modifications made in this study (Fig. 11). It should be noted that TN was not plotted in Fig. 11 since there are tens of millions background predictions and little differences (less than 1‰) were measured. The confusion matrix of a representative full-scale image was also calculated and demonstrated in Fig. 12, where the distinct

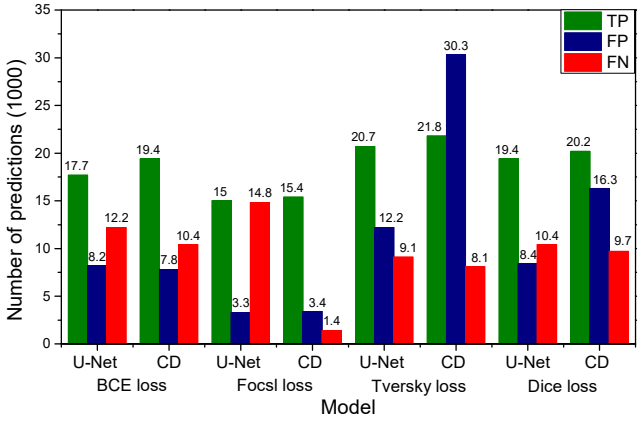


Fig. 11 Confusion matrix results of overall comparison experiments

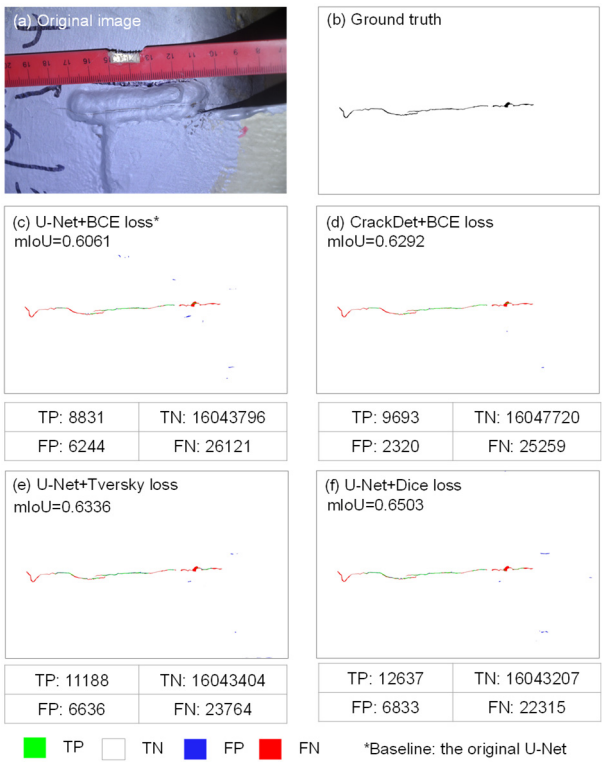


Fig. 12 Crack segmentation on full-scale image (4928 × 3264) by (c) original U-Net network; (d) backbone-improved U-Net network, and (e)-(f) loss function-improved U-Net networks

ways CrackDet and IoU-based loss functions worked were shown via quantification. Compared with crack images, full-scale images containing higher amount of background samples exacerbates the data imbalance. The improvement in mIoU of the CrackDet-backboned network (Fig. 12(d)) was mainly contributed by the suppression of FP predictions, which complies with the average level (Fig. 11). It should be noted that the IoU-based loss functions worked in a different way by increasing the tendency of predicting positive samples, where much more crack pixels were correctly identified (TP samples). In summary, both CrackDet and IoU-based loss functions improved the network capability in classifying pixels, yet via different ways.

4.2 Comparison experiments of loss functions

The performance of the outlined loss functions was studied, including both non IoU-based (Focal and BCE loss) functions and IoU-based (Tversky and Dice loss) functions. According to the vertical variation (Fig. 9), the performance of models, regardless of the backbone, on the cropped crack images was improved with the alteration of loss functions from non IoU-based to IoU-based ones. For example, compared with the U-Net model trained with Focal loss, that with Dice loss improved the mIoU on cropped crack images from 0.7018 to 0.7556. Similarly, for CrackDet, the mIoU was increased from 0.7142 when Focal loss was used to 0.7670 when Dice loss was used. Representative instances of predictions in cropped crack images made by CrackDet-backboned models with various loss functions were depicted in Fig. 13. Dice loss and Tversky loss, incorporating the IoU coefficient in their function formulations, have achieved better results compared to the other two functions with the capability to identify the main skeleton of cracks (Fig. 13).

The improvement contributed by IoU-based loss functions was also observed on full-scale images when taking U-Net as the backbone, in which the mIoU increased from 0.7334 to 0.7571 with the Dice loss. Average distributions of confusion matrix of full-scale images in test set were counted and plotted in Fig. 14, where significantly improved TP predictions of Tversky and Dice loss-trained models were observed. Examples of the segmentation maps produced by U-Net-backboned models with the four different loss functions on merged full-scale images are shown in Fig. 15. Although the original U-Net with BCE loss is robust enough to achieve high accuracy by being immune from the interferences of the ruler, handwriting and crack-like welding joints, as shown in Fig. 15(f), replacing the loss function with the Dice loss yields a significant IoU improvement of around 0.05. However, besides the increased TP samples predicted by models with IoU-based loss functions, there was also more background pixels incorrectly identified as positive (FP) on full-scale image (Figs. 15(e)-(f)). This overestimation of positive samples is also revealed in Fig. 14, and may contribute to the recorded decline of mIoU on full-scale images for CrackDet when IoU-based loss functions were used which will be further discussed in the following section. Moreover, unexpected performance reduction in cropped images of U-Net model with Focal loss was observed, compared with that trained by BCE loss. This may be caused by the inappropriate parameter selections in Eq. (4), which were recommended based on a dataset with much less class imbalance than this study. This problem might be addressed by fine-tuning the involved parameters in Focal loss to make it adaptive to specific dataset.

The optimization of loss functions enhanced segmentation results in this project by allocating higher weights on the penalty of wrongly-predicted crack pixels and hence addressing the sample imbalance problem effectively, but the adjusting parameters should be appropriately selected.

4.3 Comparison experiments of backbones

Aiming at the interferences such as illumination and handwritings, the feature extraction backbone, CrackDet, was designed to identify crack on complicted background. Ablation tests were conducted to study the necessity of each component on CrackDet.

4.3.1 Ablation tests of CrackDet

To validate the necessity of the proposed structural modifications and highlight the significance of the advanced feature extraction modules, a series of ablation tests were conducted via removing crucial components in an order, and testing the performances on full-scale images (Table 4). All models in ablation test were trained with Dice loss. CrackDet after removal of FPN, DC, and SPP was

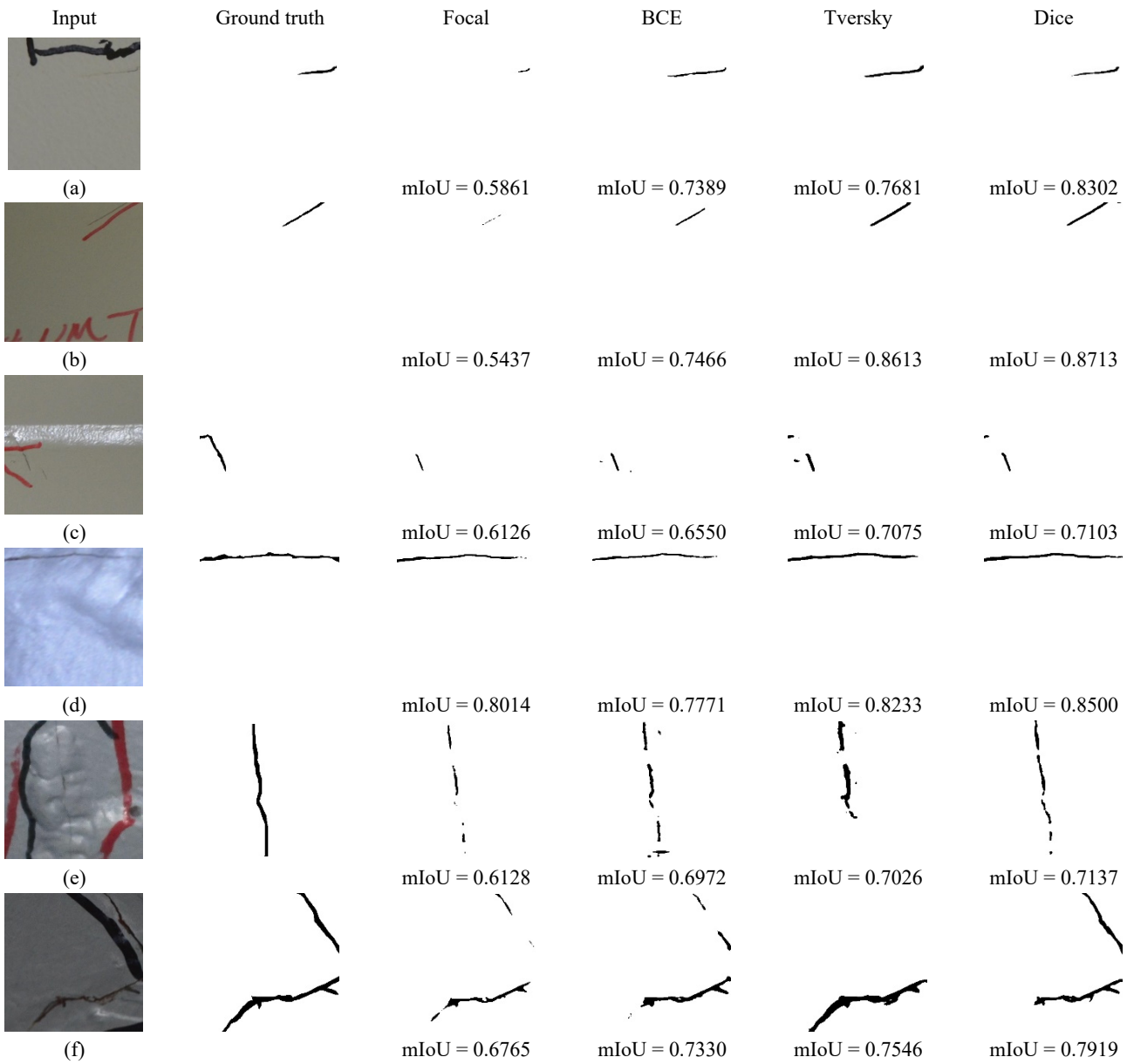


Fig. 13 Segmentation results on crack images of CrackDet models with different loss functions

regarded as the baseline, and the performance was improved gradually after adding these specialized feature extraction modules. The final outperformance of 25.27% was achieved by CrackDet compared with the baseline on full-scale images. Fig. 16 illustrates such performance increment, as the baseline network failed to recognize handwritings and markers in the background when robust feature extraction modules were missing.

4.3.2 Performance evaluation of U-Net and CrackDet

With respect to comparing results obtained by employing the two backbones (the original U-Net and the proposed CrackDet), it is found that the CrackDet network, incorporating FPN, SPP, and dilated convolution into its feature extraction module, performed better than the classical U-Net network in most cases (Table 3 and Fig. 9). Specifically, compared with the original version of U-Net, by optimizing network parameters and introducing additional modules, CrackDet achieved higher accuracy in cropped crack images (Fig. 17), in which case the mIoU was improved from 0.7206 to 0.7448 (Fig. 9). The SPP algorithm can process the input images with different sizes and aspect ratios, which may improve the scale invariance of images and reduce the over-fitting phenomenon during the model training (He *et al*. 2014). Also, when accompanied by FPN, the multi-scale problem in image segmentation can be solved. The dilated convolution expands the extraction range of the convolution kernel while maintaining the same parameters, and obtains a larger receptive field without additional parameters and calculation cost, which enhanced the feature extraction ability of the network, and resulted in higher recognition accuracy. The proposed CrackDet achieved higher IoU values than that of the classic U-Net in most cases, which indicated that by adopting additional feature extraction modules, a deeper

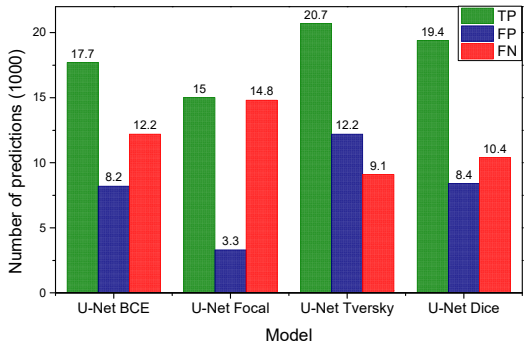


Fig. 14 Confusion matrix results of loss function comparison experiments

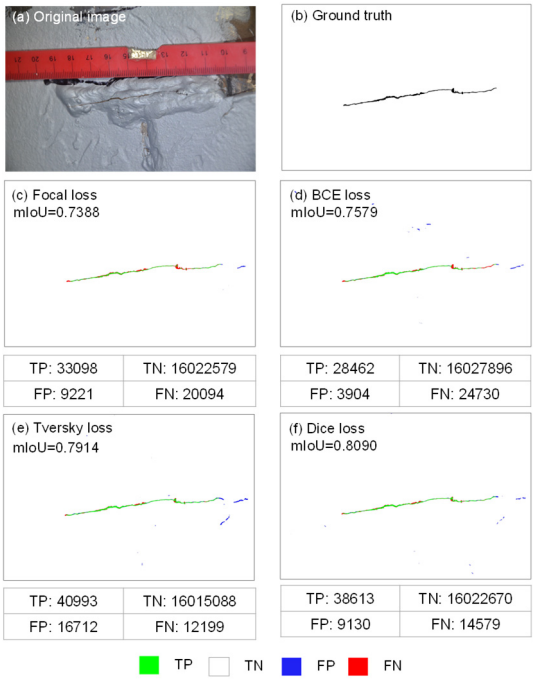


Fig. 15 Segmentation results on full-scale image of U-Net models with different loss functions

Table 4 Ablation tests of CrackDet

Structure	mIoU of full-scale images	Performance rate
CrackDet without FPN, DC, and SPP	0.5628	100.00% (baseline)
CrackDet without FPN	0.5865	104.21%
CrackDet without DC	0.5921	105.21%
CrackDet without SPP	0.6007	106.73%
CrackDet	0.7050	125.27%

*(FPN: feature pyramid network; DC: dilated convolution; SPP: spatial pyramid pooling)

network was created with stronger feature extraction ability and improved segmentation accuracy. In addition, by incorporating appropriate loss functions into the training process, the small object problem can be addressed properly.

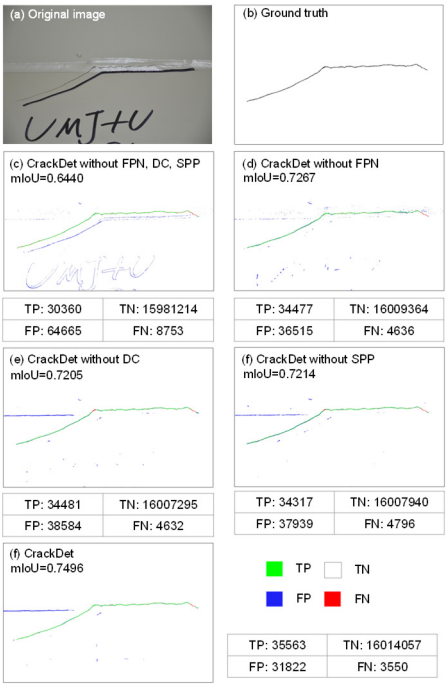


Fig. 16 Representative full-scale images predicted by ablation tests

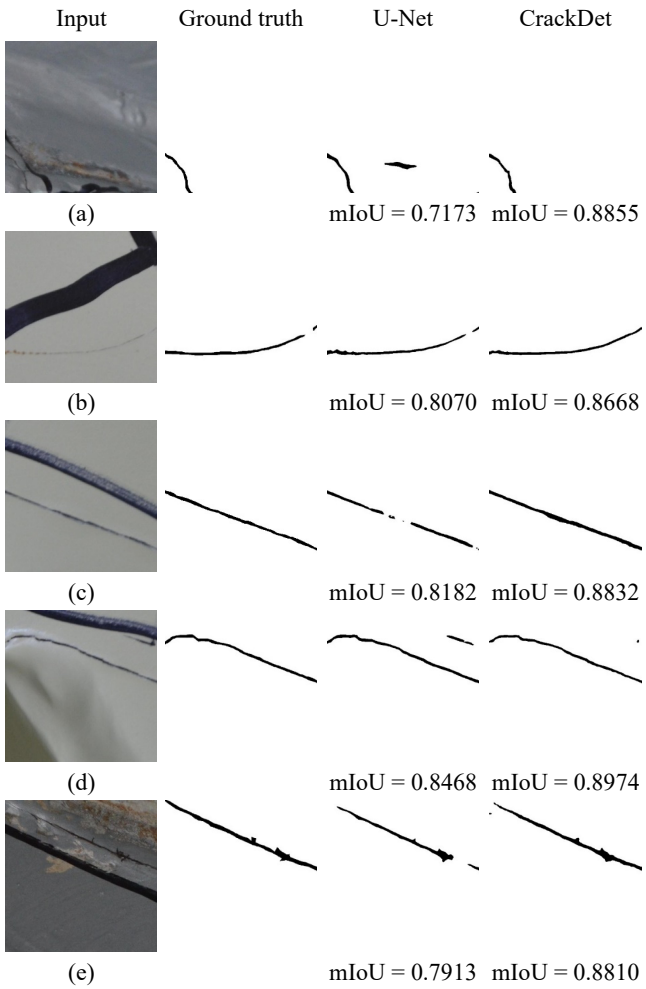


Fig. 17 Segmentation results on crack images of CrackDet and U-Net with Dice loss

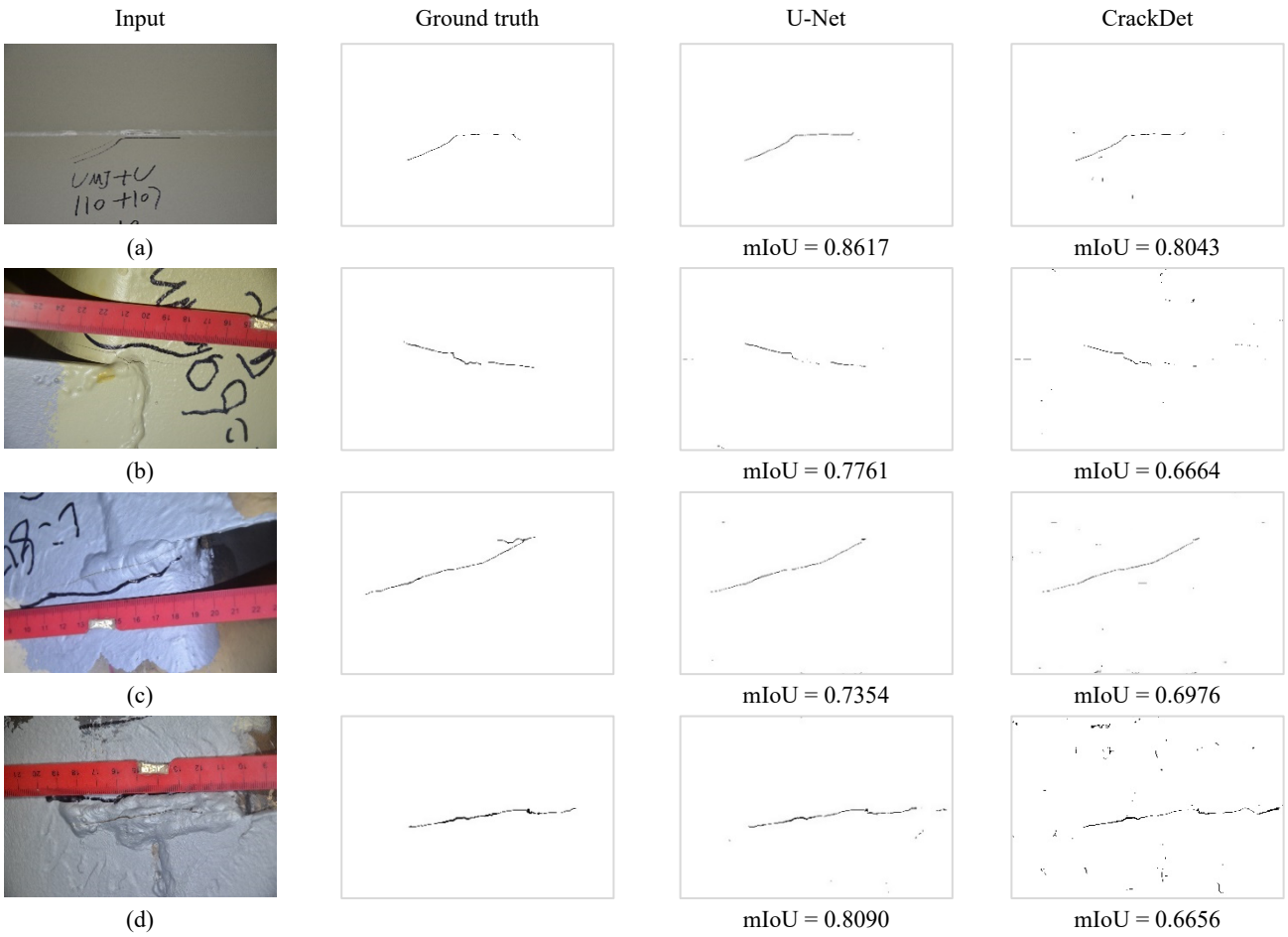


Fig. 18 Segmentation results on full-scale images of U-Net and CrackDet with Dice loss

It should be highlighted that the performance of CrackDet trained with IoU-based loss functions decreased on full-scale images, especially on the network trained by Dice loss (Fig. 18). The reasons causing such performance reduction on full-scale images were twofold. First, it is attributed to the penalty increment for the FN items in IoU-based loss functions, which results in a tendency of making more positive predictions, including FP predictions. In other words, the model trained with IoU-based loss functions is prone to recognize pixels as crack even on pure background regions, which has been revealed in Figs. 12(e)-(f) and Figs. 15(e)-(f). Second, CrackDet is capable of excluding interferences in noisy backgrounds, which results in much less FP predicted in the background images. The ability of CrackDet to distinguish background from foreground was significantly weakened when trained by IoU-based loss functions on full-scale images, and as a result, more FP predictions were made on background areas (Fig. 18). Thus, a stepwise adoption of these two strategies is worth exploring for crack segmentation on full-scale images, so that the efficiency of the CrackDet on background recognition and positive prediction tendency of the IoU-based loss functions can be leveraged to optimise the segmentation accuracy.

5. Conclusions

This study aims to address the challenges of data imbalance and complicated background when performing a deep learning-based fine crack detection task on full-scale steel girder images, by resorting to improvements of loss functions and feature extraction modules. Candidate loss functions, including the Focal, BCE, Tversky, and Dice loss were evaluated, and structural modifications were made and assessed using the U-Net as a baseline. The main findings and suggestions are listed according to the test results.

- The IoU-based loss functions (Dice and Tversky loss) outperformed the cross entropy-based loss functions (Focal and BCE loss) in addressing the small object detection challenge. The original U-Net, which used the BCE loss function in its training, gained a mIoU improvement of 6.4%, from 0.7206 to 0.7670, via taking the Dice loss as the alternative.
- With structural modifications to the feature extraction module, including dilated convolution, Spatial Pyramid Pooling layer and Feature Pyramid Network, the performance of the U-Net on crack images was improved from 0.7206 to 0.7448. The highest mIoU on crack images of 0.7670 was achieved by the CrackDet trained with Dice loss.

- Improvements of mIoU on full-scale images were recorded when CrackDet and IoU-based loss functions were adopted individually. Particularly, U-Net trained by Dice loss obtained the highest mIoU on full-scale images of 0.7571 among all models. Performance on full-scale images declined when the CrackDet was trained by IoU-based loss functions, as the simultaneous adoption of CrackDet and IoU-based loss functions caused the models to make more FP predictions on pure background images. Further improvement of crack segmentation performance on full-scale images would be expected if the CrackDet and IoU-based loss functions are employed in a two-step structure, which could exclude background images first using the proposed CrackDet, and then refining crack segmentation via the IoU-based loss functions.

The findings suggested that although CNN is a promising approach to automatically detect defects of civil infrastructure in captured images, its modules have to be discreetly designed and the mathematical principles of loss functions should also be fully considered depending on specific tasks. Further studies on CNN module manipulations and loss parameter adjustments should be carried out to explore the potentials of automated SHM, using deep learning in combination with computer vision techniques.

Acknowledgments

This study was supported by Monash University for the scholarships and the high-performance computation platform. The authors appreciate the organizations of the IPC-SHM 2020 ANCRiSST, Harbin Institute of Technology (China), and University of Illinois at Urbana-Champaign (USA) for their generously providing invaluable data from actual structures. The authors also would like to thank the chairs of IPC-SHM 2020 Prof. Hui Li, and Prof. Billie F. Spencer Jr for their leadership on the competition.

References

Abdel-Qader, I., Abudayyeh, O. and Kelly, M.E. (2003), "Analysis of edge-detection techniques for crack identification in bridges", *J. Comput. Civil Eng.*, **17**(4), 255-263. https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(255)

Bao, Y., Chen, Z., Wei, S., Xu, Y., Tang, Z. and Li, H. (2019), "The state of the art of data science and engineering in structural health monitoring", *Engineering*, **5**(2), 234-242. https://doi.org/10.1016/j.eng.2018.11.027

Bao, Y., Li, J., Nagayama, T., Xu, Y., Spencer Jr, B.F. and Li, H. (2021), "The 1st International Project Competition for Structural Health Monitoring (IPC-SHM, 2020): A summary and benchmark problem", *Struct. Health Monitor.*, **20**(4), 2229-2239. https://doi.org/10.1177/14759217211006485

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

Dung, C.V. (2019), "Autonomous concrete crack detection using deep fully convolutional neural network", *Automat. Constr.*, **99**, 52-58. https://doi.org/10.1016/j.autcon.2018.11.028

Goodfellow, I., Bengio, Y. and Courville, A. (2016), Deep learning, MIT press Cambridge.

He, K., Zhang, X., Ren, S. and Sun, J. (2014), "Spatial pyramid pooling in deep convolutional networks for visual recognition", *IEEE Transact. Pattern Anal. Mach. Intell.*, **37**(9), 1904-1916. https://doi.org/10.1007/978-3-319-10578-9_23

Huang, M.Q., Ninić, J. and Zhang, Q.B. (2021), "BIM, machine learning and computer vision techniques in underground construction: Current status and future perspectives", *Tunnell. Undergr. Space Technol.*, **108**, 103677. https://doi.org/10.1016/j.tust.2020.103677

Hutchinson, T.C. and Chen, Z. (2006), "Improved image analysis for evaluating concrete damage", *J. Comput. Civil Eng.*, **20**(3), 210-216. https://doi.org/10.1061/(ASCE)0887-3801(2006)20:3(210)

Kang, D., Benipal, S.S., Gopal, D.L. and Cha, Y.J. (2020), "Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning", *Automat. Constr.*, **118**, 103291. https://doi.org/10.1016/j.autcon.2020.103291

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B. and Fieguth, P. (2015), "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure", *Adv. Eng. Inform.*, **29**(2), 196-210. https://doi.org/10.1016/j.aei.2015.01.008

Lattanzi, D. and Miller, G.R. (2014), "Robust automated concrete damage detection algorithms for field applications", *J. Comput. Civil Eng.*, **28**(2), 253-262. https://doi.org/10.1061/(asce)cp.1943-5487.0000257

Lei, B., Ren, Y., Wang, N., Huo, L. and Song, G. (2020), "Design of a new low-cost unmanned aerial vehicle and vision-based concrete crack inspection method", *Struct. Health Monitor.*, **19**(6), 1871-1883. https://doi.org/10.1177/1475921719898862

Li, L., Wang, Q., Zhang, G., Shi, L., Dong, J. and Jia, P. (2018), "A method of detecting the cracks of concrete undergo high-temperature", *Constr. Build. Mater.*, **162**, 345-358. https://doi.org/10.1016/j.conbuildmat.2017.12.010

Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. (2017), "Feature pyramid networks for object detection", *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp. 2117-2125. https://doi.org/10.1109/CVPR.2017.106

Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P. (2018), "Focal loss for dense object detection", *Proceedings of 2017 IEEE International Conference on Computer Vision* (*ICCV*), pp. 2980-2988. https://doi.org/10.1109/ICCV.2017.324

Long, J., Shelhamer, E. and Darrell, T. (2015), "Fully convolutional networks for semantic segmentation", *IEEE Transact. Pattern Anal. Mach. Intell.*, **39**(4), 640-651. https://doi.org/10.1109/CVPR.2015.7298965

Milletari, F., Navab, N. and Ahmadi, S.A. (2016), "Fully convolutional neural networks for volumetric medical image segmentation", *Proceedings of 2016 Fourth International Conference on 3D Vision* (*3DV*), pp. 565-571. https://doi.org/10.1109/3DV.2016.79

Nash, W., Drummond, T. and Birbilis, N. (2018), "A review of deep learning in the study of materials degradation", *npj Materials Degradation*, **2**(1), 1-12. https://doi.org/10.1038/s41529-018-0058-x

Nayyeri, F., Hou, L., Zhou, J. and Guan, H. (2018), "Foreground–background separation technique for crack detection", *Comput.-Aided Civil Infrastr. Eng.*, **34**(6), 457-470. https://doi.org/10.1111/mice.12428

Prasanna, P., Dana, K., Gucunski, N. and Basily, B. (2012), "Computer-vision based crack detection and analysis", *Proceedings of SPIE - The International Society for Optical Engineering*, Vol. 8345, pp. 834542. https://doi.org/10.1117/12.915384

Ren, Y., Huang, J., Hong, Z., Lu, W., Yin, J., Zou, L. and Shen, X. (2020), "Image-based concrete crack detection in tunnels using deep fully convolutional networks", *Constr. Build. Mater.*, **234**, 117367. https://doi.org/10.1016/j.conbuildmat.2019.117367

Ronneberger, O., Fischer, P. and Brox, T. (2015), "U-net: Convolutional networks for biomedical image segmentation", *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234-241. https://doi.org/10.1007/978-3-662-54345-0_3

Russell, B.C., Torralba, A., Murphy, K.P. and Freeman, W.T. (2008), "LabelMe: a database and web-based tool for image annotation", *Int. J. Comput. Vision*, **77**(1-3), 157-173. https://doi.org/10.1007/s11263-007-0090-8

Salehi, S.S.M., Erdogmus, D. and Gholipour, A. (2017), "Tversky loss function for image segmentation using 3D fully convolutional deep networks", *International Workshop on Machine Learning in Medical Imaging*, pp. 379-387. https://doi.org/10.1007/978-3-319-67389-9_44

Spencer Jr., B.F., Hoskere, V. and Narazaki, Y. (2019), "Advances in computer vision-based civil infrastructure inspection and monitoring", *Engineering*, **5**(2), 199-222. https://doi.org/10.1016/j.eng.2018.11.030

Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S. and Cardoso, M.J. (2017), "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations", In: *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pp. 240-248. https://doi.org/10.1007/978-3-319-67558-9_28

Tversky, A. (1988), "Features of similarity", *Readings in Cognitive Science*, **84**(4), 290-302. https://doi.org/10.1037/0033-295X.84.4.327

Xu, Y., Li, S., Zhang, D., Jin, Y., Zhang, F., Li, N. and Li, H. (2018), "Identification framework for cracks on a steel structure surface by a restricted Boltzmann machines algorithm based on consumer-grade camera images", *Struct. Control Health Monitor.*, **25**(2), e2075. https://doi.org/10.1002/stc.2075

Yamaguchi, T. and Hashimoto, S. (2009), "Fast crack detection method for large-size concrete surface images using percolation-based image processing", *Mach. Vision Applicat.*, **21**(5), 797-809. https://doi.org/10.1007/s00138-009-0189-8

Yang, X., Li, H., Yu, Y., Luo, X., Huang, T. and Yang, X. (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

Yeum, C.M. and Dyke, S.J. (2015), "Vision-based automated crack detection for bridge inspection", *Comput.-Aided Civil Infrastr. Eng.*, **30**(10), 759-770. https://doi.org/10.1111/mice.12141

Yu, F. and Koltun, V. (2016), "Multi-scale context aggregation by dilated convolutions", arXiv preprint arXiv:1511.07122.

Zhang, L., Yang, F., Zhang, Y.D. and Zhu, Y.J. (2016), "Road crack detection using deep convolutional neural network", *Proceedings of IEEE International Conference on Image Processing* (*ICIP*), Phoenix, AZ, USA, pp. 3708-3712. https://doi.org/doi/10.1109/ICIP.2016.7533052

Zhao, H., Qin, G. and Wang, X. (2010), "Improvement of canny algorithm based on pavement edge detection", *Proceedings of 2010 3rd International Congress on Image and Signal Processing*, Yantai, China. https://doi.org/10.1109/CISP.2010.5646923