

Article

Crack Length Measurement Using Convolutional Neural Networks and Image Processing

Yingtao Yuan ^{1,2}, Zhendong Ge ^{1,2}, Xin Su ^{1,2}, Xiang Guo ^{1,2,*}, Tao Suo ^{1,2}, Yan Liu ³ and Qifeng Yu ^{1,2,3}

- ¹ School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China; yuanyingtao@mail.nwpu.edu.cn (Y.Y.); 13817583533@163.com (Z.G.); suxin.ds@mail.nwpu.edu.cn (X.S.); suotao@nwpu.edu.cn (T.S.); yuqifeng58@139.com (Q.Y.)
² International Research Laboratory of Impact Dynamics and its Engineering Application, Xi'an 710072, China
³ Institute of Intelligent Optical Measurement and Detection, Shenzhen University, Shenzhen 518060, China; yan_liu@mail.nwpu.edu.cn
* Correspondence: guoxiang1984@nwpu.edu.cn; Tel.: +86-135-1912-9863

Abstract: Fatigue failure is a significant problem in the structural safety of engineering structures. Human inspection is the most widely used approach for fatigue failure detection, which is time consuming and subjective. Traditional vision-based methods are insufficient in distinguishing cracks from noises and detecting crack tips. In this paper, a new framework based on convolutional neural networks (CNN) and digital image processing is proposed to monitor crack propagation length. Convolutional neural networks were first applied to robustly detect the location of cracks with the interference of scratch and edges. Then, a crack tip-detection algorithm was established to accurately locate the crack tip and was used to calculate the length of the crack. The effectiveness and precision of the proposed approach were validated through conducting fatigue experiments. The results demonstrated that the proposed approach could robustly identify a fatigue crack surrounded by crack-like noises and locate the crack tip accurately. Furthermore, crack length could be measured with submillimeter accuracy.

Keywords: crack length; image processing; convolutional neural network; fatigue crack detection



Citation: Yuan, Y.; Ge, Z.; Su, X.; Guo, X.; Suo, T.; Liu, Y.; Yu, Q. Crack Length Measurement Using Convolutional Neural Networks and Image Processing. *Sensors* **2021**, *21*, 5894. <https://doi.org/10.3390/s21175894>

Academic Editor: Iren E. Kuznetsova

Received: 3 August 2021

Accepted: 27 August 2021

Published: 1 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Fatigue cracks caused by repetitive loads, which are of great concern for structural safety, always exist in old structures including airplane and highway bridges [1]. Fatigue crack propagation testing is an essential method of studying metallic or structural fatigue life prediction in fracture mechanics. Crack length is one of the most relevant parameters that needs to be recorded during laboratorial tests.

In recent decades, extensive research on monitoring crack length propagation has been carried out. Existing approaches including the human inspection method, the electrical method [2], the compliance method [3], and acoustic emission technology [4] all have limitations for application during a fatigue test. The human inspection method requires large testing times since it needs to interrupt the fatigue test to manually measure crack lengths. The accuracy of the electrical method is easily affected by the experiment conditions and environment. Complicated calibration is needed when applying the compliance method. Acoustic emission technology is a new nondestructive testing method. It records and analyzes the signals released by materials or structures when deformation or damage occurs to detect the damage location and predict the time to failure. However, this technology relies on a good correlation between acoustic emission data and the damage mechanism, while the experimental results will be affected by environmental noise [5]. Therefore, the application of acoustic emission technology for monitoring the length of cracks is limited.

Vision-based crack detection methods have been widely studied over recent decades for their advantages of non-contact, high precision, and good real-time performance [6].

Many methods have been established based on image processing techniques including edge detection [7], Hough transform [8], image segmentation [9], identification and detection of feature points [10,11], the digital image correlation (DIC) method [12,13], and photogrammetry [14]. A general limitation of these approaches, however, is that most of these vision-based methods detect cracks by searching over the entire area of an image. As a result, there are difficulties in distinguishing true cracks from crack-like noises such as scratches and structure boundaries [15]. At the same time, accurate detection of crack tips, which is key for crack length measurement, remains challenging [10].

In recent years, machine learning and convolutional neural networks have shown a strong capability for feature extraction and target detection, and they have been used in structural health monitoring (SHM). For example, Dong [16] established a CNN model to identify microseismic events and blasts accurately. Yu [17] put forward a DCNN-based method to localize damages to smart building structures with high accuracy on raw noisy signals. Many researchers have carried out studies for crack detection based on CNN. Cha [18] used trained CNN and sliding window techniques to detect cracks from images. Other researchers [19,20] used trained CNN to divide every crack pixel from the background in the image. Although these approaches can detect the cracks with high reliability even surrounded by crack-like noises, it would be difficult to accomplish quantitative detection of crack length.

Previously, the authors investigated a crack propagation detection method [21] using classification-based CNN and an improved canny edge-detection algorithm, and verified the accuracy and efficiency of the method via central hole specimens. However, the classification-based CNN and the edge-detection algorithm only used part of the crack structure information, which reduced the reliability of the method.

In this study, we proposed an improved crack length measurement framework based on detection-based CNN and digital image processing. This framework combined the excellent feature extraction capabilities of detection-based CNN with the quantitative detection capabilities of traditional vision-based methods. This CNN-based model was established to robustly detect whole true cracks surrounded by crack-like noises. A new image-processing algorithm was proposed to accurately locate the crack tip, which could be used to calculate the length of the crack.

The manuscript is organized as follows. Section 2 presents the synopsis of the proposed algorithm first, then the architecture of CNN used in this paper and the methodology of crack tip detection are introduced and thoroughly detailed. Section 3 illustrates the experimental process and results including training dataset preparation, CNN training results, crack area location results, crack tip detection results, and crack length measurement results. Section 4 concludes the paper.

2. Methodology

2.1. Overview of the Proposed Method

The crack length measurement method proposed in this paper is shown in Figure 1, with more technical details described in the rest of this section. To begin, a fixed camera was applied to acquire images while a specimen was subject to a fatigue crack under a fatigue load F , as Figure 1a shows. These crack images contained not only crack information but also crack-like noises. At the same time, the experimental information such as fatigue cycles could be recorded. Then, to achieve robust crack detection, a convolutional neural network trained by labeled crack images was used to detect the region of the crack with the interference of a non-crack edge, as shown in Figure 1b. The architecture of the applied CNN is introduced in Section 2.2, and the process of training dataset preparation and the crack area detection results are detailed in Sections 3.1 and 3.2. Next, a crack tip location method based on digital image processing was used to detect the crack tip from the extracted crack area, as shown in Figure 1c. The principle of the proposed method is illustrated in Section 2.3, and the experimental results are presented in Section 3.3. In the

end, the crack length was calculated with the calibration result, as shown in Figure 1d. The introduction of a validation experiment and the results analysis are described in Section 3.4.

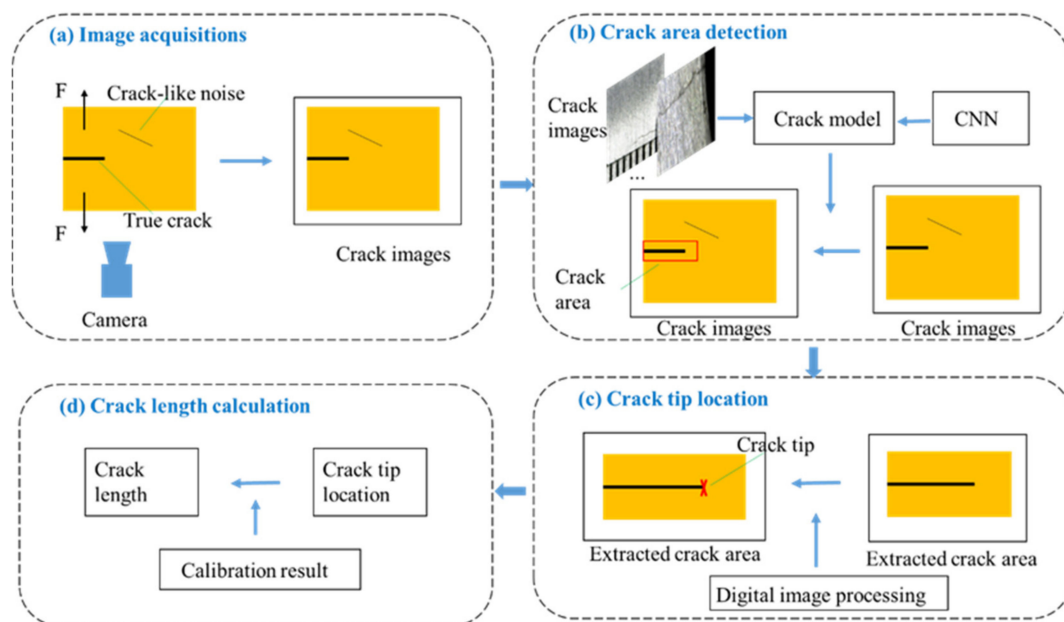


Figure 1. The scheme of the crack length measurement method: (a) image acquisitions; (b) crack area detection; (c) crack tip location; (d) crack length calculation.

2.2. Crack Area Location Method

This section explains the overall architecture of CNN used in this study for crack detection. CNN is a robust method for detecting different objects and has been widely used in automatic drive and intelligent housing systems. CNN extracted the features of objects by convolution layers which could be recognized as the filters. Then, feature maps that indicated the locations and strength of the input object were calculated. In the end, the networks learned the best parameters by adjusting themselves to reduce the error on the dataset. Thus, CNN could be used to distinguish cracks from non-crack noises. Figure 2 shows the details of the CNN architecture. The input crack image was first resized into $300 \times 300 \times 3$, where each dimension indicated height, width, and channel, respectively. Then, the crack image passed through the architecture with 6 feature maps extracted, including conv11, conv13_2, conv14_2, conv15_2, conv16_2, and conv17_2. The term “conv” refers to the convolution layer. These feature maps were used to train and detect the location of cracks. The meta-architecture of the CNN was a Single Shot Detector (SSD) [22], and the feature extractor was MobileNets [23].

A convolutional layer is one of the most important parts in CNN. As shown in Figure 3, the size of the input data was 5×5 , the size of the convolutional kernel was 3×3 , and the size of the output was 3×3 . The convolutional layer performed element-by-element multiplications. The multiplied values were summed and, adding bias, we derived the output value. The stride was a hyperparameter, describing how many of the kernel’s columns and rows slid at a time across the input data’s width and height. One of the advantages of convolution was that it could reduce input data size, and as a result the computational cost was reduced. The feature maps mentioned above are the results of data input after the convolution operation. However, the operation of convolution was time consuming, so we replaced it with a more efficient network.

MobileNets can effectively reduce the number of parameters in CNN architecture. The principle of MobileNets is shown in Figure 4. The standard convolutional layer was replaced by depthwise separable convolution, which factorized the standard convolution into two parts, a depthwise convolution and a 1×1 convolution (also called a pointwise

convolution). If an $N \times N \times P$ feature map was taken as the input by a standard convolution, and a $D \times D \times Q$ feature map was produced, then the computational cost was $N \times N \times P \times D \times D \times Q$ for each layer. When depthwise separable convolution was applied, the computational cost of a depthwise convolution was $P \times N \times N \times D \times D$, and the computational cost of a pointwise convolution was $P \times Q \times N \times N \times 1 \times 1$. The number of parameters reduced from $P \times Q \times N \times N \times D \times D$ to $P \times N \times N \times D \times D + P \times Q \times N \times N \times 1 \times 1$. As a result, the time cost for detecting cracks would be lower.

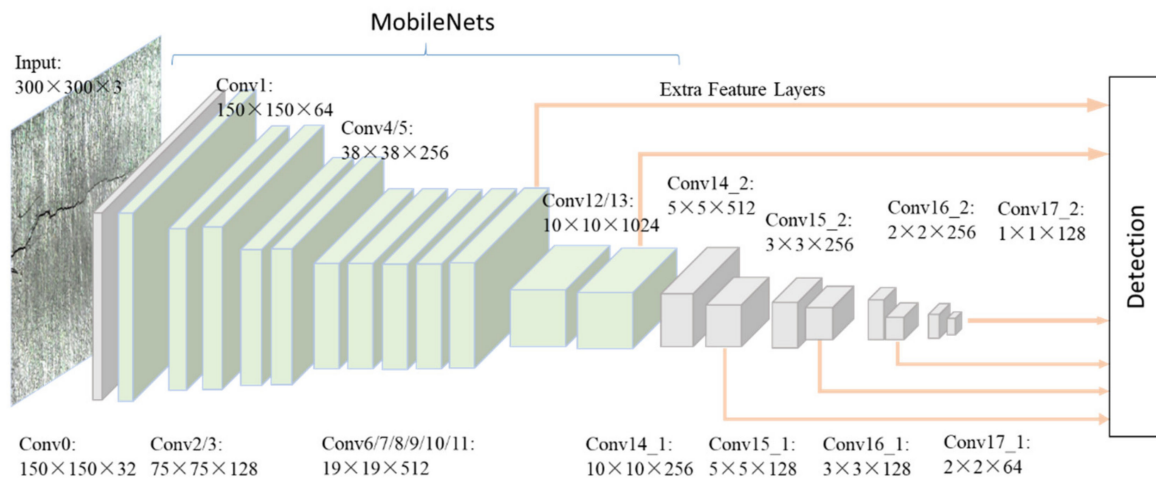


Figure 2. The architecture of the CNN for crack area location.

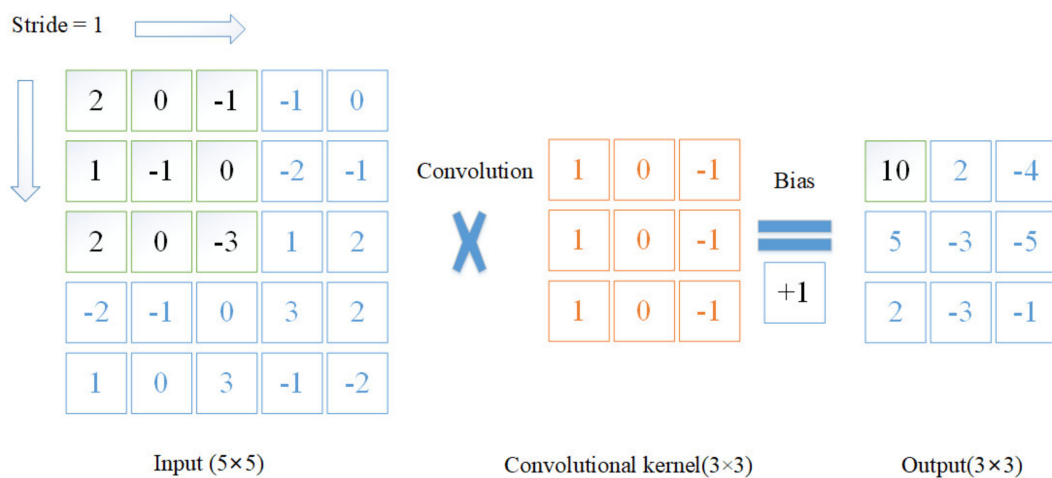


Figure 3. The principle of convolutional layers.

In this study, we used the Rectified Linear Unit (ReLU) [24] as the activation function for all layers. Briefly, the ReLU had no bounded outputs for its positive input values and the gradients are either zeros or ones. This enabled ReLU to compute faster and could achieve better accuracies. The function of ReLU is shown as:

$$y = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (1)$$

The meta-architecture used in this study was SSD. This architecture could detect objects faster without losing accuracy [25], and provided the possibility to detect cracks in real time. It used different resolutions of feature maps to predict different length cracks. As shown in Figure 5, multi-scale feature maps were extracted using a set of convolutional layers. These maps decreased in size and made predictions of cracks at multiple scales. Large-scale feature maps could be used to predict small cracks and small-scale feature

maps could be used to predict large cracks. Another method it used to detect different scale cracks was setting different aspect ratios for default bounding boxes. As shown in Figure 6, the crack was distributed in a linear fashion and a square area was not suitable for the prediction of cracks. Therefore, we set five different values including 1, 2, 3, 1/2, and 1/3 as aspect ratios for default boxes. Each feature map cell predicted the offsets relative to these default bounding boxes.

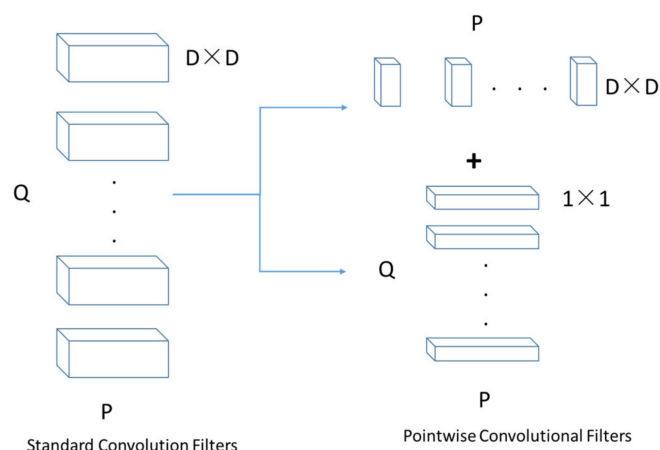


Figure 4. The principle of MobileNets.

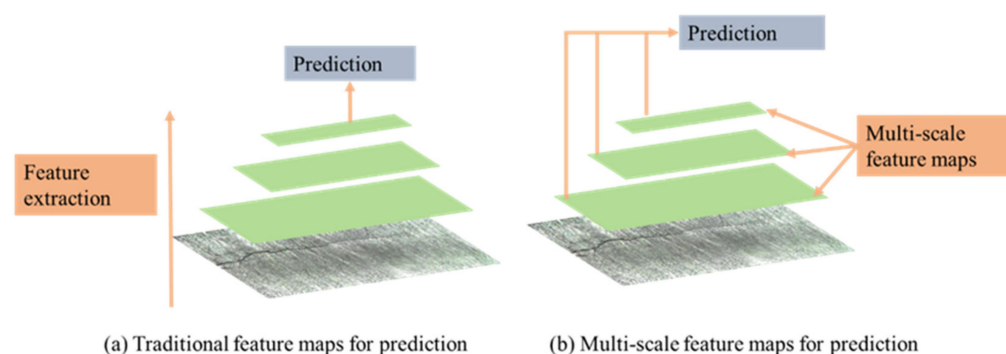


Figure 5. Multi-scale crack feature maps for detection.

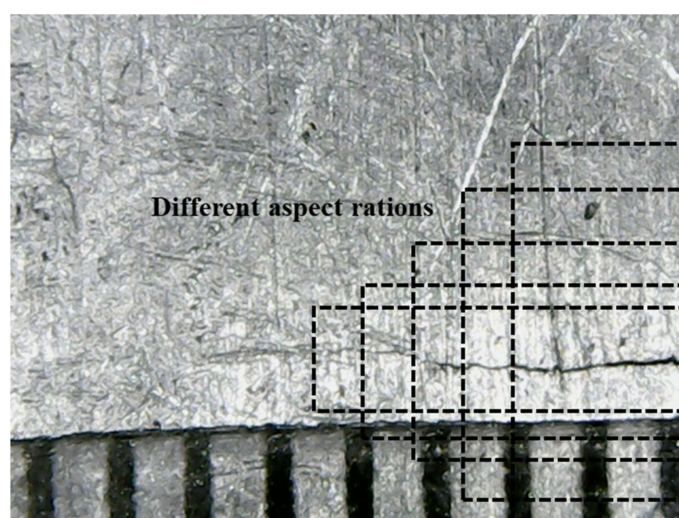


Figure 6. Different aspect ratios for crack detection.

The overall loss function was a weighted sum of the localization loss and the confidence loss:

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + L_{loc}(x, l, g)) \quad (2)$$

where N is the number of matched default boxes; L_{conf} is the confidence loss; L_{loc} is the localization loss; $x = \{1, 0\}$ is an indicator for matching the i -th default box (l) to the j -th ground-truth box (g) parameters of the crack; c means the values of confidence.

The localization loss is a Smooth L1 loss [26,27]:

$$L_{loc}(x, l, g) = \sum_{i \in Pos} \sum_{m \in \{cx, cy, w, h\}} x_{ij} \cdot smooth_{L1}(l_i - \hat{g}_j^m) \quad (3)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx}) / d^w; \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy}) / d^h$$

$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right); \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right)$$

where (cx, cy) means the center of the default bounding box (d), and w and h are the width and height of the default bounding box.

The confidence loss refers to the reliability of predictions that the box contains a crack:

$$L_{conf}(x, c) = - \sum_{i \in Pos} x_{ij} \cdot \log(\hat{c}_i^1) - \sum_{i \in Neg} \log(\hat{c}_i^0) \quad (4)$$

2.3. Crack Tip Detection Method

Once the crack area was detected using CNN, the next step was to robustly locate the crack tip. As shown in Figure 7, the algorithm proposed to locate the crack tip could be divided into four steps. First, an image threshold segmentation method was applied to divide the crack and the background. Then, we used an image morphological operation method to connect the broken crack. Next, a noise-free crack image was obtained after removing small-area noises. In the end, the crack tip could be located by searching the image.

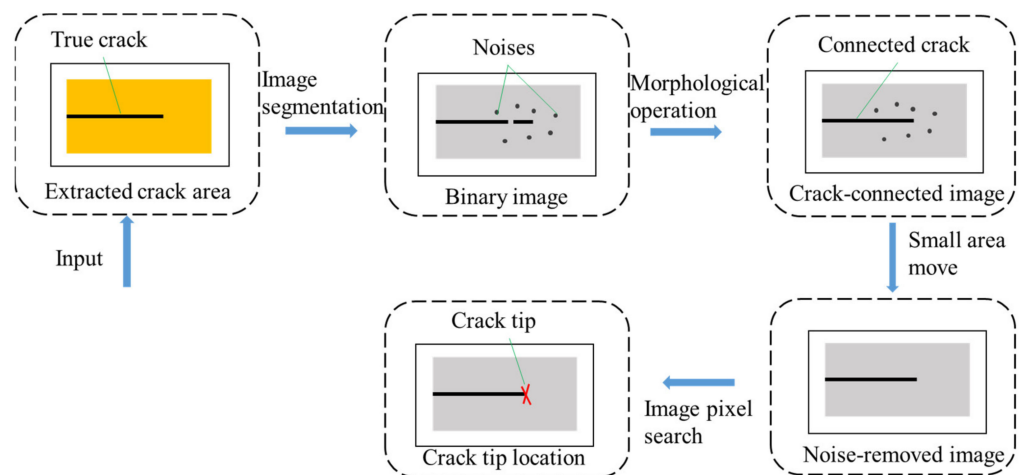


Figure 7. The method proposed to detect crack tips.

In this study, we used a global threshold segmentation method to divide the crack and background to keep the information about the crack tip. During the process of crack propagation, the brightness in the location of the crack was darker than the background. Therefore, we set a grayscale value T to separate the crack and the background. Every

crack image pixel was divided into two groups after threshold segmentation. The function of threshold segmentation used in the study was:

$$g(x,y) = \begin{cases} 0 & f(x,y) > T \\ 255 & f(x,y) \leq T \end{cases} \quad (5)$$

As the crack tip was small and would be easily disturbed by noises, the area of cracks, especially the crack tip, was disconnected after the operation of threshold segmentation. We needed to enhance the area of the crack before removing noises.

Morphological operators [28] included dilation and erosion. The principle of dilation is shown in Figure 8. The dilation of region A by a structuring element B produced a new region, adding a layer of pixels to the boundary of regions. Dilation could fill in the holes and gaps between different regions. The principle of erosion is shown in Figure 9. The erosion of area A by a structuring element B produced a new region, which was smaller than A. Erosion could remove small-scale details, which meant it could be used to remove noises. The closing operation dilated an image and then eroded the dilated image. This was useful for filling small holes and gaps while preserving the shape and size of the objects. Thus, we could use the closing operation to connect the crack and keep the shape of the crack. According to the shape of crack, we could use a rectangular structuring element to close the crack image.

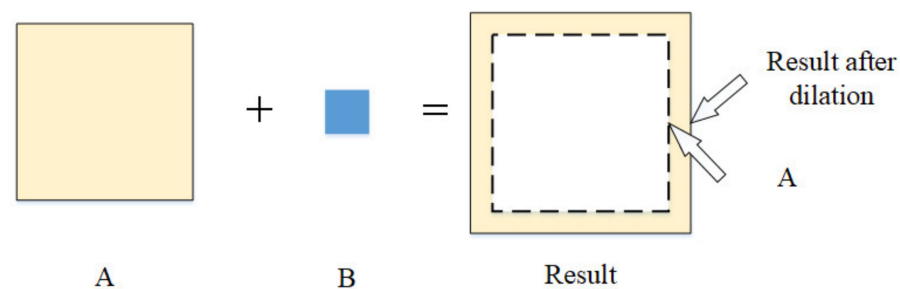


Figure 8. The principle of dilation.

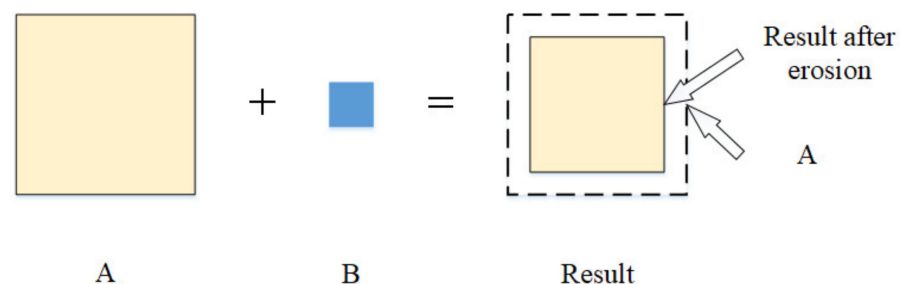


Figure 9. The principle of erosion.

After the crack connection, we could label the regions by searching the binary image and recording the number of pixels. We found that the noises had a small area. Therefore, we could remove the noises by removing small-area objects. As a result, a clear crack binary image was obtained. In the end, we could search the pixel location of 255 and define the rightmost pixel as the crack tip.

3. Experimental Results and Analysis

3.1. Training Data Preparation and Parameter Initialization

CNN is a data-driven approach. Before training the CNN model, the authors carried out fatigue experiments to collect crack images. As shown in Figure 10, a measuring magnifier was used to capture the crack images. The total number of raw images was 4928 (all images with 640×480 pixel resolutions). Some examples of crack images used for

training CNN are shown in Figure 11. There were three kinds of crack images including images with large cracks, images with small cracks, and crack images in larger resolution.



Figure 10. Experiments for collecting crack data: the left image shows the measuring magnifier used in the experiments, and the right image shows the experimental environment.

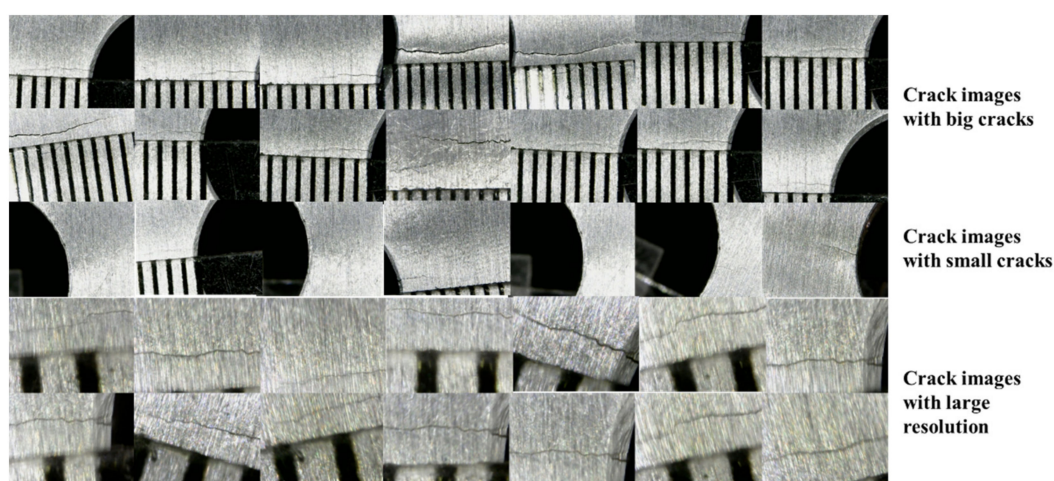


Figure 11. Examples of images used for training CNN.

These crack images were divided into two groups according to a fivefold cross-validation principle of 80% for training and 20% for validation. All the crack images were divided into two parts including 3928 images as training datasets and 1000 images as validation sets.

These crack images were manually labeled with LabelImg [29]. The information of the crack including the location and category in each image was recorded. In this study, the CNN model was built in the framework of TensorFlow. Before training the model, the dataset was converted into TFRecords Format, which is suitable for TensorFlow framework. At that point, we established a crack dataset.

A crack was a new category for us to train and detect. The transfer learning method was not suitable for our CNN model, since research demonstrated that the performance relies on the similarity of two tasks [30,31]. We initialized the model parameters with a normal distribution with mean of 0 and standard deviation of 0.03, and the biases were initialized with a constant zero vector. According to Huang [25], the initial learning rate was set as 0.004. The batch size was set as 1 due to the limitation of our computer.

3.2. CNN Training Results

The training process was carried out on a workstation with a GPU (GeForce GTX 1080Ti). The code was programmed by Python 3.6, and the environment was established by TensorFlow1.12. The training process was performed in Windows systems.

The learning rate was manually adjusted over the training iterations, so that the parameters could converge to the global optimum. In general, larger learning rates were preferred in the beginning and smaller learning rates were used for fine-tuning. The total loss with the training process over steps is shown in Figure 12. In the first iteration, we set the learning rate equal to 0.004, and trained 55.6K steps. The value of total loss decreased quickly at the beginning and slowly stabilized at about 6.5 in the end. Then, we set the learning rate equal to 0.0004 to train 95K steps. The value of total loss stabilized at about 6.13 in the end. Finally, we used the learning rate of 0.00004 to train 60K steps. The value of total loss stabilized at about 6.1 in the end.

Figure 13 shows the results over the training process. The Intersection of Union (IOU) is defined as the area of intersection divided by the area of union between the predicted bounding box and the ground-truth bounding box. It has been widely used to measure the accuracy of object detection tasks. A value of 0.5 or 0.75 was the threshold we used to judge whether the location of the predicted crack was correct or not. From the results, we could see that the value of mean average precision (mAP) increased quickly at the beginning and slowly stabilized in the end.

Table 1 summarizes the final results. In the training process, the mAP increased from 0.71 to 0.86 at 0.5IOU and increased from 0.29 to 0.35 at 0.75IOU. It could also validate that the parameters converged to the global optimum by changing the learning rate.

Table 1. Crack detection precision.

Result	First Step	Second Step	Third Step
mAP (0.5IOU)	0.71	0.84	0.86
mAP (0.75IOU)	0.29	0.34	0.35

After the training, the trained CNN model was applied to other crack images for further validation. As shown in Figure 14, the trained CNN model could detect cracks with high precision not only for large cracks but also for small cracks. It could also detect cracks in different resolution images. At the same time, the method could divide the crack from crack-like noises effectively.

3.3. Crack Tip Location Results

This section describes the results of the crack tip detection method. Figure 15a shows the results after locating the area of cracks using the trained CNN model. The region of the crack was extracted as a new image, which is shown in Figure 15b. Figure 15c shows the result of a binary image after image segmentation. In this study, the grayscale value T was 83. If the pixel gray value was higher than 83, it would be redefined as 0. Otherwise, if the pixel gray value was lower than 83, the number would be redefined as 255.

As we can see in Figure 16b, the crack tip was disturbed after global segmentation. The crack tip information would be easily missed in the next operation. Before removing the noises in the picture, we connected the crack tip first. Figure 16c shows the results of the crack region after morphological operation. We designed a rectangular structuring element with length of 15 and width of 3, using the element with the closing operation to connect the crack. The crack tip image before connection and after connection can be seen in Figure 16b,d. The different regions of the crack were perfectly connected after the closing operation.

Figure 16e,f show the results of the crack image after moving the noises area. As we can see, the noises could be moved clearly while the number of area thresholds was set

as 200. At this point, we could recognize the crack tip as the rightmost point where the greyscale value was 255.

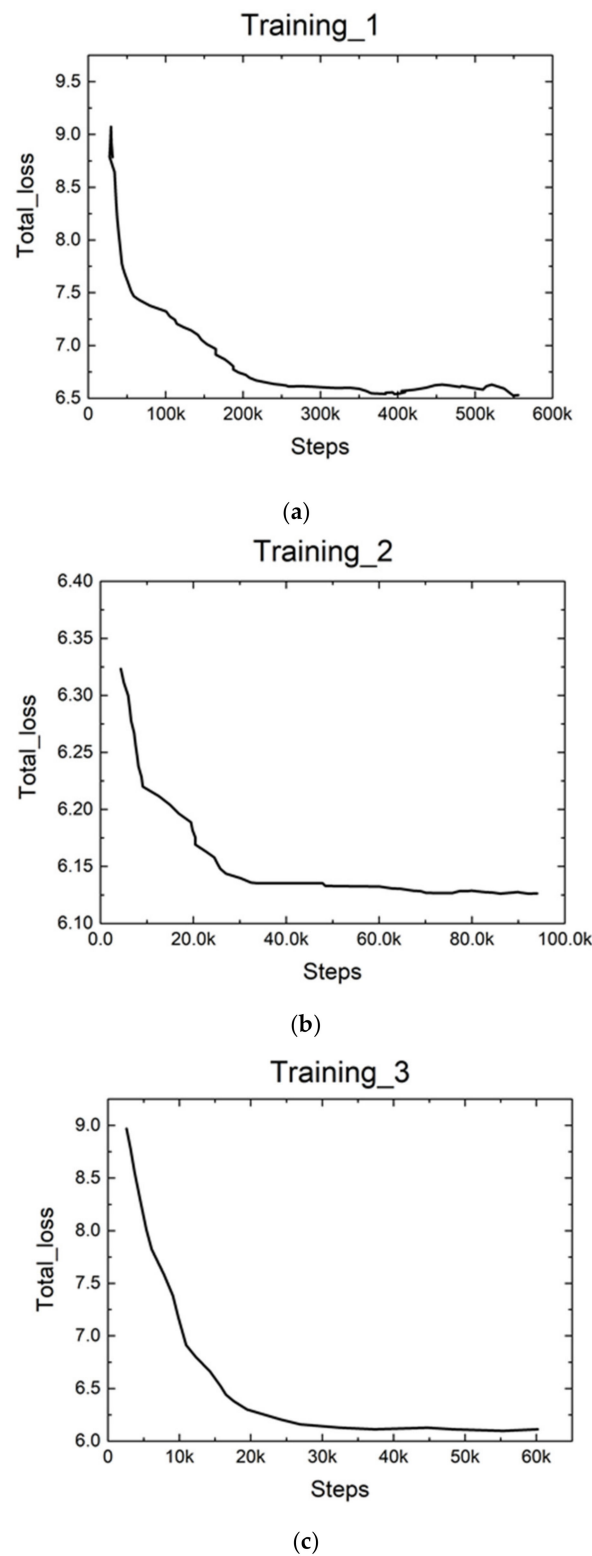


Figure 12. The total loss during the training process: (a) the total loss over the first training process; (b) the total loss over the second training process; (c) the total loss over the third training process.

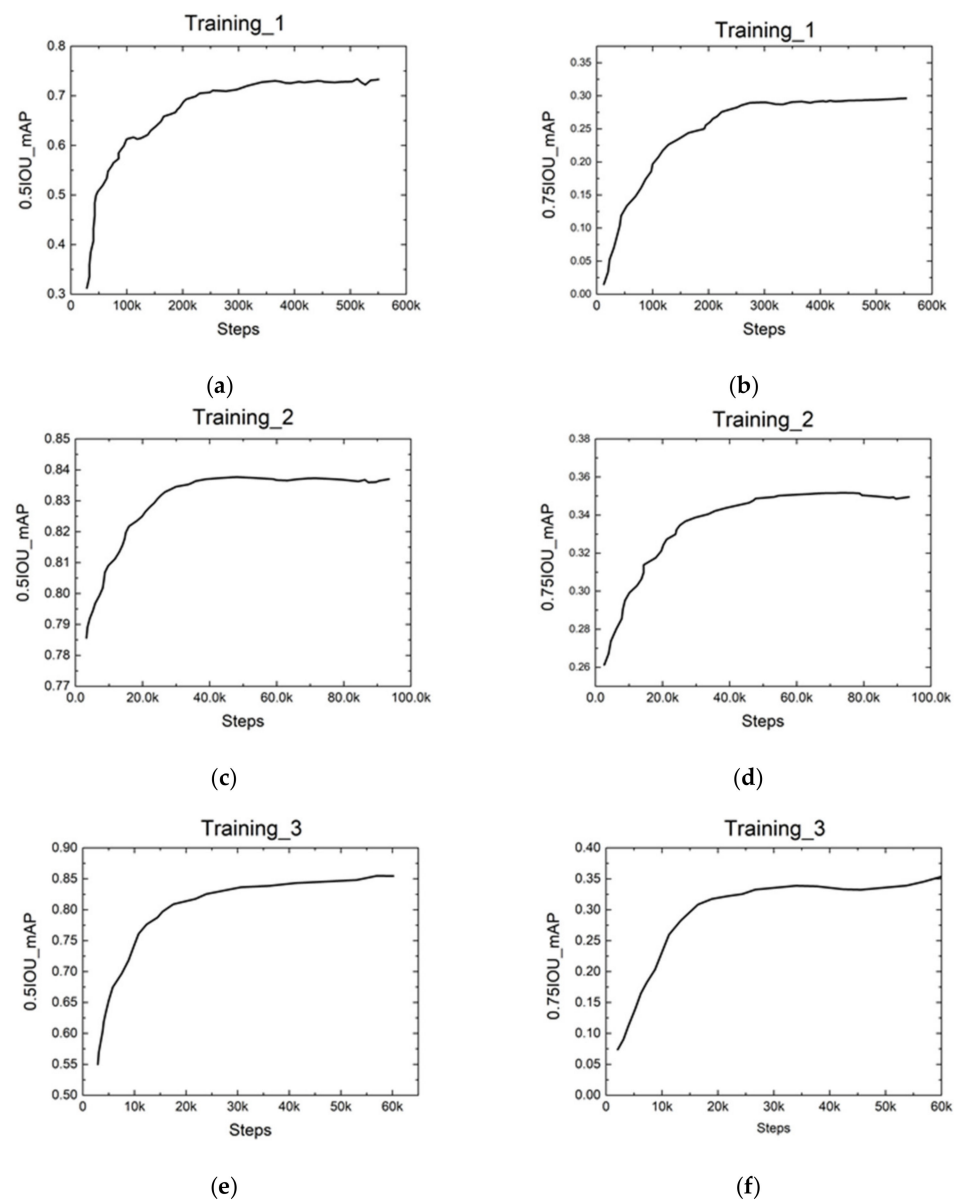


Figure 13. The results of mAP over the training process: (a,b) the results over the first training step; (a) the increase in mAP@0.5IOU over steps; (b) the increase in mAP@0.75IOU over steps; (c,d) the results over the second training step; (c) the increase in mAP@0.5IOU over steps; (d) the increase in mAP@0.75IOU over steps; (e,f) the results over the third training step; (e) the increase in mAP@0.5IOU over steps; (f) the increase in mAP@0.75IOU over steps.

3.4. Accuracy Evaluation Results

This section shows the results of evaluation for the proposed method in fatigue experiments. The fatigue experimental environment and compact tensile (CT) specimen are shown in Figure 17. A camera was used to take pictures of cracks under different fatigue load cycles. The resolution of the images was 3840×2748 . The specimen was made of aluminum alloy.

During a fatigue experiment, the crack length will increase. The crack length reference point detection method was the same as for the detection of the crack tip. The crack length could be calculated as the difference in horizontal value between the crack tip and the reference point. Then, the distance was converted into physical distance from pixel distance. The scale ruler was pasted on the surface of the specimen as shown in Figure 18. Each division on the scale was 1 mm. After calibration, we obtained the result that 1 mm was equal to 219.5 pixels, which meant 1 pixel was equal 4.56×10^{-3} mm.

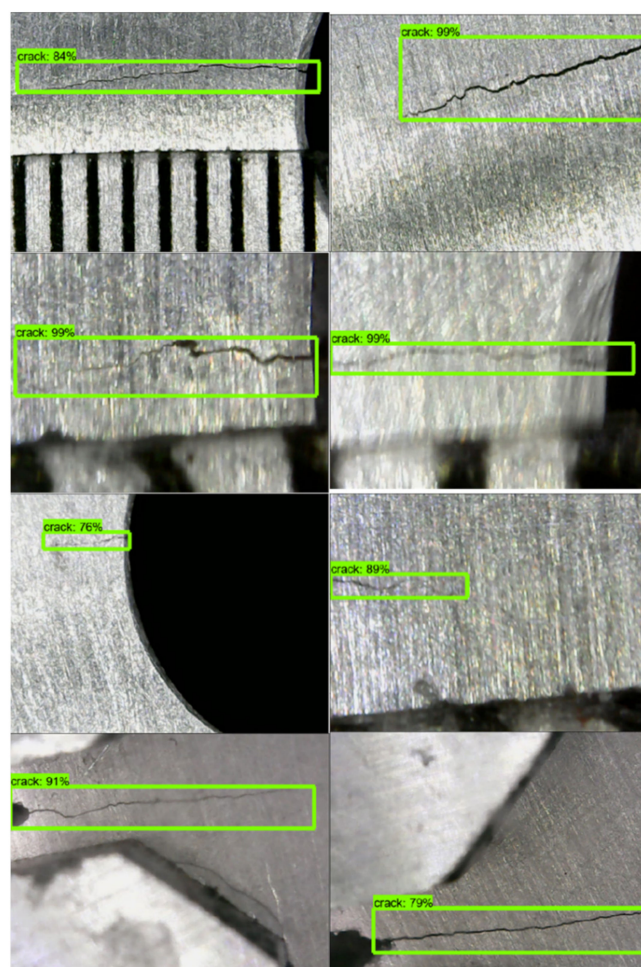


Figure 14. Crack detection results.

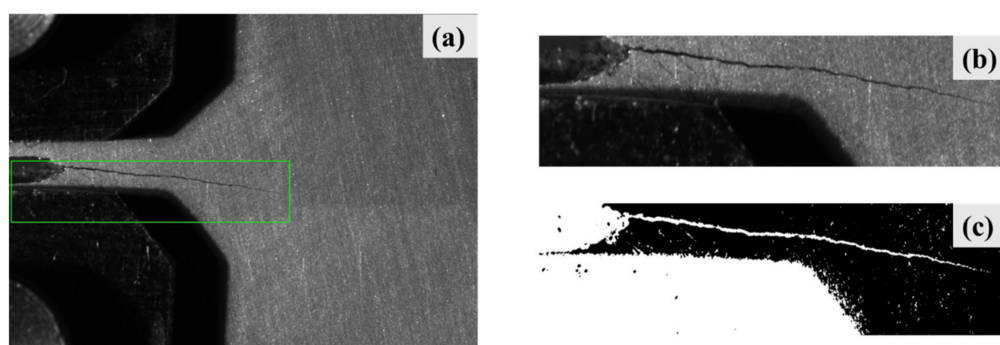


Figure 15. Results after CNN detection and threshold segmentation: (a) crack detection results; (b) extracted crack region; (c) crack region after threshold segmentation.

Figure 19 illustrates the crack length results of the proposed method and the results of human inspection. We used a measuring magnifier to check the crack length and compare the results with the proposed method. The crack length increased with the fatigue cycles. Figure 20 illustrates the absolute error and relative error at different stages. Comparing crack length results measured by the proposed method with the human inspection method, the maximum absolute error was less than 0.15 mm, and the maximum relative error was less than 2%. Figure 21 explains why the errors existed. The morphology operation not only connected the different crack areas, but also the noises and the crack. As a result, the error of crack length calculated was larger. According to the comparison results, the crack length could be measured effectively and accurately by the proposed method in this paper.

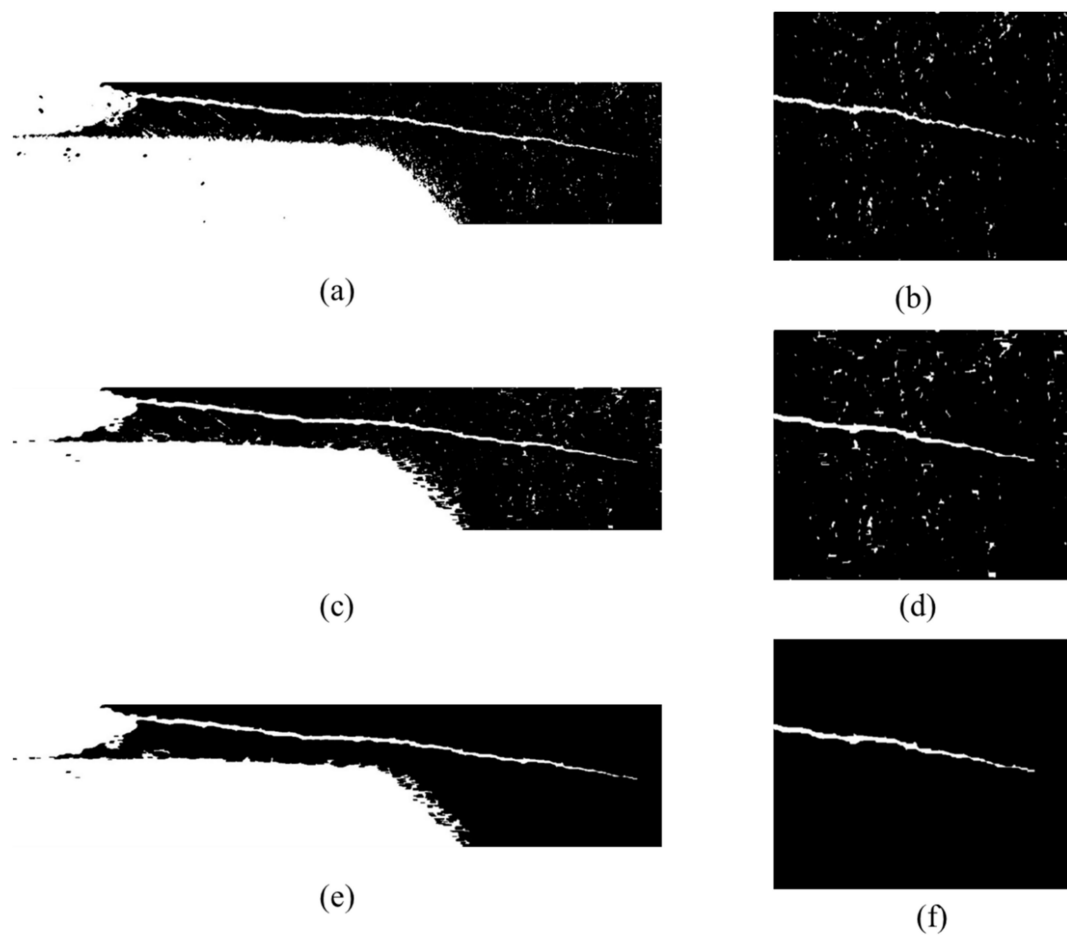


Figure 16. Results after morphological operation and small area removal: (a) crack region after threshold segmentation; (b) crack tip after threshold segmentation; (c) crack region after morphological operation; (d) crack tip after morphological operation; (e) crack region after small area removal; (f) crack tip after small area removal.

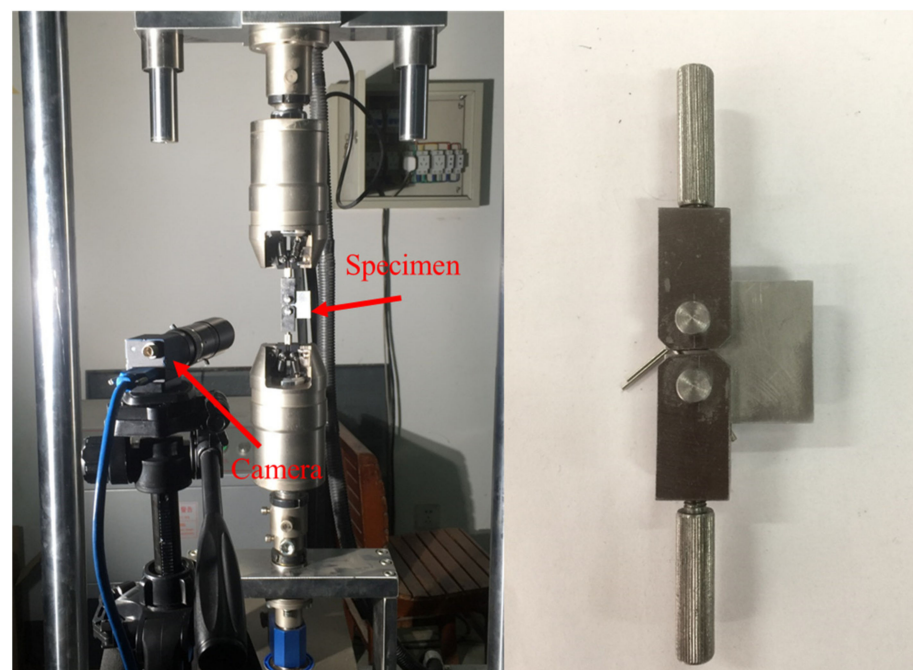


Figure 17. Experimental environment and the CT specimen.

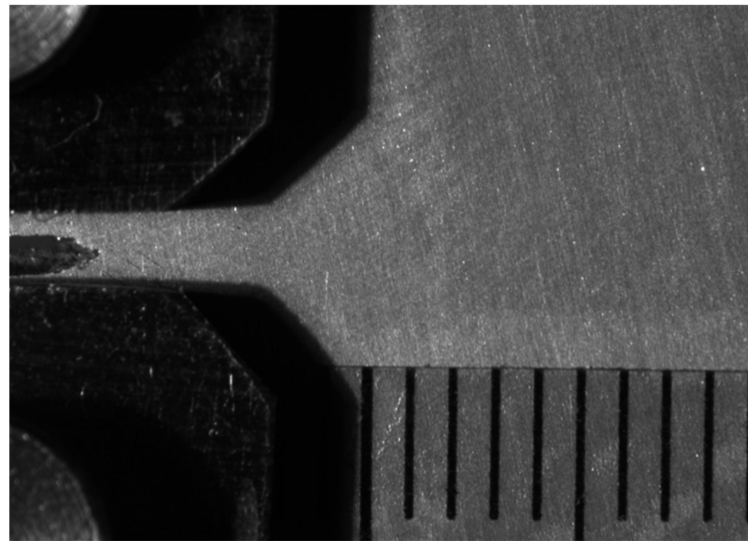


Figure 18. Crack image calibration method.

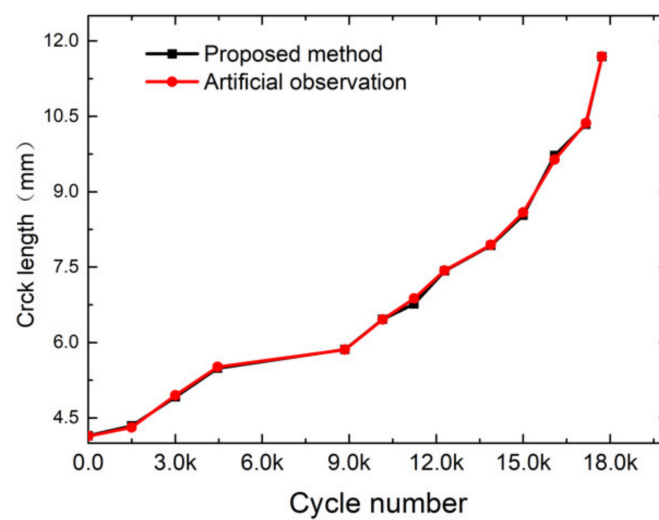


Figure 19. Crack length measurement results over fatigue experiments.

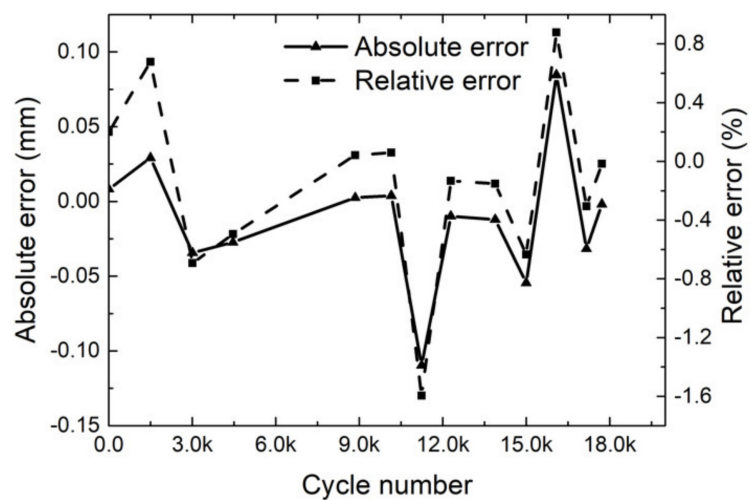


Figure 20. The absolute error and relative error of crack length.

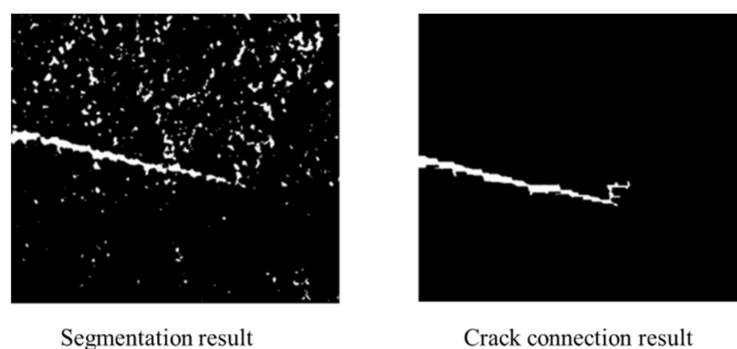


Figure 21. Image details of the reason for errors.

4. Conclusions

In this paper, a non-contact framework for the fatigue crack length measurement is presented through the study of a robust crack area location method and crack tip detection method. A crack image dataset was established for the training of CNN. The trained CNN could locate the crack accurately with the inference of crack-like edges, which enhanced the robustness of the proposed method. Then, a new algorithm for crack tip location was presented. First, the extracted crack image was converted into a binary image. Next, the morphological operation was used to connect the disturbed crack. The noises in the images were removed by setting an area threshold. Finally, the crack tip could be easily detected. The performance of the proposed method was experimentally validated through a CT specimen fatigue test. The results demonstrated that the proposed method could be used to measure crack length effectively and accurately. The method proposed in this paper could be used to identify the location of cracks from images and monitor crack length during fatigue experiments. Its application will improve the study of fracture mechanics. This work established a new framework for measuring crack length by locating the crack tip. A limitation of this study was that the image segmentation threshold was manually selected to keep the information about the crack tip. An automatic image segmentation method needs to be developed in the future. Another further study will focus on implementing this method on mobile platforms for automated crack inspection. Furthermore, crack width is an essential parameter for the serviceability of reinforced concrete construction. The proposed method will need to be modified to meet crack width measurement requirements.

Author Contributions: Y.Y. and X.G. designed the experiments; Y.Y., Z.G., X.S. and Y.L. performed the experiments and analyzed the data; Y.Y. wrote the paper; X.G., T.S. and Q.Y. supervised the study. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Nos. 12072279, 11372256, 11527803, 11602201, 11502216, and 11602202) and Natural Science Basic Research Plan in Shaanxi Province of China (No. 2018JQ1060).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

Acknowledgments: The authors wish to thank Lixuan Che for her technological help when training the crack CNN model.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Barazzetti, L.; Scaioni, M. Crack measurement: Development, testing and applications of an automatic image-based algorithm. *ISPRS J. Photogramm.* **2009**, *64*, 285–296. [\[CrossRef\]](#)
2. Gao, H.L.; Shen, S.S.; Yun, Y. Fatigue crack length real time measurement method based on camera automatically tracking and positioning. *Appl. Mech. Mater.* **2012**, *130*, 3111–3118. [\[CrossRef\]](#)

3. Ewest, D.; Almroth, P.; Sjödin, B.; Simonsson, K.; Leidermark, D.; Moverare, J. A modified compliance method for fatigue crack propagation applied on a single edge notch specimen. *Int. J. Fatigue* **2016**, *92*, 61–70. [CrossRef]
4. Chen, C.; Chen, X.; Guo, S. Experimental study on acoustic emission characteristic of fatigue crack growth of self-compacting concrete. *Struct. Control. Health Monit.* **2019**, *26*, e2332. [CrossRef]
5. Verstrynge, E.; Lacidogna, G.; Accornero, F.; Tomor, A. A review on acoustic emission monitoring for damage detection in masonry structures. *Constr. Build. Mater.* **2020**, *268*, 121089. [CrossRef]
6. Hao, X.; Du, W.; Zhao, Y.; Sun, Z.; Zhang, Q.; Wang, S.; Qiao, H. Dynamic tensile behaviour and crack propagation of coal under coupled static-dynamic loading. *Int. J. Min. Sci. Technol.* **2020**, *30*, 659–668. [CrossRef]
7. Krull, B.; Patrick, J.; Hart, K.; White, S.; Sottos, N. Automatic optical crack tracking for double cantilever beam specimens. *Exp. Tech.* **2016**, *40*, 937–945. [CrossRef]
8. Sun, H.; Liu, Q.; Fang, L. Research on fatigue crack growth detection of M (T) specimen based on image processing technology. *J. Fail. Anal. Prev.* **2018**, *18*, 1010–1016. [CrossRef]
9. Zhang, W.; Zhang, Z.; Qi, D.; Liu, Y. Automatic crack detection and classification method for subway tunnel safety monitoring. *Sensors* **2014**, *14*, 19307–19328. [CrossRef]
10. Kong, X.; Li, J. Vision-based fatigue crack detection of steel structures using video feature tracking. *Comput.-Aided Civ. Inf.* **2018**, *33*, 783–799. [CrossRef]
11. Kong, X.; Li, J. Non-contact fatigue crack detection in civil infrastructure through image overlapping and crack breathing sensing. *Automat. Constr.* **2019**, *99*, 125–139. [CrossRef]
12. Li, D.; Huang, P.; Chen, Z.; Yao, G.; Guo, X.; Zheng, X.; Yang, Y. Experimental study on fracture and fatigue crack propagation processes in concrete based on DIC technology. *Eng. Fract. Mech.* **2020**, *235*, 107166. [CrossRef]
13. Vanlanduit, S.; Vanherzeele, J.; Longo, R.; Guillaume, P. A digital image correlation method for fatigue test experiments. *Opt. Laser. Eng.* **2009**, *47*, 371–378. [CrossRef]
14. Valença, J.; Dias-da-Costa, D.; Júlio, E.; Araújo, H.; Costa, H. Automatic crack monitoring using photogrammetry and image processing. *Measurement* **2013**, *46*, 433–441. [CrossRef]
15. Yeum, C.M.; Dyke, S.J. Vision-based automated crack detection for bridge inspection. *Comput.-Aided Civ. Inf.* **2015**, *30*, 759–770. [CrossRef]
16. Dong, L.; Tang, Z.; Li, X.; Chen, Y.; Xue, J. Discrimination of mining microseismic events and blasts using convolutional neural networks and original waveform. *J. Cent. S. Univ.* **2020**, *27*, 3078–3089. [CrossRef]
17. Yu, Y.; Wang, C.; Gu, X.; Li, J. A novel deep learning-based method for damage identification of smart building structures. *Struct. Health Monit.* **2019**, *18*, 143–163. [CrossRef]
18. Cha, Y.J.; Choi, W.; Büyüköztürk, O. Deep learning-based crack damage detection using convolutional neural networks. *Comput.-Aided Civ. Inf.* **2017**, *32*, 361–378. [CrossRef]
19. Zhang, A.; Wang, K.C.P.; Li, B.; Yang, E.; Dai, X.; Peng, Y.; Fei, Y.; Liu, Y.; Li, J.; Chen, C. Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. *Comput.-Aided Civ. Inf.* **2017**, *32*, 805–819. [CrossRef]
20. Yang, X.; Li, H.; Yu, Y.; Luo, X.; Huang, T.; Yang, X. Automatic pixel-level crack detection and measurement using fully convolutional network. *Comput.-Aided Civ. Inf.* **2018**, *33*, 1090–1109. [CrossRef]
21. Guo, X.; Yuan, Y.; Liu, Y. Crack propagation detection method in the structural fatigue process. *Exp. Tech.* **2021**, *45*, 169–178. [CrossRef]
22. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision—ECCV, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
23. Howard, A.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
24. Krizhevsky, A.; Sutskever, I.; Hinton, G. Imagenet classification with deep convolutional neural networks. In Proceedings of the Conference Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
25. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S.; et al. Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017.
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
27. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef]
28. Pitas, I. *Digital Image Processing Algorithms and Applications*; Wiley & Sons: Hoboken, NJ, USA, 2000.
29. Tzutalin, D. Labellmg. Git Code. Available online: <https://github.com/tzutalin/labellmg> (accessed on 1 February 2018).
30. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014.
31. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1717–1724.