Advanced Steel Construction – Vol. 22 No. 3 (2026) 255–262
DOI:10.18057/IJASC.2026.22.3.1

FATIGUE CRACK RECOGNITION AND QUANTIFICATION UNDER COMPLEX BACKGROUND BASED ON REGION DETECTION

Yu-Hang Liu, Zhi-Yuan Yuan Zhou *, Bo-Hai Ji, Fei-Ya Rong and Tian Gao

School of Civil and Transportation Engineering, Hohai Univ., No.1, Xikang Rd., Nanjing 210000, China
** (Corresponding author: E-mail: yz-zhiyuan@hhu.edu.cn)*

ABSTRACT

Accurate measurement of fatigue crack length is essential for routine maintenance of steel bridges, and automated technologies have already been applied in maintenance operations. However, conventional image processing strategies suffer from complex and multi-source interference in steel box girders. This study proposed a two-stage method combining region recognition and morphological processing to diminish noise in fatigue crack images, thereby improving the accuracy of crack recognition and length measurement. The object-recognition-former (OBFormer) model was built to detect and isolate the fatigue crack regions, leveraging the advanced feature extraction module and localization capabilities. Optimized Canny algorithm was utilized to suppress noise and extract the skeletal line of the fatigue crack from detected regions. The length of the crack was calculated by modifying the coordinate system of the crack plane and quantifying the number of pixels along the skeletal line. Experimental results showed that the OBFormer model effectively handled complex background interference and accurately detected the fatigue crack regions, with a mean average precision (mAP) of 91%. After morphological processing, the length of the crack skeleton was accurately calculated with a relative error of less than 5%. The proposed methodology provides a reliable and efficient solution for fatigue crack measurement, offering significant potential for real-world applications in structural health monitoring and bridge maintenance.

ARTICLE HISTORY

Received: 25 March 2025
Revised: 23 July 2025
Accepted: 2 August 2025

KEYWORDS

Steel box girder;
Fatigue crack;
Computer vision;
Deep learning

1. Introduction

Orthotropic steel decks (OSDs) have been widely used in long-span bridges for their impressive spanning capabilities[1], whereas they are particularly vulnerable to fatigue cracks under the high load traffics[2]. Fatigue cracks could significantly undermining both mechanical performance and durability of bridges[3–5]. As the number of cracks continues to increase, the maintenance pressure correspondingly intensifies. Based on computer vision, automatic defect inspection systems had been developed to replace manual inspection. Particularly, crack detect model based on neural networks had been proved to be applicable to engineering practice, and showed its accuracy and high efficiency[6–8].

In early stage, computer vision tasks focused on objection detection which were applicable to the pavement crack detection challenge and had gotten great results[9]. For example, Chu et al.[10] proposed the Ghost-YOLO model to address the heavy workload associated with crack detection in CNNs, achieving mean average precision of 84.77%. Ye et al.[11] applied the YOLOv7 architecture to concrete damage detection and found that the model demonstrated strong crack detection capabilities in complex environment. With the rapid development of computer vision technology, the focus of crack detection task was moved to semantic segmentation. Semantic segmentation could acquire the accurate geometrical shape through pixel-level classification. This method showed great performance in concrete crack detection[12–15].

Compared to pavement cracks, fatigues cracks in OSDs show undistinct visual characteristics mainly due to the small width. In the meantime, the surfaces of OSDs exhibit noise which shows a resemblance to the visual characteristics of fatigue cracks and reduces the accuracy of neural networks. Therefore, semantic segmentation method showed some problems when it was applied to crack detection on OSDs images. Specifically, some models misclassified small objects with visual features similar to cracks. For example, Dong et al. [16] demonstrated that edges of weld lines might be detected as small fatigue crack fragments by using the proposed pixel-level fatigue crack segmentation method. Xu et al.[17] proposed fusion convolutional neural network, which had a small probability of misclassifying handwritten notes as fatigue cracks.

This study proposes a method for recognizing and quantifying fatigue cracks in OSDs to overcome interference targets while considering model lightweight. OBFormer model is designed and employed to identify cracks regions and to extract localized images. An optimized edge detection algorithm is applied to obtain the crack skeleton. Finally, the number of skeletal pixels is counted to calculate the crack length from distortion-corrected 2d image.

2. Crack detection on 2d image

2.1. Architecture of OBFormer

OBFormer is divided into encoder part and decoder part. Encoder is composed of multiple Transformer blocks. Each block outputs a feature map *Feat_i,* which is passed to next block as well as the decoder. Decoder mainly consists of Up-Down feature fusion (UDFF) module and classification-repression-interaction (CRI) module. In training progress, images are divided into multiple overlapping patches in the Overlap Patch Embedding (OPE) module, and each patch is encoded into a vector representation. After processing through the OPE module, the feature maps are fed into the Transformer block to compute the different-level feature maps. UDFF module adjusts the sizes and channels of deep-level feature maps and then fuse these features. The processed feature maps are transmitted to CRI module to extract discriminative features. All the features from different layer are concatenate to form a comprehensive feature vector to calculate the output.

Transformer block consists of Efficient Self-Attention module (ESA) mechanism, Mix Feedforward Network (Mix-FFN), and Overlapped Patch Merging (OPM) module. The mathematic progress is shown in **Fig. 1**.

The ESA[18] mechanism is based on the Self-Attention (SA) mechanism[19]. Feature maps are firstly flattened into sequence, and then the Query (Q), Key (K), and Value (V) matrices are computed to establish relationships between any two pixels. These matrices are then passed to the Mix-FFN function to generate the deep-level feature maps. Subsequently, the overlap patch merging (OPM) module reorganizes the metrics back into feature maps. The ESA mechanism is estimated as:

$$\text{Attention}\left(Q,K,V\right)=\text{softmax}\left(QK^{T}/\sqrt{d_{K}}\right)V \tag{1}$$

where Q, K, and V are tensors describing the queries, keys, and values respectively, and d_K is dimension of K. The Q, K, and V have the same dimensions of $N\times$c, where $N=h\times w.$ is the length of the feature sequence. To improve the computational efficiency of the self-attention mechanism, ESA introduces a reduction ratio R to reduce the spatial dimensions of features sequence as follows:

$$\hat{K}=\text{Reshape}\left(N/R,C\times R\right)\left(K\right) \tag{2}$$

$$K=\text{Linear}\left(C\bullet R,C\right)\left(\hat{K}\right) \tag{3}$$

Where K is sequence to be reduced, Reshaped(N/R, $C\bullet R$)(K) refers to reshape K to the one with shape from $N\times C$ to (N/R) × ($C\bullet R$), then the new K can obtain with the dimension of (N/R) × C through a linear transformation. As a result, the complexity of self-attention is reduced from O(N^2) to O(N^2/R).

Positional code is fixed in traditional transformer [20]. When the test resolution is different from that the training one, the positional code needs to be interpolated, which lead to the drop of accuracy. Mix-FFN consider the effect of zero padding to the leak location information by directly using a 3×3 convolution in the feed-forward network (FFN). Mix-FFN is formulated as:

$$\text{Mix-FFN}(x) = \text{MLP}(\text{GELU}(\text{Conv}3\times3(\text{MLP}(x))) + x \quad (4)$$

where x is the feature from the ESA module. Mix-FFN mixes a 3 × 3 convolution and an MLP into each FFN.

The OPM module is based on Patch Merging (PM) operation in Swin Transformer[21]. The PM module was initially designed to combine non-overlapping image or feature patches, which lacks the ability to preserve continuity between patches. To address this limitation, OPM introduces an overlapped operation on the feature maps. By incorporating overlapping regions between patches, OPM maintains the continuity between patches, thereby improving accuracy of the OBFormer.

The feature maps extracted by the encoder are fed into the UDFF module. Three deep-level feature maps are stacked after channel and size transformation. The stacked feature maps are then passed through an MLP layer to adjust channels and size. The UDFF module strengthens the connection between deep-level and shallow-level feature maps, enhancing feature fusion and contextual information exchange.

The feature maps from the UDFF module are fed to the CRI module for decoding. Feature maps of each scale are fed into classification, regression, and interaction branches, followed by convolution and Sigmoid activation. The outputs from the three decoupled heads are stacked and flattened. Finally, the results from the three decoupled heads are stacked again to generate the final detection output.

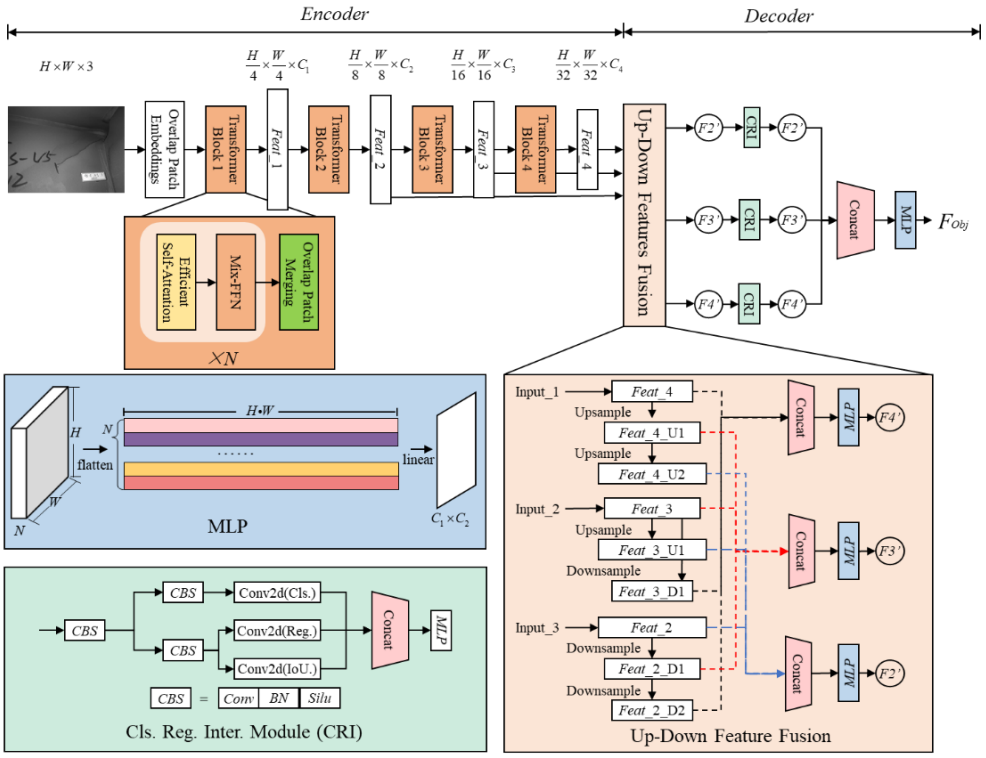


Fig. 1 Structure of OBFormer

2.2. Construction of the dataset

During long-term monitoring and crack detection for two steel bridges, 2,350 images depicting fatigue cracks was collected to create annotated dataset. The images presented several challenges, including issues of overexposure and underexposure, which were attributed to variations in collection environmental conditions. To address these challenges, the data augmentation techniques were used to enhance the diversity of the data and improve generalization capabilities of OBFormer. The dataset was divided into three subsets in an 8:1:1 ratio for the purposes of model training, validation, and test.

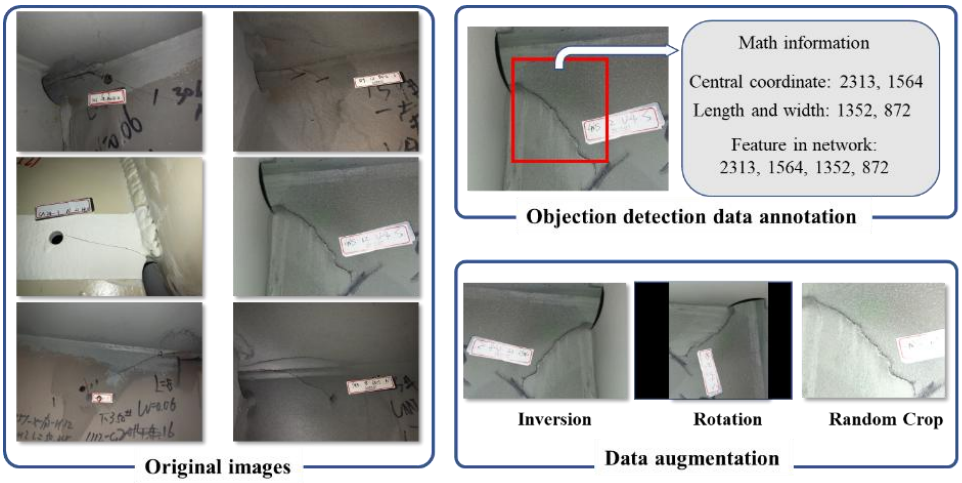


Fig. 2 Dataset construction

2.3. Network training and inference

2.3.1. Training parameters and metrics

In this study, the OBFormer was established and evaluated under the software system of Ubuntu 18.04. The hardware mainly consists of a Ryzen 9-5900X CPU and RTX 3090 GPU with 24GB RAM. The training phase

employed transfer learning strategy to initialize the parameters of the encoder. The initial weights were derived from the training results of the common data set ImageNet. Simultaneously, frozen training strategy was used to optimize training. Learning rate was reduced from 1e-4 to 1e-7 through a cosine annealing decay method[22], in conjunction with the AdamW optimizer[23], over 300 training epochs. The hyperparameters are shown in **Table 1**.

Table 1
Sets of hyperparameter

Hyperparameter	set
Input size	448×448
Batch size	8
Learning rate	1e-4 to 1e-7
Epochs	300
Frozen epochs	20
Optimizer	AdanW

Average precision (*AP*) was utilized to assess the performance of model. *AP* measures the ratio that a sample identified as positive is genuinely a true positive, as shown in Eq. (5) as

$$AP = \frac{1}{N}\sum_{N} TP/(TP+FP) \tag{5}$$

where *TP* refers to true positive samples, which are target pixels that have been correctly classified. *FP* denotes false positive samples, which are background pixels erroneously classified as target pixels.

2.3.2. Training results of OBFormer

Training results are shown in the **Fig. 3**. The loss value curve indicates that both the training and validation value begin to stabilize after the 200th epoch. After the 250th epoch, the training and validation loss values converge to similar levels, indicating good model convergence without signs of underfitting or overfitting. To verify the superiority of OBFormer, two representative state-of-the-art object detection models are selected, YOLOv7 and Mask R-CNN, for comparison. For model training, all hyperparameters are configured as specified in **Table 1**. The quantitative comparison metric used was mAP, and the training results of the three models are illustrated in **Fig. 3** (b). As observed from the curves, all models cease to fluctuate after the 50th epoch, with mAP values steadily increasing thereafter. After the 250th epoch, the values reach a stable region, where OBFormer achieves a final mAP of 0.91, significantly outperforming the other two models. **Fig. 3** (c) shows the detection outcomes for various targets within crack images were notably satisfactory. Specifically, the *AP* values for cracks, stop-holes, and labels were 81%, 95%, and 98%, yielding an overall mAP of 91% across the three categories. **Table 2** presents a comparison of OBFormer, YOLOv7, and Mask R-CNN in terms of model size and inference efficiency on the custom dataset. OBFormer has fewer parameters and FLOPs, indicating a more lightweight architecture. Meanwhile, OBFormer achieves an FPS of 126.7 under the same conditions, substantially higher than the other two models, demonstrating its superior real-time processing capability.

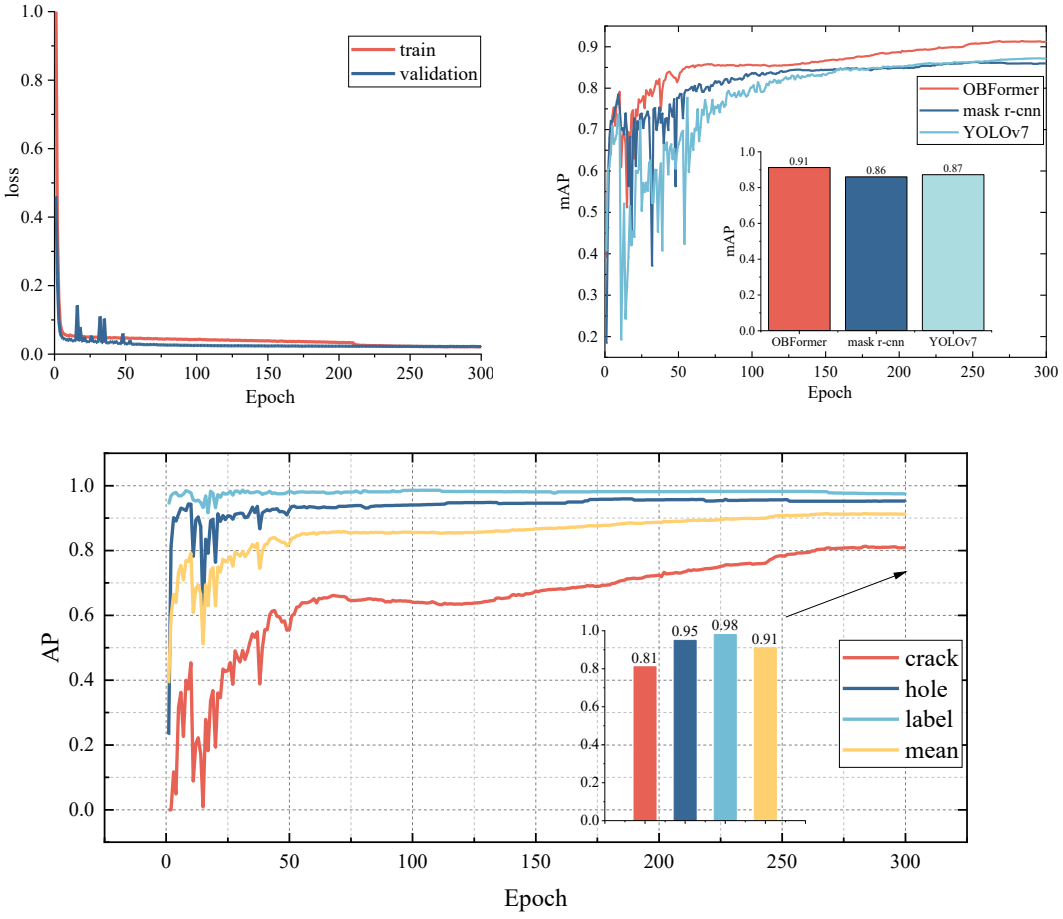


Fig. 3 Training results of OBFormer. (a) Loss curve. (b) Comparison with object detection model. (c) Average precision calculated for all categories

Table 2
Model size

Model	Parameter(M)	Flops(G)	FPS
YOLOv7	37	60	78.3
Mask R-CNN	44	100	35.6
OBFormer	**4.53**	**10**	**126.7**

The deployed results of the model applied to the test dataset are shown in **Fig. 4**, which shows various typical fatigue cracks, stop-hole and label areas. In instances of single crack, it was practical to reliably estimate the area of the cracks, which serves to reduce the quantity of background pixels. In images containing both stop-holes and multiple cracks, the proposed methodology effectively distinguished and detected the areas associated with each individual crack and stop-holes. This approach avoids the generation of a single prediction

box for multiple cracks. The deployed results indicate that the OBFormer exhibits a strong suitability with the test dataset, demonstrating adequate generalization capabilities across all data. The OBFormer is capable of accurately predicting the presence of cracks, stop-holes, and labels information.

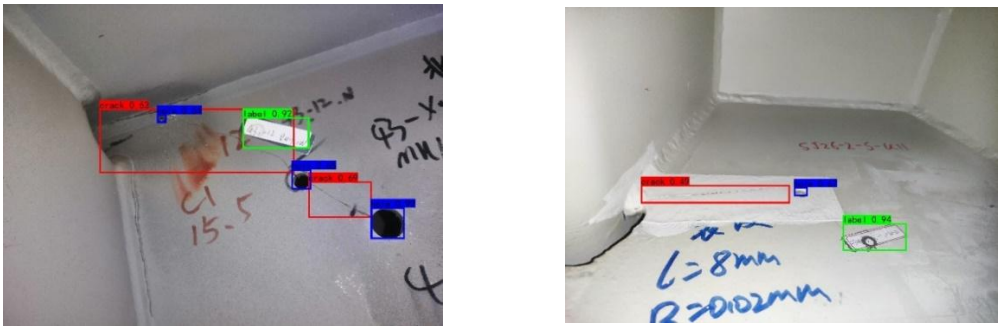


Fig. 4 Results of the detection of representative samples

3. Crack feature extraction

3.1. Optimized edge detection algorithm

The edge detection algorithm presented is derived from the Canny detection algorithm[24], with the Gaussian filtering substituted by bilateral filtering. The bilateral filter incorporates both spatial and range information of pixel points. There is a clear difference in grayscale values between crack and background in image. The diversity enhances the range weight and reduces the spatial weight when convolution kernel is applied to the crack region. The adjustment reduces the smoothing effect on the image, thereby preserving the edge information. In non-crack regions, the filter reduces range weight and increases spatial weight, allowing for smooth processing of the background areas.

Dense bright spots caused by the flash lights result in significant gradients which still exist after smoothing with a large convolution kernel. Based on the characteristics of images collected from steel box girders, where the background is typically dark with low grayscale values and the fatigue crack regions generally have even lower grayscale values, a threshold segmentation preprocessing step is introduced before smoothing the images. All bright spots in the image with grayscale values above the threshold are set to the threshold value by applying an adaptive threshold segmentation method. The mathematical formulation of the algorithm is delineated in Eq. (6)

$$I_{(y)} = \frac{1}{W_q}\sum_{x\in S} I(x)G_S(x,y)G_r(I(x),I(y)) \tag{6}$$

$$G_S(x,y) = \exp(-\frac{\|x-y\|^2}{2\sigma_s^2}) \tag{7}$$

$$G_r(I(x),I(y)) = \exp(-\frac{\|I(x)-I(y)\|^2}{2\sigma_r^2}) \tag{8}$$

$$W_p = \sum_{x\in s} G_d \bullet G_r \tag{9}$$

where S represents the set of pixel points within the neighborhood during the filtering procedure. The x and y denote the coordinates of the target pixel point within neighborhood. $I(x)$ indicates the pixel value of each pixel point in the set S, while $I(y)$ represents the pixel value following the application of filtering processes. The G_S represents the geometric space that characterizes pixel values in bilateral filtering. G_r represents the absolute value of the grayscale variation between a designated point and the central point within the neighborhood. Furthermore, W_p represents the sum of the weights of each pixel value within the filtering window, used for weight normalization, expressed as Eqs.(7)-(9). σ_d and σ_r represent the variances of the image and the grayscale space after filtering.

3.2. Results of feature extraction

The deployed results of OBFormer on given images generated parameters related to the predicted bounding box for the target object, which comprises four values: (b_x, by, b_w, b_h). The b_x and by denote the coordinates of the center of the bounding box, while b_w and b_h indicate width and height. The pixel position information for the four corners of the bounding box in the original image, represented as [a, b, c, d], can be obtained through Eq. (10).

$$a = b_x - b_h/2,\ b = b_x + b_h/2, c = b_y - b_W/2, d = b_y + b_W/2 \tag{10}$$

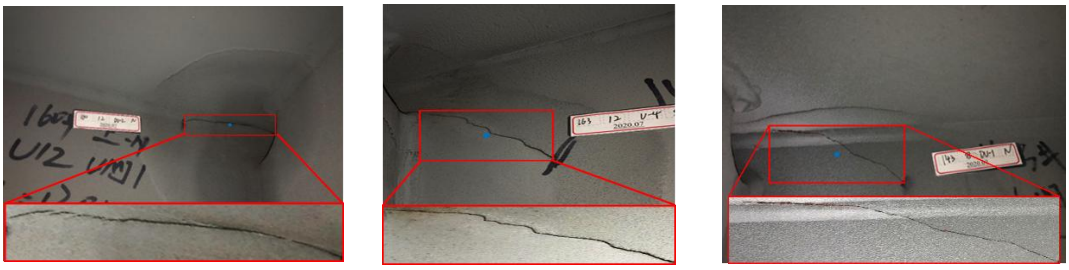


Fig. 5 OBFormer detection results

While OBFormer accurately identifies object-level bounding boxes, precise characterization of structural defects such as cracks requires additional pixel-level edge information. Therefore, an optimized edge detection algorithm was utilized to analyze the images presented in section 3.1, with the results depicted in **Fig. 6**. The figure illustrates the efficacy of the algorithm in identifying various object edge information within the images, such as cracks, labels, handwritten text, and the contours of over-welded holes. However, noise was present in each image, primarily due to inadequate lighting conditions inside the OSDs. The usage of flashlight during photography, in conjunction with the rough texture of the steel surface, intensified the noise present in the images.

To quantify the denoising performance of the two algorithms, structural similarity (SSIM), peak signal-to-noise ratio (PSNR), and Error Rate were used. **Fig. 7** illustrates the results of three representative images processed by two different denoising methods: Gaussian filtering and bilateral filtering. It can be observed that bilateral filtering consistently achieves higher SSIM and PSNR values across all three images, indicating its superior ability to preserve edge features and overall structural information compared to Gaussian filtering. Moreover, the Error Rate is generally lower under bilateral filtering, further confirming its effectiveness in removing noise while minimizing the loss of image details.

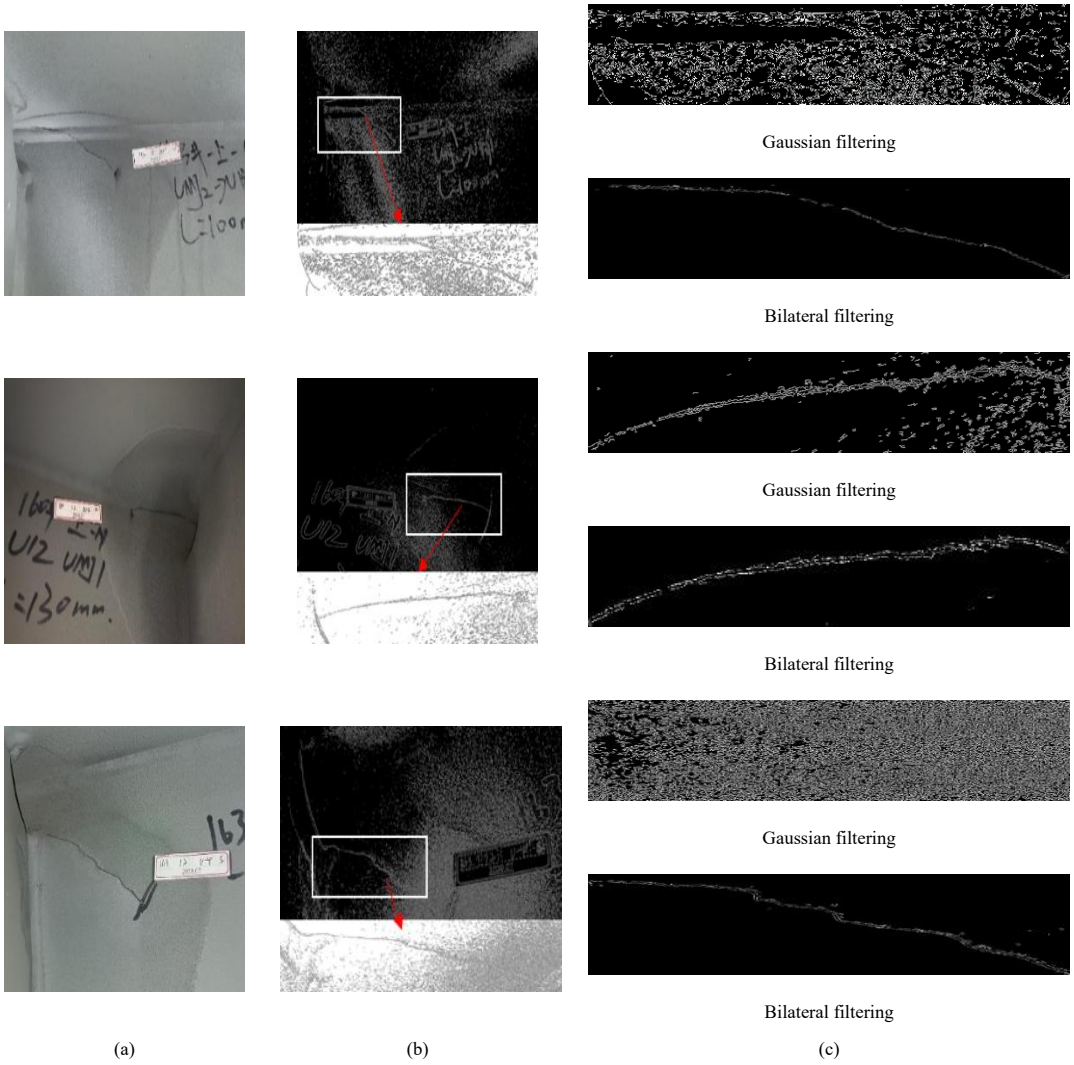


Fig. 6 Results of edge detection. (a) Original image, (b) Whole image, (c) Cropped image

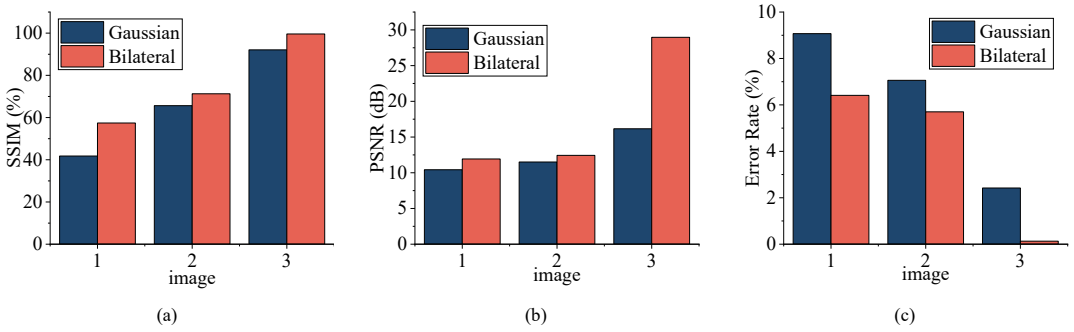


Fig. 7 Comparison of denoising performance between gaussian and bilateral filtering

Bridge maintenance necessitates the collection of data regarding the locations and geometric characteristics of cracks. In the analyzed images, crack regions were manually cropped and repositioned, and color inversion was applied to improve the visibility. Nonetheless, the images were considerably impacted by noise, which obscured the visibility of the crack areas. The presence of text and noise partially disrupted the continuity of the crack paths, complicating the quantification of relevant features. This challenge was exacerbated by the fact that pixels representing the cracks made up only a small proportion of the total pixel count in the images. As a result, the detection process struggled to identify crack areas and corresponding shapes. Therefore, accurately delineating and extracting crack areas from the images was essential for obtaining precise geometric features.

Given that cracks often occupy a small portion of the image and are susceptible to interference from noise and annotation artifacts, crack regions were manually cropped and centered to reduce the influence of surrounding distractors. This preprocessing step also facilitates more effective denoising. The cropped image was centered on the crack region, reducing the influence of weld-joints, labels, and handwritten notes. However, the image that had undergone Gaussian filtering still contained a significant level of noise, which hindered the accurate delineation of the crack area. The outcomes through the application of bilateral filtering demonstrated that the majority of the noise had been eliminated, resulting in an image that primarily comprises pixel information relevant to the cracks. The edges of the cracks were clearly defined, and the geometric features were emphasized, thereby enhancing the ability to recognize the crack shape.

3.3. Morphological processing

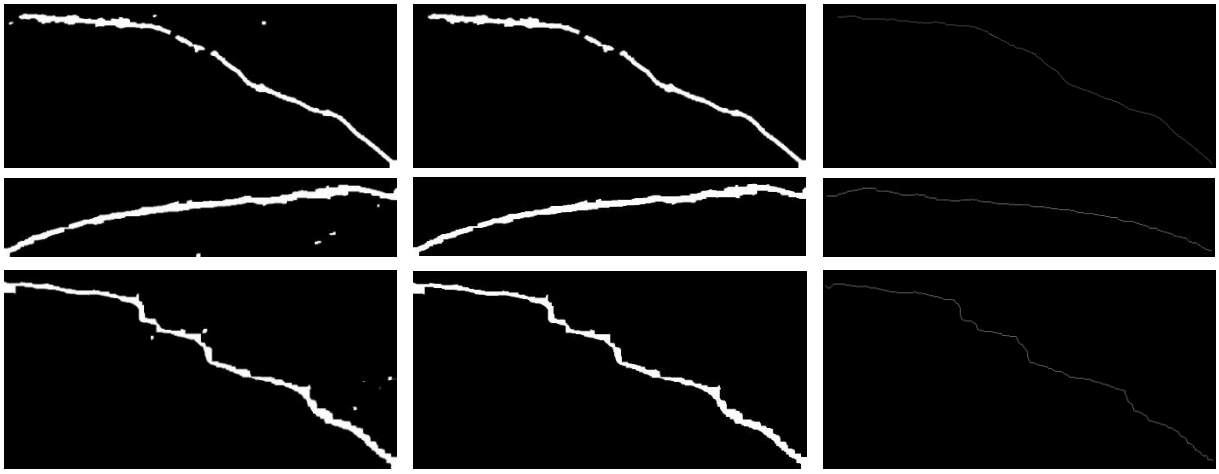
The edge detection results may contain noise or boundary discontinuities making it difficult to directly extract complete crack regions. Therefore, a seed-filling method based on region growing were used to construct connected domains of cracks. The additions enhanced the coherence of the method and

improved the interpretability of the crack extraction process. Crack edges were first detected through feature extraction and then integrated into a connected domain to represent the entire crack area. In this study, a seed-filling method based on the region growing algorithm was employed to establish the connected domain of the crack. Subsequently, a morphological skeletonization technique was implemented, involving an iterative procedure that utilized erosion and opening operations. The iterative process diminished the connected domain, resulting in linear representations while maintaining the topological integrity of the crack shape. The mathematical model is delineated as Eq. (11).

$$S_k(A) = \bigcup_{k=0}^{n} (A \ominus kB) - (A \ominus kB) \bigcirc (B) \tag{11}$$

where A is the original binary image. B is the convolution kernel used for operations. The symbol $\ominus$ is used to indicate the morphological erosion operation. The symbol $\bigcirc$ represents the morphological opening operation, which is defined as the sequence of erosion followed by dilation. The variable k indicates the number of iterations applied during the erosion process. Furthermore, n represents the maximum value at which the resultant set $A \ominus B$ becomes an empty set.

The results obtained from the application of the detection and connected domain algorithm to the fatigue crack images are shown in **Fig. 8**. The images displayed a certain level of point-like noise and discontinuities. To acquire the total crack area, morphological processing was applied to the edge features. A morphological dilation operation was performed on the image, which aided in connecting crack edges in regions. An erosion operation was then used to the dilated image, using a convolution kernel size identical to that of the dilation, to remove noise. The Zhang-Suen algorithm was used to skeletonize the crack, refining the area into lines composed of single-pixel points. The missing segments within the crack curves typically do not exceed 20 pixels, resulting in minimal impact on the overall morphology of the cracks, although they can affect the total pixel count to some extent. To address this, a curve fitting approach was employed to complete the skeleton lines. Compared to the original discontinuities, this method significantly reduces errors in crack pixel quantification.



(a) (b) (c)

Fig. 8 Results of the morphological processing (a)Enhancement of Canny edge detection outcomes, (b) Results of morphological processing, (c) Crack skeletal lines

4. Feature quantification method

4.1. Monocular theory

In the manual gathering of crack images, a single camera was used for imaging capture. However, achieving a shooting angle that was perpendicular to the crack plane posed significant challenges, thereby complicating the accurate depiction of crack geometric features within the image plane. The monocular graphical representation is shown in **Fig. 9**. This study was conducted indoor using a steel box girder specimen with fatigue crack. Given that the gathering of crack images primarily relies on portable cameras in daily maintenance, this study utilized a smartphone as the shooting device. The experimental platform and crack collection process are shown in **Fig. 10**.

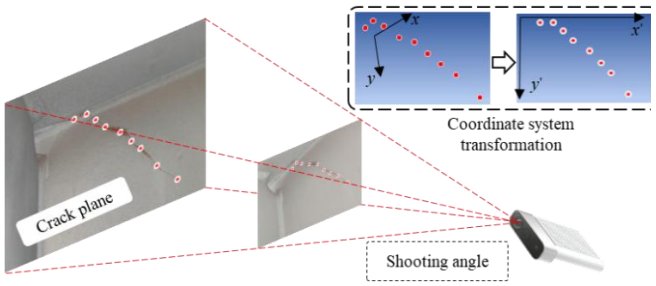


Fig. 9 Method of monocular vision

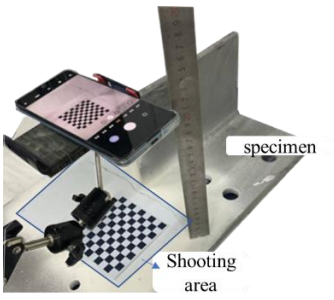


Fig. 10 Experimental platform

The challenge of achieving a shooting angle perpendicular to the crack plane is a key limitation of the monocular vision system. Furthermore, the imaging process may introduce various distortions attributable to factors such as the lens manufacturing processes and the precision of installation. The distortion and intrinsic parameters of the camera are inherent attributes that can be quantified through a procedure known as camera calibration. The mathematical model that describes these parameters is delineated in the following equations:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{12}$$

$$M = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{13}$$

$$x' = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + \left[2p_1 xy + p_2 \left(r^2 + 2x^2 \right) \right] \quad (14)$$

$$y' = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + \left[p_1 \left(r^2 + 2y^2 \right) + 2p_2 xy \right] \quad (15)$$

where H signifies the homography transformation matrix, which articulates the correspondence between points in the pixel coordinate system and those in the world coordinate system. M refers to the specific formulation of the camera intrinsic parameter matrix. The variables x and y denote the coordinates prior to correction, whereas x' and y' indicate the coordinates following correction. The k_1, k_2, k_3, p_1, p_2, f_x, f_y, u_0, and v_0 are variables that necessitate estimation. The proposed solution methodology involves initially calculating the intrinsic and extrinsic parameters of the camera through the homography matrix, subsequently employing nonlinear least squares to estimate the distortion parameters. The derived parameters are calculated using the maximum likelihood estimation approach.

4.2. Experiment of feature quantification

Position the chessboard calibration plate, measuring 10 cm by 10 cm, on the surface of the steel box beam specimen. Adjust the camera orientation and capture a series of images of the chessboard calibration plate from various angles for camera calibration. The parameters of the camera are shown in **Table 3**.

Table 3
the result of parameters calculation

Distortion parameters		Camera parameters	
k_1	0.09093	f_x	775.75
k_2	0.09174	f_y	770.83
k_3	-0.52407	u_0	488.24
p_1	-0.01217	v_0	329.91
p_2	0.00228		

This study employs a standard camera projection model, utilizing the intrinsic camera matrix and pose parameters to perform forward projection from 3D points to 2D image coordinates for reprojection error calculation. The multi-angle corner detection results are shown in **Fig. 11**, with an average reprojection error computed as 0.052 pixels.

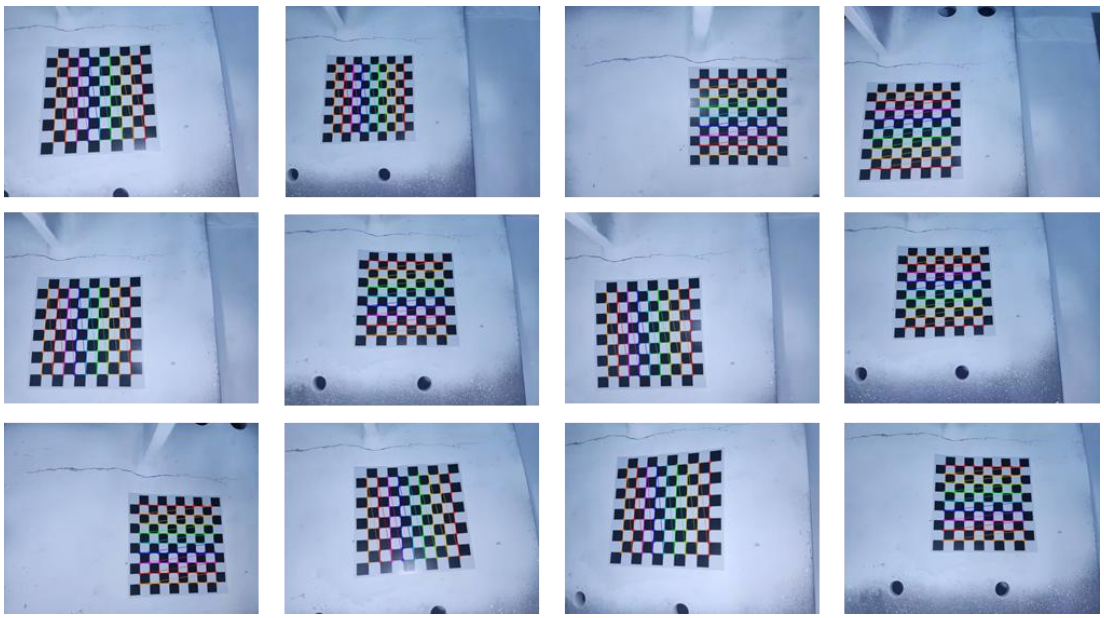
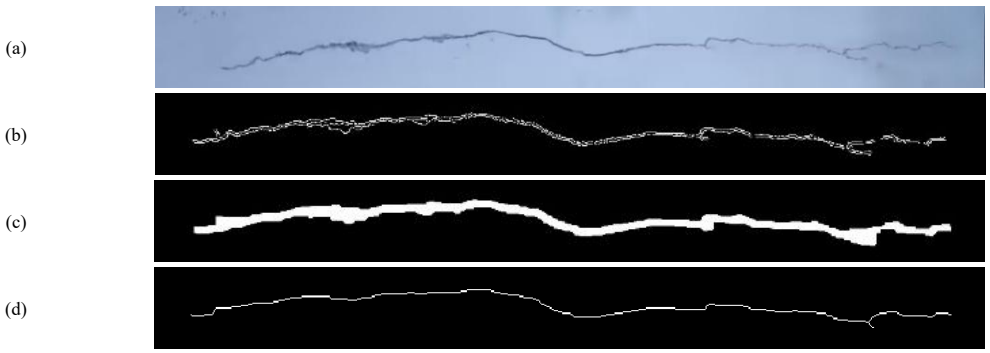


Fig. 11 Chessboard corner detection results

After obtaining the camera intrinsic parameters, feature quantification analysis was conducted on 9 fatigue crack specimen images captured at varying distances and angles. The OBFormer model was employed to detect and crop local images depicting fatigue cracks. The image processing techniques described in Section 3 are applied to the acquired images to identify the crack edges and the associated connected components, following dilation and erosion operations. Subsequently, crack skeletons are extracted from the images of the connected components. The presentative experimental results are shown in **Fig. 12**.

Fig. 12 (e) illustrates the quantified crack length results from nine images captured under varying shooting angles and distances. The angles include ±45º and 90º, and the distances are 20 cm, 30 cm, and 40 cm. In the figure, blue bars represent the crack lengths extracted through image-based recognition, while the red line indicates the variation in relative error. As observed, the relative error ranges from a minimum of 1.8% to a maximum of 4.4%, all within 5%. These results demonstrate that the proposed crack skeleton extraction and breakpoint completion algorithm achieves high stability and accuracy under different imaging conditions.



(e)

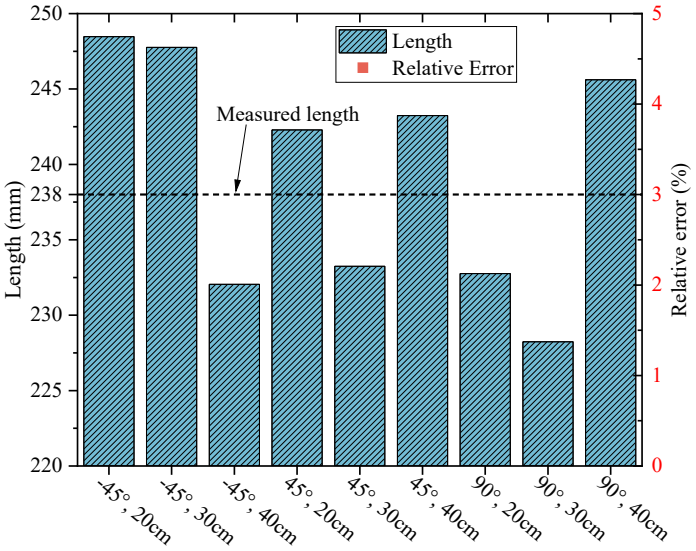


Fig. 12 Results of feature quantification

(a) Cropped results, (b) Optimization algorithm for extracting crack edges,
(c) Crack connected component, (d) Crack skeletal line, (e) Quantification results

5. Conclusion

Recognizing and quantifying of fatigue cracks in steel box girders is highly relevant for daily maintenance, as it impacts the maintenance strategies. The primary objective of this study is to establish a methodology for quantifying crack length through the implementation of area recognition algorithms. Therefore, the study utilizes the OBFormer object detection model alongside daily maintenance image data to calculate the crack length. The efficacy of this approach has been confirmed.

From the previous research, the following conclusions were made:

1. The OBFormer model exhibits high effectiveness in detecting cracks while overcoming image interference on surfaces of OSDs, resulting in mAP of 91.2%.
2. The truncation method for threshold segmentation, combined with an optimized Canny edge detection algorithm, effectively extracts fatigue crack edges under various interference conditions. Additionally, when employed alongside post-processing algorithms, this approach facilitates the accurate extraction of the crack skeletal line.
3. The method for quantifying fatigue crack length in steel box girders, integrated with monocular vision technology, allows for the precise measurement of fatigue crack length, typically achieves a relative error of less than 5%.

Acknowledgment

This work was supported by National Natural Science Foundation of China [grant numbers 52308166, 52378153].

References

[1] Z.Q. Fu, B.H. Ji, D.D. Zhao, M.Y. Yang, Corrosion evaluation of steel material on steel box girder bridge by comprehensive scoring method, Materials Research Innovations 18 (2014) S2-840-S2-844. https://doi.org/10.1179/1432891714Z.000000000465.

[2] M.D. Sangid, The physics of fatigue crack initiation, International Journal of Fatigue 57 (2013) 58–72. https://doi.org/10.1016/j.ijfatigue.2012.10.009.

[3] K. Sun, X. Jiang, X. Qiang, Z. Lv, C. Fan, Equivalent Vehicle Model Based on Traffic Flow of Long-Span Steel Box Girder Bridge, J. Bridge Eng. 28 (2023) 04023061. https://doi.org/10.1061/JBENF2.BEENG-6086.

[4] Y. Wang, L. Chao, J. Chen, S. Jiang, Fatigue Crack Propagation Law of Corroded Steel Box Girders in Long Span Bridges, Computer Modeling in Engineering and Sciences 140 (2024) 201–227. https://doi.org/10.32604/cmes.2024.046129.

[5] X. Jiang, K. Sun, X. Qiang, D. Li, Q. Zhang, XFEM-based Fatigue Crack Propagation Analysis on Key Welded Connections of Orthotropic Steel Bridge Deck, Int J Steel Struct 23 (2023) 404–416. https://doi.org/10.1007/s13296-022-00701-3.

[6] C.-Y. Wang, A. Bochkovskiy, H.-Y.M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, (2022). http://arxiv.org/abs/2207.02696 (accessed July 17, 2024).

[7] S. Teng, Z. Liu, G. Chen, L. Cheng, Concrete Crack Detection Based on Well-Known Feature Extractor Model and the YOLO_v2 Network, Applied Sciences 11 (2021) 813. https://doi.org/10.3390/app11020813.

[8] G. Ye, S. Li, M. Zhou, Y. Mao, J. Qu, T. Shi, Q. Jin, Pavement crack instance segmentation using YOLOv7-WMF with connected feature fusion, Automation in Construction 160 (2024) 105331. https://doi.org/10.1016/j.autcon.2024.105331.

[9] H. Zunair, A. Ben Hamza, Sharp U-Net: Depthwise convolutional network for biomedical image segmentation, Computers in Biology and Medicine 136 (2021) 104699. https://doi.org/10.1016/j.compbiomed.2021.104699.

[10] Y. Chu, X. Xiang, Y. Wang, B. Huang, Pavement Disease Detection through Improved YOLOv5s Neural Network, Computational Intelligence and Neuroscience 2022 (2022) 1–12. https://doi.org/10.1155/2022/1969511.

[11] G. Ye, J. Qu, J. Tao, W. Dai, Y. Mao, Q. Jin, Autonomous surface crack identification of concrete structures based on the YOLOv7 algorithm, Journal of Building Engineering 73 (2023) 106688. https://doi.org/10.1016/j.jobe.2023.106688.

[12] Z. Shi, N. Jin, D. Chen, D. Ai, A comparison study of semantic segmentation networks for crack detection in construction materials, Construction and Building Materials 414 (2024) 134950. https://doi.org/10.1016/j.conbuildmat.2024.134950.

[13] S. Qin, T. Qi, T. Deng, X. Huang, Image segmentation using Vision Transformer for tunnel defect assessment, Computer Aided Civil Eng (2024) mice.13181. https://doi.org/10.1111/mice.13181.

[14] R. Ali, J.H. Chuah, M.S.A. Talip, N. Mokhtar, M.A. Shoaib, Crack Segmentation Network using Additive Attention Gate—CSN-II, Engineering Applications of Artificial Intelligence 114 (2022) 105130. https://doi.org/10.1016/j.engappai.2022.105130.

[15] Y.-C. Chen, R.-T. Wu, A. Puranam, Multi-task deep learning for crack segmentation and quantification in RC structures, Automation in Construction 166 (2024) 105599. https://doi.org/10.1016/j.autcon.2024.105599.

[16] C. Dong, L. Li, J. Yan, Z. Zhang, H. Pan, F.N. Catbas, Pixel-Level Fatigue Crack Segmentation in Large-Scale Images of Steel Structures Using an Encoder–Decoder Network, Sensors 21 (2021) 4135. https://doi.org/10.3390/s21124135.

[17] Y. Xu, Y. Bao, J. Chen, W. Zuo, H. Li, Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images, Structural Health Monitoring 18 (2019) 653–674. https://doi.org/10.1177/1475921718764873.

[18] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J.M. Alvarez, P. Luo, SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers, Advances in Neural Information Processing Systems 34 (2021) 12077–12090.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is All you Need, Advances in Neural Information Processing Systems 30 (2017).

[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, (2020). http://arxiv.org/abs/2010.11929 (accessed November 27, 2023).

[21] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Montreal, QC, Canada, 2021: pp. 9992–10002. https://doi.org/10.1109/ICCV48922.2021.00986.

[22] I. Loshchilov, F. Hutter, SGDR: Stochastic Gradient Descent With Warm Restarts, in: International Conference on Learning Representations, 2017.

[23] I. Loshchilov, F. Hutter, Decoupled Weight Decay Regularization, arXiv Preprint arXiv:1711.05101 (2019). http://arxiv.org/abs/1711.05101 (accessed September 7, 2024).

[24] J. Canny, A Computational Approach to Edge Detection, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 (1986) 679–698. https://doi.org/10.1109/TPAMI.1986.4767851.