

# Identification of the Surface Cracks of Concrete Based on ResNet-18 Depth Residual Network

Rong Wang <sup>1,\*</sup>, Xinyang Zhou <sup>1</sup>, Yi Liu <sup>1</sup>, Dongqi Liu <sup>1</sup>, Yu Lu <sup>1</sup> and Miao Su <sup>1,2,\*</sup> 

<sup>1</sup> School of Civil Engineering, Changsha University of Science and Technology, Changsha 410114, China; zhouxinyang@stu.csust.edu.cn (X.Z.); dycyly@163.com (Y.L.); 13941417126@163.com (D.L.); 17373853726@163.com (Y.L.)

<sup>2</sup> Key Laboratory of Safety Control of Bridge Engineering, Ministry of Education, Changsha University of Science & Technology, Changsha 410114, China

\* Correspondence: rae\_wangr@163.com (R.W.); sumiao@csust.edu.cn (M.S.)

**Abstract:** To ensure the safety and durability of concrete structures, timely detection and classification of concrete cracks using a low-cost and high-efficiency method is necessary. In this study, a concrete surface crack damage detection method based on the ResNet-18 residual network was developed. This method was implemented by training a model with images to extract the cracks, where the image processing algorithms and deep learning were combined. The results show that the computational accuracy can meet the requirements by utilizing the established image dataset and appropriate model hyperparameters. The trained model had high recognition accuracy when the  $256 \times 256$  resolution images were adopted, and the worst accuracy of crack recognition in the test set was over 90%. The average accuracy in the test set was 91.3% when considering environmental interference generated by processing the images with a brightness adjustment, salt-and-pepper noise, and localized interference. Then, it was demonstrated that the present model possesses good robustness for crack identification in different environments. The average recognition accuracy when dealing with images of a real bridge, which are outside the training dataset, was 99.7%. The residual network model developed in this study has the advantages of low cost, high efficiency, and practicality compared to traditional detection methods. Compared to the existing deep learning methods, the model created in this study requires less computational resources and storage space, and shows a faster training speed and higher accuracy.

**Keywords:** bridge engineering; concrete cracks; deep learning; residual network; image recognition



**Citation:** Wang, R.; Zhou, X.; Liu, Y.; Liu, D.; Lu, Y.; Su, M. Identification of the Surface Cracks of Concrete Based on ResNet-18 Depth Residual Network. *Appl. Sci.* **2024**, *14*, 3142. <https://doi.org/10.3390/app14083142>

Academic Editor: Syed Minhaj Saleem Kazmi

Received: 3 March 2024

Revised: 2 April 2024

Accepted: 7 April 2024

Published: 9 April 2024



**Copyright:** © 2024 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

Health inspections of buildings play a critical role in ensuring the safety, stability, and durability of structures. These inspections typically include exterior assessments, material performance testing, structural life testing [1,2], crack testing, and other related evaluations. Researchers, both domestically and internationally, began studying concrete crack identification in the 20th century. Starting from various perspectives, researchers have developed methods for identifying cracks that are suitable for different scenarios. However, these methods come with certain limitations and restrictive conditions. In the 1950s, acoustic detection was utilized for concrete crack detection. This method offers the advantages of being non-contact and having high sensitivity. However, it is susceptible to noise interference within the concrete and has limited depth penetration. In the 1970s, resistivity detection [3] was adopted, which improved the detection depth, but required drilling, resulting in high costs and complex operations. The emergence of laser detection [4] in the early 21st century, characterized by its high precision and non-contact nature, has been significantly influenced by various environmental factors. In addition, various detection methods, such as ultrasonic detection [5] and thermal imaging detection [6], have been developed, resulting in high costs and complex operations. The researchers utilize

the aforementioned methods to detect bridge cracks, thus providing a variety of technical approaches and strategies for the timely identification and assessment of bridge cracks. However, the development and enhancement of more precise and efficient crack detection technology remains crucial to ensure the safety and reliability of bridge structures.

Currently, research in the field of concrete surface crack recognition is primarily classified into two domains: one relies on digital image processing techniques, while the other leverages deep learning methodologies.

In digital image processing-based methods, researchers primarily employ manual discriminant features to identify cracks [7–9]. They use multiple feature laws such as frequency, edge, HOG (histogram of orientation gradient), gray, texture, and entropy, and have designed some feature recognition conditions to restrict the method to complete crack recognition. This approach necessitates the manual selection and optimization of features, which may present certain limitations when addressing complex road environments and diverse crack morphologies. In a study utilizing digital image processing technology, Wang [10] employed a technique to quantitatively measure crack width and length by enumerating the number of pixels in both vertical and horizontal projections of cracks. Additionally, crack segmentation and identification were achieved through the use of the region-growing method. With these methods, the size of the crack can be accurately measured and the extent of the crack regions can be effectively determined. Liu et al. [11] utilized 3D scene reconstruction based on 2D image processing to assess the presence of cracks. V. Kosin et al. [12] proposed an innovative integrated digital algorithm that is combined with the phase-field fracture model. This algorithm has demonstrated high robustness in producing results, even when faced with various levels of noise. Malek K et al. [13] utilized digital image processing in conjunction with the Canny algorithm to detect cracks using AR technology. This approach eliminates the need for external processors in AR image processing and enables real-time image processing. Strini et al. [14] proposed a novel method based on the Euclidean graph. In contrast to traditional binary image methods, this new algorithm allows for independent selection of specific cracks according to objective criteria and accurately captures the characteristics of fracture skeletons.

On the other hand, deep learning-based approaches have been extensively employed in the field of road surface crack detection. Deep learning-based crack identification is a method that utilizes deep learning algorithms and artificial neural networks for crack detection and recognition. Researchers have developed convolutional neural networks and utilized them to automatically learn and extract data characteristics for the purpose of detecting concrete cracks [15–17]. In 2014, Szegedy et al. [18] proposed an initial architecture for deep learning without increasing the amount of calculation. In 2016, Zhang et al. [19] were the first to employ deep learning algorithms for road crack detection, subsequently enhancing the network and demonstrating that convolutional neural networks outperform support vector machines in classifying crack images. The designed network structure exhibits relative simplicity; however, the sample scenarios of the training data lack comprehensiveness. This resulted in a slow final recognition speed and low accuracy. The pixel-level classification method was introduced by Ni et al. [20], while Li et al. [21] further proposed a refined pixel-level multiple damage detection approach based on this, thereby enhancing its practical applicability. However, the detection time for images with excessively large pixel size is extended, leading to diminished efficiency.

In recent years, significant progress has been made in crack detection through the application of deep learning-based crack recognition algorithms and convolutional neural networks. These advancements have largely benefited from breakthroughs in general deep learning algorithms. Researchers have proposed many kinds of models based on convolutional neural networks, which can adapt to different environments and achieve better testing results. Additionally, they employed a multi-sensor integrated image acquisition system to mitigate overfitting and enhance the accuracy of structural defect identification. Vishal Mandal et al. [22], Wang [23], Li et al. [24], Liu et al. [25], and Zhang et al. [26] proposed and refined crack detection methods based on the YOLO family of algorithms.

The YOLO series represents an object detection algorithm that simplifies the task of detecting objects by transforming it into a single regression problem. This approach achieves real-time object detection through grid partitioning and regression of predictive boxes on the image. The YOLO series algorithms demonstrate exceptional real-time performance and detection speed, allowing for the simultaneous detection of multiple objects in images. This offers significant advantages in object detection. Wang et al. [27] proposed an efficient mobile attention X-network, MA-Xnet, for crack detection based on two models, U-Net and Dual Attention Network (DANet). Yuan et al. [28] improved U-Net and proposed ECA-UNet, which has better recognition performance than the other two models in large-scale structural crack recognition. Zhu et al. [29] proposed a method for calculating the geometric features of cracks based on the concepts of skeleton extraction and function fitting. This approach aims to better capture the shape and orientation of cracks through the utilization of the Mask R-CNN algorithm for image refinement, making it suitable for both wider and narrower cracks. However, the model necessitates additional computational resources, including increased video memory and computing power, as well as a longer training period.

Zhang et al. [30] employed an enhanced ResNet-50 as the underlying network architecture in their experiments to overcome the limitations of U-Net [31–33]. These limitations include suboptimal crack segmentation performance and difficulties in identifying narrow cracks, while also improving inference speed compared to the standard U-Net. Li et al. [34] acknowledged the limitations of ResNet-34 in handling intricate data relationships and proposed GM-ResNet, a new ResNet with favorable evaluation metrics. ResNet is a widely adopted deep convolutional neural network model that incorporates residual connections to effectively capture intricate image features. This approach addresses the issue of vanishing gradients commonly encountered in conventional deep networks. The residual learning approach facilitates the smooth flow of information across network layers by introducing cross-layer connectivity. This enhances the trainability and optimization capabilities of the network, ultimately improving its overall performance. The ResNet models commonly utilize deeper network architectures, which enable the extraction of more complex feature representations and improve the model's accuracy. In the field of crack detection, researchers have developed new models based on the ResNet module, which have yielded remarkable outcomes. Wang et al. [35] proposed a detection method for bridge pavement crack detection based on the InceptionResNet-v2 [36] module. This method can capture multi-scale features of cracks, improving efficiency without pre-training. Nirmala et al. [37] proposed a method that utilizes the principles of Pixel Intensity Similarity Measurement (PIRM) to address the issue of noise and to detect cracks. They employed VGG-16, ResNet-50, and InceptionResNet-v2 models for this purpose. The experimental results demonstrate that the ResNet-50 model exhibits superior suitability across all categories of noisy images. Yu et al. [38] proposed RUC-net, an innovative architecture that integrates UNet and ResNet. They introduced the spatial channel squeeze and excitation (scSE) attention module [39–42] while utilizing the focal loss function. The approach aims to attenuate the less relevant information and address the issue of classification imbalance, in order to optimize the crack detection method. Man et al. [43] employed the ResNet model to detect cracks and found that ResNet-50 achieved higher accuracy compared to ResNet-34. This indicates the superior performance of deep structural frameworks in identifying tunnel defects. The field of engineering places a greater emphasis on the precision of identifying cracks, making the enhancement of identification rate more valuable. The study utilizes the ResNet-18 model due to its comparable performance with ResNet-50, while also minimizing computational and memory requirements. However, it is important to exercise caution when utilizing ResNet-18 with small datasets, as it has a higher susceptibility to overfitting compared to ResNet-50. The accuracy and efficiency of model detection can be improved by creating a dataset using TensorFlow and optimizing the model with the Adam optimization algorithm.

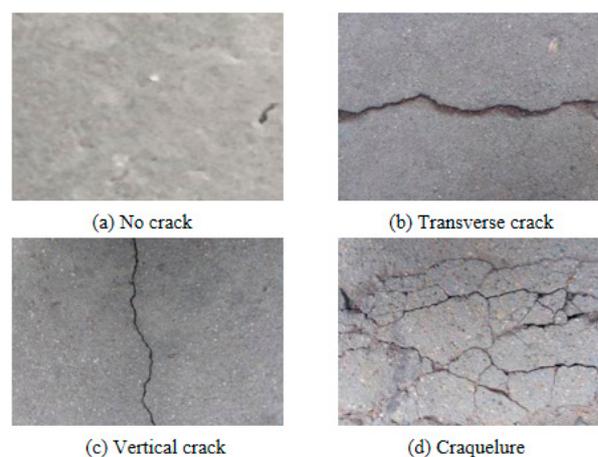
However, the previously mentioned methods still have certain limitations. For example, when dealing with complex backgrounds in images of cracks, it becomes unachievable to simultaneously identify and classify cracks in concrete imagery. In addition, the algorithm employed in those studies was implemented at an early stage, and recognition accuracy and efficiency of image processing were found to be relatively low.

Therefore, it is crucial to investigate an efficient deep neural network algorithm with reduced training time for accurate identification of cracks in concrete structures and effective discrimination of interference targets in complex scenarios. Simultaneously, the network model should possess lightweight characteristics to ensure stable operation on unmanned platforms. In this paper, an image recognition method based on the ResNet-18 deep convolutional neural network is proposed, which realizes the recognition of concrete cracks and the detection and classification of crack types. Initially, a database comprising 2000 authentic images depicting the state of concrete surfaces was meticulously curated and manually annotated for the purpose of training deep learning models. The second step involved establishing a crack recognition model based on the ResNet-18 grid, followed by conducting model training and parameter optimization. Subsequently, the model's prediction accuracy was analyzed by evaluating its recognition performance on both the training set and test set. Furthermore, this study also examined the accuracy of identifying concrete cracks in images that are affected by environmental noise interference and real bridge scenarios.

## 2. Concrete Cracking Dataset

### 2.1. Concrete Surface Image Acquisition

The process of collecting concrete crack data primarily entails conducting on-site photography and retrieving pre-existing images from online sources. The former approach utilizes ubiquitous smartphones, commonly carried by individuals in their daily lives, to collect a comprehensive dataset of 600 high-resolution ( $1028 \times 1028$ ) images depicting cracks. The latter approach primarily involves searching through the Shutterstock website (<https://www.shutterstock.com>, accessed on 12 December 2023), which allows for the collection of manipulated images captured under diverse shooting conditions and with various equipment. In this study, a dataset of 2000 images in total was collected, which was subsequently divided as presented in Table 1. The dataset consisted of 500 images depicting transverse cracks, another 500 images illustrating vertical cracks, an additional set of 500 images showcasing craquelure, and finally, a collection of 500 images representing samples devoid of any visible cracks. Exemplary images from the dataset are displayed in Figure 1.



**Figure 1.** Typical samples in the dataset.

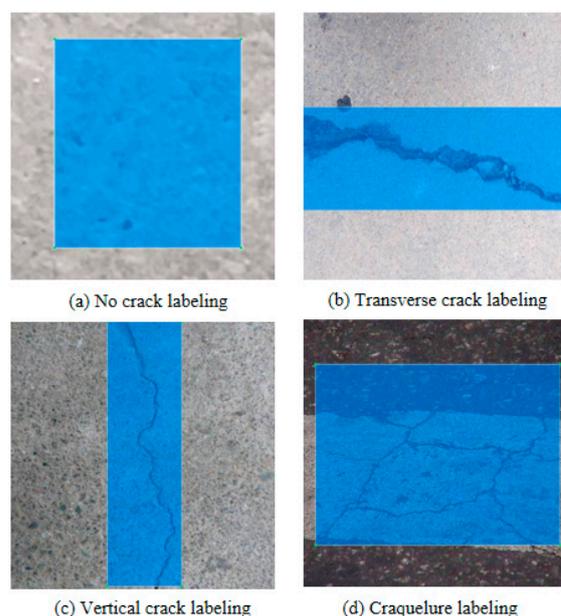
**Table 1.** Dataset of concrete images.

Image Type	Quantities	Total
No crack	500	2000
Transverse crack	500	
Vertical crack	500	
Craquelure	500	

To fulfill the prerequisites of network model training and mitigate overfitting in network model optimization, while concurrently minimizing computational expenses, we implemented adjustments by uniformly resizing the acquired images. The cropped image had a standardized resolution of  $256 \times 256$  pixels.

## 2.2. Labeling of Concrete Crack Types

The annotation of images in the dataset plays a crucial role in improving the accuracy of training model performance for detecting and recognizing concrete cracks. The image annotation tool `labellmg` (<https://github.com/tzutalin/labellmg>, accessed on 12 December 2023) was employed in this study to annotate cracks, offering the convenience of labeling multiple categories and generating XML files directly. The annotation process involves manually marking the precise location of cracks in the original image and annotating the corresponding crack type adjacent to each marker. Subsequently, an XML file is generated containing information about crack locations for utilization by deep learning models for further training. Following completion of annotations, the program randomly selects 30% of images as a test group, while the remaining 70% are allocated for training and validation purposes. The four types of crack annotations in this paper are shown in Figure 2.

**Figure 2.** The labeling of various images using `labellmg`.

Furthermore, multiple types of cracks were identified in some of the images during the image labeling process. For this particular type of image, we applied different labels to various cracks during the labeling process. As a result, the database for model training and validation contains multiple labeled cracks within the same image. Therefore, the proposed model in this study is capable of addressing the issue of image recognition that involves multiple cracks. Some example diagrams of annotations in the database containing images of multiple cracks are depicted in Figure 3.

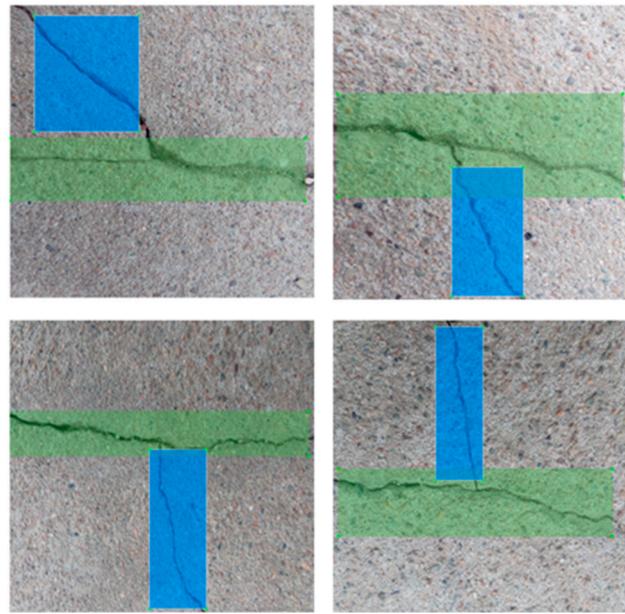


Figure 3. Labeling with multiple crack images.

### 3. Deep Learning Model

#### 3.1. ResNet-18 Network

The convolutional neural network (CNN) is a deep learning model primarily employed for image recognition and computer vision tasks. The convolution operation, which serves as the fundamental component of a CNN, extracts features by applying a filter (also known as a convolution kernel or feature detector) to the input data, as depicted in Figure 4. The primary steps of a CNN entail the traversal of a convolution layer’s filter across the input image, resulting in the generation of a feature map through element-wise multiplication and summation between the input image and the filter. By incorporating multiple filters within the convolutional layer, it becomes feasible to extract more sophisticated features. The function is commonly applied to the resultant feature map following the convolution operation. The incorporation of pooling layers is intended to decrease the spatial dimensions of the feature map and network parameters, while retaining the most significant features. After a series of convolutional and pooling layers, the resulting feature map undergoes transformation into a one-dimensional vector and subsequently traverses the fully connected layer. The output layer determines the appropriate activation function based on the specific task at hand.

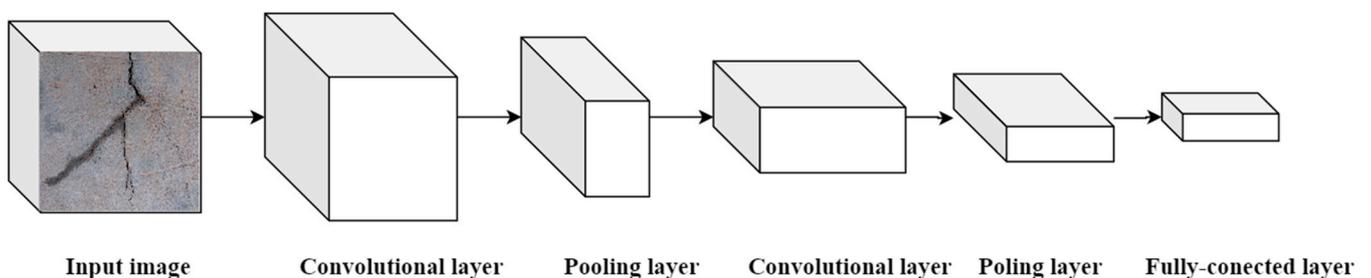


Figure 4. Convolutional neural network data handling processes.

During the training process, the convolutional neural networks (CNNs) utilize the backpropagation algorithm to iteratively update the network’s weights and biases, facilitating gradual acquisition of feature representations from input data. The utilization of convolution operations and hierarchical structures in convolutional neural networks

enables effective feature extraction and learning from images, thereby enhancing robust image processing and analysis capabilities.

The deep residual network (DRN) is a deep neural network architecture commonly employed for image classification and deep learning tasks in scientific research. Traditional deep neural networks encounter challenges such as the problem of vanishing and exploding gradients, which impede effective training with increasing network depth. Moreover, the growing number of parameters and computational complexity exacerbates this problem. The incorporation of residual connections in deep residual networks effectively tackles the challenge of training deep networks, thereby enabling the implementation of more intricate network architectures and facilitating optimization.

ResNet introduces residual blocks, which consist of identity maps and residual maps. The fundamental form of residual blocks can be expressed as follows:

$$y = F(x, \{W_i\}) + x \tag{1}$$

where  $x$  is the input,  $F$  is the residual mapping,  $\{W_i\}$  represents the learnable parameters in the residual map (such as convolution kernel weights), and  $y$  is the output of the residual block.

The primary innovation of ResNet lies in the introduction of residual learning, a concept that has significantly advanced deep neural network architectures. In the conventional deep convolutional neural network, input data are processed through one or more layers to obtain the output. However, in a residual network, an additional skip connection is incorporated alongside the main pathway of network layers. This allows for direct skipping of one or more layers and subsequent addition to the output of the main pathway. The introduction of ResNet effectively addresses the pervasive issues of gradient vanishing and explosion in deep neural networks, thereby enabling the network to safely increase its depth for improved performance.

The ResNet-18 functional model was employed in this study based on practical requirements and available computational resources. It encompasses 18 layers (excluding the activation function and pooling layer), rendering it the smallest model within the ResNet series. By virtue of its residual structure, ResNet-18 facilitates direct backpropagation of gradients to earlier layers, effectively mitigating the issue of gradient vanishing. Moreover, in comparison to the deeper models within the ResNet series, such as ResNet-50 and ResNet-101, the network structure diagram of ResNet-18 is illustrated in Figure 5. The present model exhibits a reduced scale, necessitating fewer computational resources and storage capacity; nevertheless, it also showcases enhanced training and inference speed alongside improved performance.

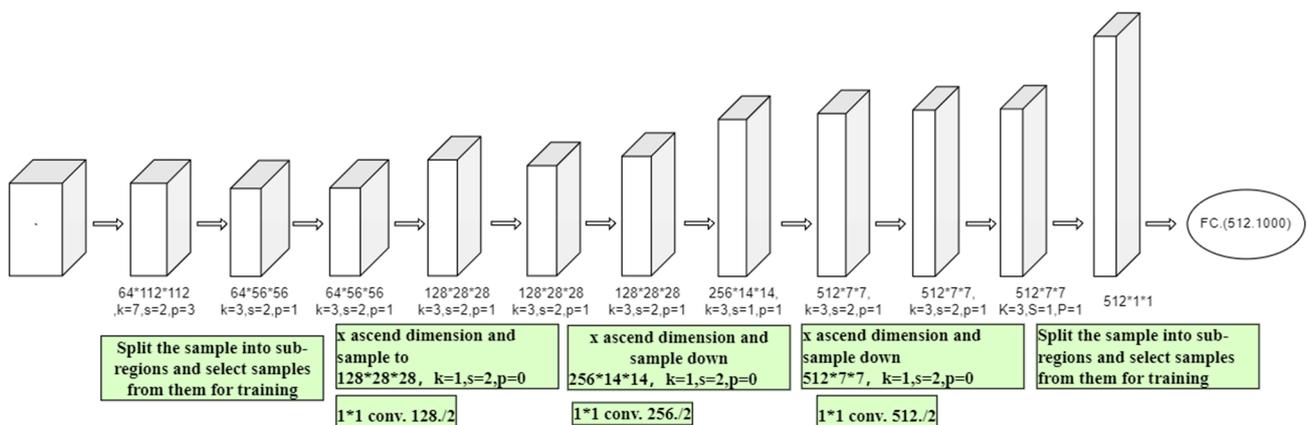


Figure 5. ResNet-18 network structure.

### 3.2. Selection of Hyperparameters

The model was compiled using the Adam optimizer and the cross-entropy loss function in this study. In the training process, the input of training set data leads to prediction results through forward propagation. The obtained results are subsequently compared with the actual labels, and loss is computed to update model parameters through back propagation, as depicted in Figure 6. The model’s performance is assessed by calculating the loss and accuracy on the validation set, which serves as an indicator of its generalization ability. The hyperparameters are fine-tuned and the model architecture is refined based on the performance of the model on the validation set and the weight file obtained from the initial round of training, with the aim of optimizing and enhancing its overall performance. Upon completion of model training, the performance of the model is assessed by evaluating it using the test dataset to obtain objective metrics such as recognition accuracy and loss rate.

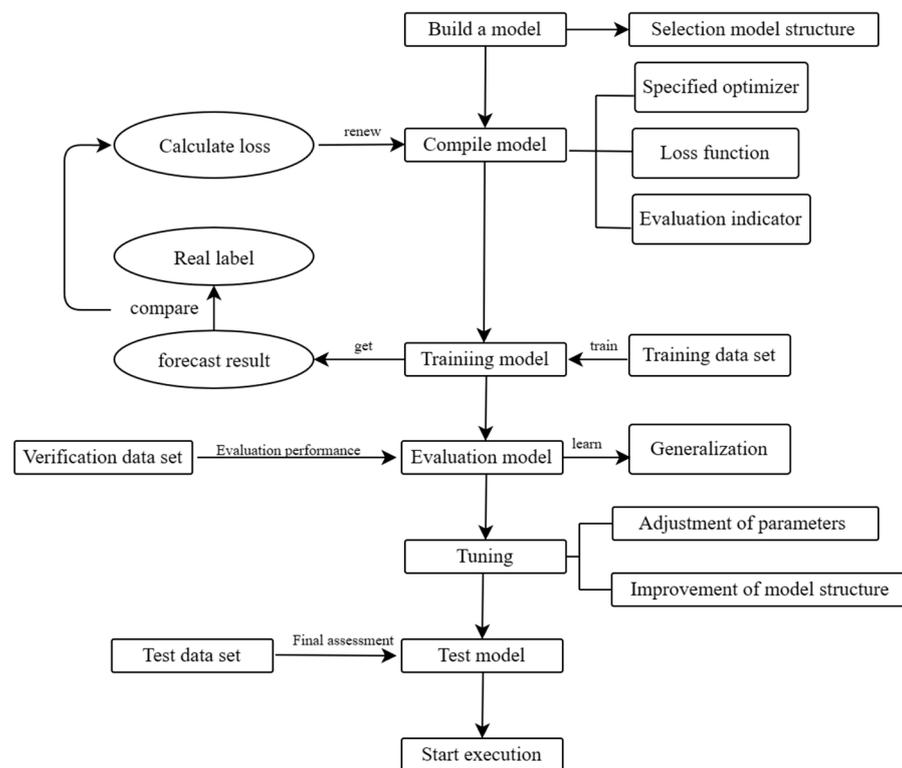


Figure 6. Training process of the model.

The incorporation of the rectified linear unit (ReLU) activation function in this study greatly amplified the expressive capacity, generalization capability, and training efficiency of neural networks through the introduction of nonlinearity, facilitation of sparse activation patterns, and mitigation of gradient vanishing problems, while ensuring stability and expedited computation. The mathematical expression of ReLU is as follows:

$$\text{ReLU} = \begin{cases} X, & x > 0; \\ 0, & x \leq 0. \end{cases} \quad (2)$$

The function expression demonstrates that the rectified linear unit (ReLU) effectively sets all values less than or equal to zero, while preserving values greater than zero.

The model learning rate is a crucial hyperparameter employed for updating the model parameters during the training process of a neural network. It determines the degree to which the product of the gradient and learning rate influences parameter changes in each update. If the learning rate is set too low, it will significantly decelerate parameter updates,

thereby impeding the training process and potentially obstructing convergence towards the optimal solution. Excessively high learning rates can lead to a rapid and unstable pace of parameter updating, potentially impeding the training process or even causing failure to converge. The learning rate formula of the piecewise constant decay formula can be expressed as:

$$lr = lr_0 \cdot \gamma^{(\text{int}(T/T'))} \quad (3)$$

where  $lr$  is the current learning rate,  $lr_0$  is the initial learning rate,  $\gamma$  is the attenuation factor,  $T$  is the current training cycle (epoch), and  $T'$  is the number of cycles in each stage.

The Adam optimization function was employed in this study to dynamically adjust the learning rate during training, enabling adaptation to gradient changes across different parameters. This approach accelerates convergence speed and facilitates quicker attainment of improved convergence effects. The Adam optimization function is a gradient-based optimization algorithm that utilizes estimations of first-order and second-order moments. The algorithm exhibits exceptional adaptability to handling large-scale datasets and sparse gradients, making it highly effective in tackling optimization challenges encountered in such scenarios. Furthermore, the algorithm can be easily implemented by solely storing the first- and second-moment estimations of each parameter, without requiring any additional memory for past gradient storage. Its mathematical expression is as follows:

$$mt = \beta_1 \cdot m + (1 - \beta_1) \cdot \delta \quad (4)$$

$$vt = \beta_2 \cdot v + (1 - \beta_2) \cdot \delta^2 \quad (5)$$

In this context,  $mt$  represents the first-order moment and  $vt$  represents the second-order moment.  $\delta$  represents the instantaneous gradient.  $\beta_1$  and  $\beta_2$  are hyperparameters that control the decay rates of the first- and second-moment estimates, with values of 0.9 and 0.999, respectively. In each iteration, the first-order moment  $mt$  and second-order moment  $vt$  are updated based on the current gradient. By continuously estimating the first and second moments of the current gradient, the learning rate is adaptively adjusted.

Specifically, the learning rate decay formula is as follows:

$$lr_t = \alpha \cdot \sqrt{\frac{(1 - \beta_2^t)}{(1 - \beta_1^t)}} \quad (6)$$

where  $lr_t$  represents the learning rate decay,  $t$  is the current iteration number, and  $\alpha$  is the initial learning rate. The decay of  $lr_t$  consists of two components controlling the decay of the second-order moment and the first-order moment.

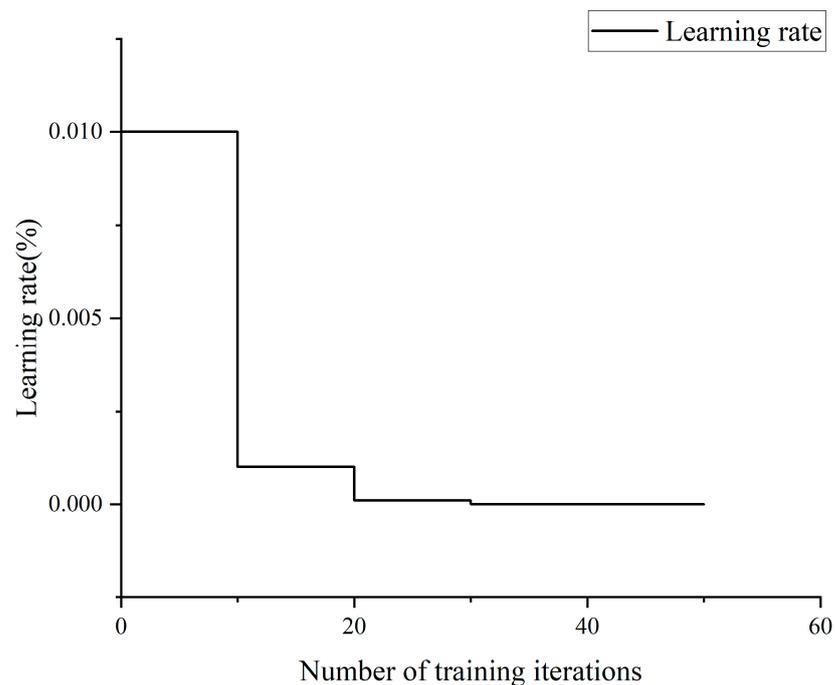
$$W_t + 1 = W_t - lr_t \cdot mt / (\sqrt{vt} + \epsilon) \quad (7)$$

where  $W_t + 1$  is the parameter of the iterative model at time  $t$ , and  $\epsilon$  is a small number used for numerical stability, with a value of  $1.0 \times 10^{-8}$ . In each iteration, the parameter  $W_t$  is updated through the learning rate  $lr_t$  based on the current gradient and the estimation of the first-order moment  $mt$  and the second-order moment  $vt$ .

The initial learning rate for training in this study was set at 0.01 and subsequently adjusted during the training iterations. The piecewise constant decay method was utilized in this study. After completing 10 training cycles, the learning rate was decreased to 0.1 times its initial value, as described in detail in Table 2 and Figure 7. The learning rate scheduling strategy facilitates a gradual decrease in the learning rate throughout the training process, promoting a more meticulous exploration of optimal model parameters. An elevated initial learning rate promotes the model's rapid convergence towards the global optimum during early stages of training, followed by a gradual decrease in the learning rate to ensure a more stable and consistent convergence as training progresses. The proposed attenuation method can significantly enhance both the training efficiency and accuracy of the model.

**Table 2.** Learning rates with training epochs.

Learning Rate	Training Epochs
0.01	0
0.01	10
0.001	10
0.001	20
0.0001	20
0.0001	30
0.00001	30
0.000001	40
0.000001	50

**Figure 7.** Learning rate change graph.

### 3.3. Model Implementation

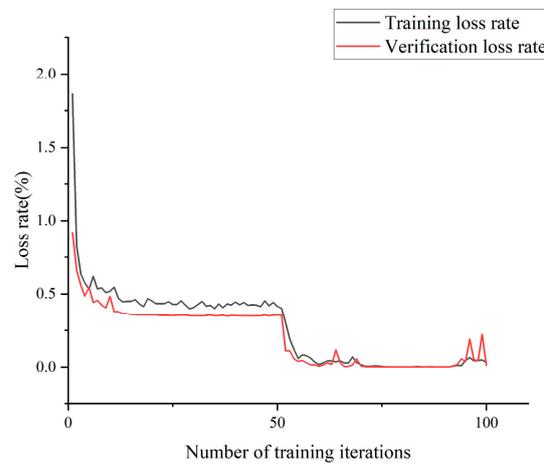
The adoption of TensorFlow as the primary data processing methodology in this study aimed to optimize the seamless operation of the model. The TensorFlow framework offers an efficient and scalable approach for managing large-scale datasets, encompassing preprocessing steps to enhance the quality and diversity of data prior to model ingestion. Additionally, it facilitates batch processing and enables the partitioning of data into smaller batches for training purposes. The TensorFlow framework can also be employed for parallel data processing, where the overlap between data ingestion and model computation is leveraged to optimize the overall efficiency of data processing.

Concurrently, in order to accommodate the diverse functionalities required in model recognition processes, this study employed the PyTorch learning library for loading and processing training data, test data, and validation data. The PyTorch framework enables the customization of neural network architectures, facilitates tracking of tensor operations, and automates gradient calculation, thereby augmenting the efficiency of data processing.

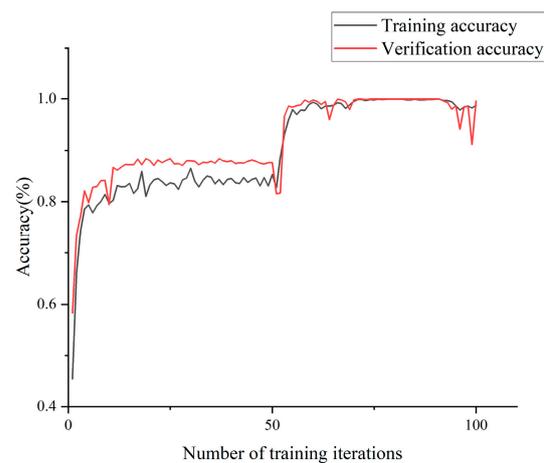
#### 4. Model Performance Analysis and Discussion

##### 4.1. Concrete Crack Identification Accuracy

The model was trained using the aforementioned dataset, with a resolution set at  $256 \times 256$  pixels. The loss rate and accuracy exhibited variations throughout the training iterations, as depicted in Figures 8 and 9, respectively.



**Figure 8.** Change curve of model loss rate during training.

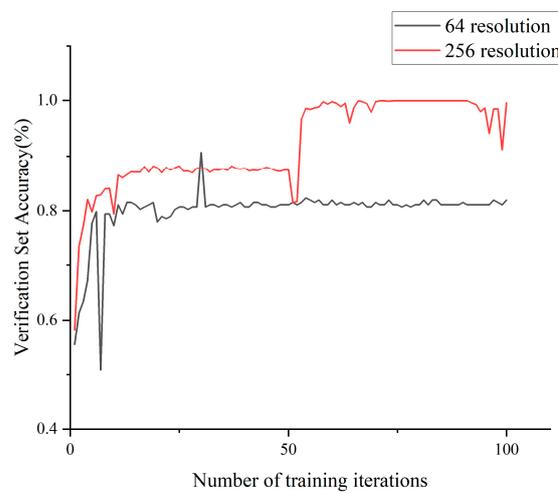


**Figure 9.** Change curve of model accuracy during training.

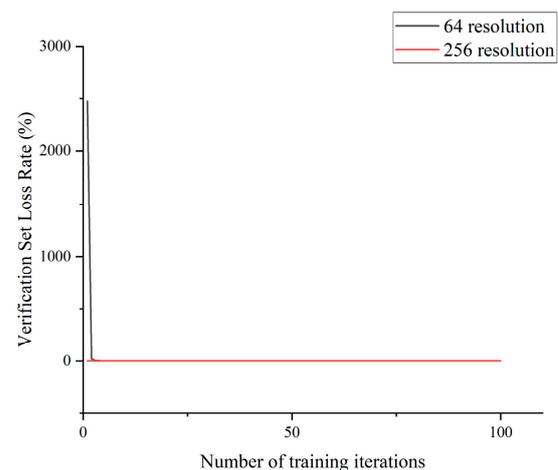
In terms of model parameter selection, resolution plays a pivotal role in determining model accuracy and loss rate, while also significantly impacting the volume of input data utilized by the model. Higher resolution entails an increased number of pixels and more intricate information, thereby leading to larger input dimensions and heightened computational demands. Conversely, lower resolutions effectively reduce the dimensionality of inputs and alleviate computational complexity. Therefore, the careful selection of an appropriate resolution facilitates effective control over the computational complexity of the model while simultaneously meeting practical requirements. In practical applications, the selection of appropriate image resolution parameters necessitates considering the characteristics of training data, computational resources available for the model, and application scenarios.

Considering the limited computational resources and the low-resolution nature of crack images, it was advisable to restrict the training resolution in this study to a maximum of  $512 \times 512$  pixels. The resolution was set to  $64 \times 64$ , while keeping the other parameters unchanged. The model was subsequently retrained, and its loss rate and accuracy were

compared against a validation set with a resolution of  $256 \times 256$ . The comparative results are presented in Figures 10 and 11.



**Figure 10.** Verification set model accuracy comparison graph.



**Figure 11.** Verification set model loss rate comparison graph.

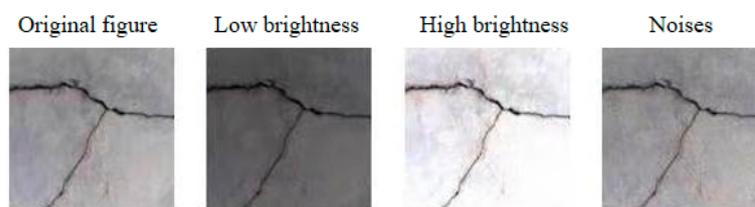
According to the training results, when the resolution was set to  $256 \times 256$ , the model demonstrated consistent accuracy and loss rates, aligning with our experimental expectations. However, excessively low resolution may result in a loss of intricate details, impeding the model's ability to effectively capture and learn subtle changes or features within images. After conducting a meticulous analysis of the experimental findings and taking into account computational complexity, we ascertained that configuring the model resolution parameter to  $256 \times 256$  would yield superior performance in our experiment.

#### 4.2. Performance of Image Environment Interference

Due to various external environmental factors or limitations in the shooting equipment during the process of collecting fractures, inconsistencies may arise in the quality of the collected fracture images, leading to image interference. The training process of the model takes into consideration this issue, addressing both overall and local interference through image processing and model training.

The overall interference of the image is first taken into consideration. By incorporating the overall interference, a range of potential external factors can be simulated, including noise during image transmission and distortion resulting from image compression. By incorporating global interference into the image, the model can acquire a wider range

of feature representations, enhance its adaptability to real-world interference scenarios, and demonstrate improved versatility across various types of interferences. The results of processing the original dataset images to simulate possible scenarios are shown in Figure 12. The model's robustness is improved by utilizing data augmentation techniques, which generate various datasets by introducing variations in brightness and adding noise to the original image. The following methods were employed: Python and the OpenCV library were utilized to implement a random selection process for determining pixel locations within our image database based on the noise level. Subsequently, these selected pixels were assigned with either the brightest or darkest color in order to introduce salt-and-pepper noise. The image's brightness was manipulated by multiplying each pixel value with the corresponding adjustment factor, incorporating both the original image and parameters for brightness adjustment. To ensure that our results fell within an acceptable range of expression, we constrained all pixel values between 0 and 255.



**Figure 12.** Image processing result.

The addition of salt-and-pepper noise can effectively simulate the presence of noise interference that images may encounter in real-world scenarios, including sensor noise, camera noise, and other similar disturbances. By introducing salt-and-pepper noise through random manipulation of the brightest or darkest pixels in an image, this technique enhances the adaptability and effectiveness of the model in handling distractions during both training and reasoning processes. Incorporating salt-and-pepper noise into the training data enables the model to acquire a more robust feature representation and enhance its resilience to noise. Furthermore, revising the image brightness can effectively address variations in image appearance under different lighting conditions. The classification performance of images under diverse illumination conditions can be significantly improved by optimizing the image brightness, enabling effective adaptation to variations in light and dark levels. The adjustment of brightness can be achieved by precise calibration of pixel values, ensuring a controlled modification in overall luminance without inducing overexposure or underexposure.

The model was trained using a dataset generated through various processing methods, based on the aforementioned training parameters. The training results are presented in Table 3.

**Table 3.** Training results of different datasets.

Treatment	Accuracy	Loss Rate
original	0.9286	0.3175
Low brightness	0.9003	0.5660
High brightness	0.9151	0.4571
Add noise	0.9244	0.3626

Additionally, local image interference was effectively mitigated by adjusting parameters such as brightness, introducing controlled noise, or modifying localized textures to accurately simulate specific interferences in different regions of the image. These interferences encompass uneven illumination, localized faults, and dirt. By incorporating local noise into the image, the model can effectively integrate features that demonstrate enhanced sensitivity to localized perturbations induced by noise. The aforementioned training significantly improves the model's capacity to precisely identify and classify images amidst

various local disruptions, consequently enhancing its accuracy and resilience for real-world implementations. Taking uneven illumination as an example, we selected a naturally illuminated image containing a crack as the original sample to investigate the impact of local variations in illumination intensity on the recognition accuracy of this model. The effects of local shadows and intense light in specific areas of the image were simulated using various image processing techniques, including the application of mask layers, adjustment of pixel values, and filtering. Consequently, three distinct crack images were obtained. The model was trained using the three crack images for the purpose of image recognition. Figures 13–15 depict the original crack images along with their corresponding length and width parameters, as well as the results obtained from the model’s recognition process.

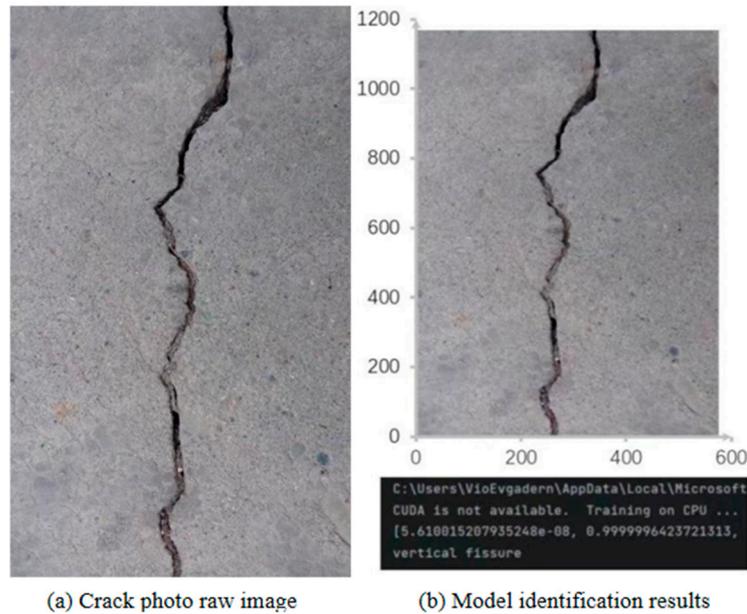


Figure 13. Crack image recognition results under uniform illumination.

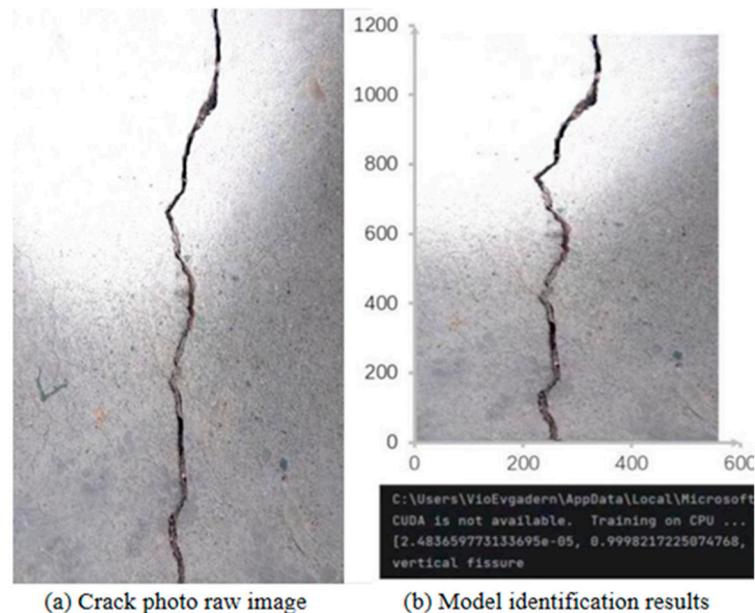
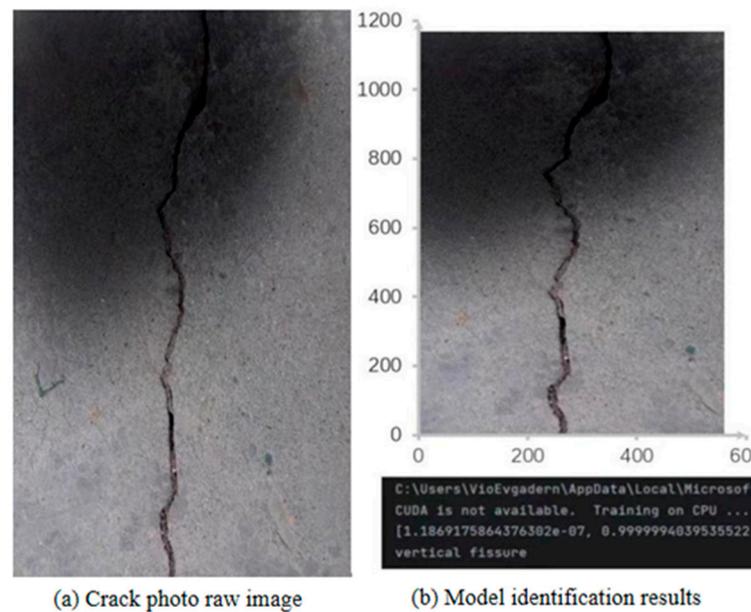


Figure 14. Crack image recognition results under local strong light.



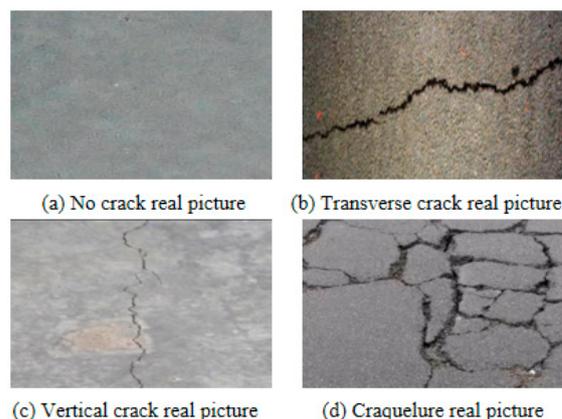
**Figure 15.** Crack image recognition results under local shadow.

Based on the recognition results obtained from the model output, it can be observed that this model accurately assigns labels to different crack types in the image even under varying local light intensities. This indicates that the proposed model demonstrates robust classification performance in the presence of local external interferences. The significance of this finding lies in the model's ability to effectively acquire and comprehend the characteristics of cracks under varying lighting conditions, accurately associating them with their corresponding labels. This observation suggests that the model has attained a sufficient level of proficiency in abstract expression capabilities during the training phase, enabling it to effectively extract crack-related information from local variations in light. Despite variations in lighting conditions affecting specific regions of the image, the model consistently exhibited precise classification and accurate label output. The aforementioned statement highlights the model's exceptional adaptability to local variations in illumination, showcasing its remarkable ability to accurately classify data under intricate real-world conditions.

In conclusion, the incorporation of global and local interference into the image enhances the model's robustness, ensuring that external factors do not impact its recognition outcomes. This processing approach effectively improves the image classification performance of the model under diverse interference factors, making it more suitable for real-world applications.

#### 4.3. Performance of Real Bridge Images Outside the Dataset

Finally, in order to investigate the practical applicability of the model in real-world scenarios, two approaches are employed for gathering authentic images depicting bridge cracks. The first approach involves conducting on-site visits to directly observe the cracks present on various bridges and gather corresponding images. The second approach entails searching for authentic images available on the Internet. Subsequently, a total of 30 photographs were collected from each of the four categories and employed to evaluate the precision of the trained model through recognition. The evaluation of the model's performance in real-world scenarios and the provision of feedback for further improvement and optimization were facilitated. The objective of this identification verification experiment was to validate the practicality of the model, thereby providing a reliable tool and judgment basis for bridge safety detection in various fields. An example of the collected live crack images is shown in Figure 16, and the model output is shown in Table 4.



**Figure 16.** Example of fissures in real pictures.

**Table 4.** Real photo recognition results.

Type of Fissure	Number of Actual Pictures	Accuracy	Best Recognition Rate
No crack	30	100%	99.90%
Transverse crack	30	96.70%	99.10%
Vertical crack	30	96.00%	99.90%
Craquelure	30	93.30%	99.90%

According to the results in the table, it can be seen that the model performed best in recognizing crack-free images with 100% accuracy, where the best recognition rate was 99.90%. The recognition accuracy for cracks and longitudinal cracks was relatively low, but the best recognition rate was 99.90%. Based on the analysis of the image, it is plausible that the recognition of the model could be compromised due to bifurcation of vertical cracks, resulting in a certain probability of misclassification as craquelures.

In summary, the model demonstrated robust recognition capabilities in practical applications and was able to accurately classify various types of cracks. Relevant professionals can promptly comprehend and implement appropriate measures to ensure the smooth operation of structures and facilities by effectively categorizing cracks. Consequently, the exemplary performance of this model will yield positive ramifications for real-world engineering applications by enhancing efficiency and mitigating risks. As a result, it holds significant implications for both the engineering industry and societal development.

## 5. Conclusions

Aiming at the problem of concrete crack detection problem, a well-trained concrete crack image recognition model was realized based on ResNet-18. The main conclusions are as follows:

- (1) A dataset of concrete surface cracks was established by setting appropriate image parameter features, and a model based on the ResNet-18 residual neural network was developed. The dataset was used to conduct multiple epochs of training for the model. The appropriate hyperparameters were chosen so that the model has a high recognition accuracy, and the results show that the worst accuracy of the model created in this paper in the test set was over 90%.
- (2) Considering the impact of the objective factors of the outside world during the collection process, the method of data enhancement was used to process the image in the original dataset before training iteration. The accuracy of the model training under different environmental interference was 91.3%, which proves the robustness of the model.
- (3) The accuracy of the image recognition model was analyzed under varying lighting conditions. The accuracy rate of recognition results under each condition was more

than 99%. This verifies the good accuracy and stability of this model for dealing with environmental interference.

- (4) The actual photo collected from an actual concrete bridge was further analyzed to verify the efficiency of the model. The average recognition rate for various crack types was more than 99.7%. The model established in this study holds potential for practical engineering applications in crack identification.

**Author Contributions:** Conceptualization, R.W. and M.S.; methodology, X.Z.; software, Y.L. (Yi Liu) and Y.L. (Yu Lu); validation, D.L.; formal analysis, Y.L. (Yi Liu) and R.W.; investigation, Y.L. (Yu Lu); data curation, X.Z.; writing—original draft preparation, R.W.; writing—review and editing, M.S.; visualization, X.Z.; supervision, M.S.; funding acquisition, R.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the project of Undergraduate Innovation and Entrepreneurship Training of Changsha University of Science and Technology (Grant No. 202210536008).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Ma, Y.F.; He, Y.; Wang, L.; Zhang, J. Probabilistic reconstruction for spatiotemporal sensor data integrated with Gaussian process regression. *Probabilistic Eng. Mech.* **2022**, *69*, 103264. [[CrossRef](#)]
2. Ma, Y.F.; He, Y.; Wang, L.; Zhang, J. Probabilistic life prediction for reinforced concrete structures subjected to seasonal corrosion-fatigue damage. *ASCE J. Struct. Eng.* **2020**, *146*, 04020117. [[CrossRef](#)]
3. Kimmo, K.; Aku, S.A.; Jari, P.K. Adaptive meshing approach to identification of cracks with electrical impedance tomography. *Inverse Probl. Imaging* **2014**, *8*, 127–148.
4. Khong, H. Application of Laser Shearography for Detecting the Cracks in Concrete. Master's Thesis, University of Maryland, College Park, MD, USA, 2007.
5. Kang, D.; Cha, Y.J. Autonomous UAVs for Structural Health Monitoring Using Deep Learning and an Ultrasonic Beacon System with Geo-Tagging. *Comput. Aided Civ. Infrastruct. Eng.* **2018**, *33*, 885–902. [[CrossRef](#)]
6. Tomita, K.; Lin, C.M.Y. A review of infrared thermography for delamination detection on infrastructures and buildings. *Sensors* **2022**, *22*, 423. [[CrossRef](#)] [[PubMed](#)]
7. Bang, Y.L.; Yun, Y.K.; Seong-Tae, Y.; Jin-Keun, K. Automated image processing technique for detecting and analysing concrete surface cracks. *Struct. Infrastruct. Eng.* **2013**, *9*, 567–577.
8. Kim, H.; Ahn, E.; Shin, M.; Sim, S.H. Crack and Non-Crack Classification from Concrete Surface Images Using Machine Learning. *Struct. Health Monit.* **2019**, *18*, 725–738. [[CrossRef](#)]
9. Kim, H.; Lee, J.; Ahn, E.; Cho, S.; Shin, M.; Sung-Han, S. Concrete crack identification using a UAV incorporating hybrid image processing. *Sensors* **2017**, *17*, 2052. [[CrossRef](#)] [[PubMed](#)]
10. Wang, C.Y. Research on Crack Detection Method of Bridge Bottom Surface Based on Image Recognition Processing. Doctoral Dissertation, Beijing Jiaotong University, Beijing, China, 2016; pp. 24–26. [[CrossRef](#)]
11. Liu, Y.F.; Cho, S.; Spencer, B.F.; Fan, J.S. Concrete Crack Assessment Using Digital Image Processing and 3D Scene Reconstruction. *J. Comput. Civ. Eng.* **2016**, *30*, 04014124. [[CrossRef](#)]
12. Kosin, V.; Fau, A.; Jailin, C.; Hild, F.; Wick, T. Parameter identification of a phase-field fracture model using integrated digital image correlation. *Comput. Methods Appl. Mech. Eng.* **2024**, *420*, 116689. [[CrossRef](#)]
13. Malek, K.; Mohammadkhorasani, A.; Moreu, F. Methodology to integrate augmented reality and pattern recognition for crack detection. *Comput. Aided Civ. Infrastruct. Eng.* **2023**, *38*, 1000–1019. [[CrossRef](#)]
14. Strini, A.; Schiavi, L. Euclidean graphs as crack pattern descriptors for automated crack analysis in digital images. *Sensors* **2022**, *22*, 5942. [[CrossRef](#)] [[PubMed](#)]
15. Cha, Y.J.; Choi, W.; Büyüköztürk, O. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Comput. Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [[CrossRef](#)]
16. Kim, B.; Cho, S. Automated Vision-Based Detection of Cracks on Concrete Surfaces Using a Deep Learning Technique. *Sensors* **2018**, *18*, 3452. [[CrossRef](#)] [[PubMed](#)]
17. Kim, B.; Yuvaraj, N.; Sri Preethaa, K.R. Surface crack detection using deep learning with shallow CNN architecture for enhanced computation. *Neural Comput. Appl.* **2021**, *33*, 9289–9305. [[CrossRef](#)]

18. Szegedy, C.; Liu, W.; Jia, Y. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
19. Zhang, L.; Yang, F.; Zhang, Y.D.; Zhu, Y.J. Road crack detection using deep convolutional neural network. In Proceedings of the International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3708–3712. [\[CrossRef\]](#)
20. Ni, F.T.; Zhang, J.; Chen, Z.Q. Pixel-level crack delineation in images with convolutional feature fusion. *Struct. Control. Health Monit.* **2019**, *26*, e2286. [\[CrossRef\]](#)
21. Li, S.; Zhao, X.; Zhou, G. Automatic pixel-level multiple damage detection of concrete structure using fullyconvolutional network. *Comput. Aided Civ. Infrastruct. Eng.* **2019**, *34*, 616–634. [\[CrossRef\]](#)
22. Mandal, V.; Uong, L.; Adu-Gyamfi, Y. Automated road crack detection using deep convolutional neural networks. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 5212–5215.
23. Wang, L. Research on Classification and Detection of Pavement Cracks Based on Machine Vision. Master’s Thesis, Harbin Institute of Technology, Harbin, China, 2019.
24. Li, B.; Wang, C.; Ding, X.Y.; Ju, H.J.; Guo, Z.P.; Li, Z.Y. Surface defect detection algorithm based on improved YOLOv4. *J. Beijing Univ. Aeronaut. Astronaut.* **2023**, *49*, 710–717. (In Chinese) [\[CrossRef\]](#)
25. Liu, Y.; Zhou, T.; Xu, J.; Hong, Y.; Pu, Q.; Wen, X. Rotating target detection method of concrete bridge crack based on YOLO v5. *Appl. Sci.* **2023**, *13*, 11118. [\[CrossRef\]](#)
26. Zhang, J.; Qian, S.; Tan, C. Automated bridge crack detection method based on lightweight vision models. *Complex Intell. Syst.* **2023**, *9*, 1639–1652. [\[CrossRef\]](#)
27. Wang, Y.; Wang, J.; Wang, C.; Wen, X.; Chen, Y.; Guo, Y.; Cao, R. MA-X-net: Mobile-attention X-network for crack detection. *Appl. Sci.* **2022**, *12*, 11240. [\[CrossRef\]](#)
28. Yuan, H.; Jin, T.; Ye, X. Modification and evaluation of attention-based deep neural network for structural crack detection. *Sensors* **2023**, *23*, 6295. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Zhu, L.; Li, D.B.; Yan, X.Z. Intelligent Tunnel Crack Detection Method Based on Improved Mask R-CNN Deep Learning Algorithm. *J. Graph.* **2023**, *44*, 177–183.
30. Zhang, M.X.; Xu, J.; Liu, X.P. Improved U-Net pavement crack detection method. *Comput. Eng. Appl.* **2023**, 10–11. Available online: <http://kns.cnki.net/kcms/detail/11.2127.TP.20231008.1724.026.html> (accessed on 6 April 2024). (In Chinese)
31. Qiao, W.; Zhang, H.; Zhu, F.; Wu, Q. A crack identification method for concrete structures using improved U-net convolutional neural networks. *Math. Probl. Eng.* **2021**, *2021*, 6654996. [\[CrossRef\]](#)
32. Yu, X.; Ta-Wen, K.; Shih-Pang, T.; Chen, Y.; Chen, S.; Wang, J.; Chen, T. EnRDeA U-net deep learning of semantic segmentation on intricate noise roads. *Entropy* **2023**, *25*, 1085. [\[CrossRef\]](#)
33. Wu, Q.; Song, Z.; Hu, C.; Lu, Y.; Zhou, L. A highway pavement crack identification method based on an improved U-net model. *Appl. Sci.* **2023**, *13*, 7227. [\[CrossRef\]](#)
34. Li, X.; Xu, X.; He, X.; Wei, X.; Yang, H. Intelligent crack detection method based on GM-ResNet. *Sensors* **2023**, *23*, 8369. [\[CrossRef\]](#)
35. Wang, J.; He, X.; Faming, S.; Lu, G.; Cong, H.; Jiang, Q. A Real-Time Bridge Crack Detection Method Based on an Improved Inception-ResNet-v2 Structure. *IEEE Access* **2021**, *9*, 93209–93223. [\[CrossRef\]](#)
36. Han, S.; Li, H.; Li, M.; Rose, T. A deep learning based method for the non-destructive measuring of rock strength through hammering sound. *Appl. Sci.* **2019**, *9*, 3484. [\[CrossRef\]](#)
37. Paramanandham, N.; Rajendiran, K.; Florence, G.P.; Yeshwant, S.P.; Sanjeeve, R.M.; Kumar, P. Pixel intensity resemblance measurement and deep learning based computer vision model for crack detection and analysis. *Sensors* **2023**, *23*, 2954. [\[CrossRef\]](#) [\[PubMed\]](#)
38. Yu, G.; Dong, J.; Wang, Y.; Zhou, X. RUC-net: A residual-unet-based convolutional neural network for pixel-level pavement crack segmentation. *Sensors* **2023**, *23*, 53. [\[CrossRef\]](#) [\[PubMed\]](#)
39. Lan, C.; Zhang, L.; Zhang, Y.; Fu, L.; Sun, C.; Han, Y.; Zhang, M. Attention mechanism combined with residual recurrent neural network for sound event detection and localization. *EURASIP J. Audio Speech Music. Process.* **2022**, *2022*, 29. [\[CrossRef\]](#)
40. Qiao, W.; Liu, Q.; Wu, X.; Ma, B.; Li, G. Automatic pixel-level pavement crack recognition using a deep feature aggregation segmentation network with a SCSE attention mechanism module. *Sensors* **2021**, *21*, 2902. [\[CrossRef\]](#) [\[PubMed\]](#)
41. Xu, S.; Deng, J.; Huang, Y.; Ling, L.; Han, T. Research on insulator defect detection based on an improved MobilenetV1-YOLOv4. *Entropy* **2022**, *24*, 1588. [\[CrossRef\]](#) [\[PubMed\]](#)
42. Yang, R.; Li, W.; Shang, X.; Zhu, D.; Man, X. KPE-YOLOv5: An improved small target detection algorithm based on YOLOv5. *Electronics* **2023**, *12*, 817. [\[CrossRef\]](#)
43. Man, K.; Liu, R.; Liu, X.; Song, Z.; Liu, Z.; Cao, Z.; Wu, L. Water leakage and crack identification in tunnels based on transfer-learning and convolutional neural networks. *Water* **2022**, *14*, 1462. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.