applied sciences

MDPI

Article

An End-To-End Model for Pipe Crack Three-Dimensional Visualization Based on a Cascade Neural Network

Xia Fang [1], Yang Wang [1], Yong Li [1], Jie Wang [1,*] and Libin Zhou [2]

[1] School of Mechanical Engineering, Sichuan University, Chengdu 610065, China; 18215575946@163.com (X.F.); 18855523729@163.com (Y.W.); liyong123lk@163.com (Y.L.)

[2] School of Computer, Data & Information Sciences, College of Letters & Science, University of Wisconsin Madison, Madison, WI 53706, USA; lzhou228@wisc.edu

* Correspondence: wangjie@scu.edu.cn; Tel.: +86-138-0801-5321

Received: 20 January 2020; Accepted: 10 February 2020; Published: 14 February 2020

Abstract: With the continuous progress of machine vision technology, crack detection in pipelines has been greatly improved. For crack detection in deep holes, inner tubes, and other environments, it is not only necessary to detect the existence of cracks, but also to collect important information regarding the crack detection direction for further analysis. Because shooting with a frontal field of view causes the real side wall images to produce certain distortions, the detection and calibration of cracks requires a certain amount of professional technology and time. It usually takes a long time to collect the image to eliminate the distortion, and then to identify the crack and mark the direction according to the data line. Therefore, a simple and efficient end-to-end neural network model for crack recognition and three-dimensional visualization are proposed by using a cascade network and simple recognition technology in conjunction with inertial navigation equipment. In addition, we screen the crack data via pixel calibration and eliminate the ambiguous data to make the visualization more accurate. Experiments in pipelines and burrows show that the accuracy, performance, and efficiency of the proposed method reached a high level.

Keywords: crack detection; cascading neural networks; distortion correction; three-dimensional visualization; end-to-end model

1. Introduction

With the development of machine vision technology, the detection of cracks is applied to many places, such as pipeline damage [1]. Nonlinear models play an increasingly important role in detecting the direction of stress cracks in microhole and crack distributions in cylinders and pipes. This method is also applicable to the analysis of subsurface stress distributions. Because of the relationship between cost and precision, deep ground microdrilling is often used to collect internal rock fractures, and underground drilling is treated with water injection fracturing. Therefore, it is necessary to conduct all-round explorations and crack calibrations of the internal environment [2]. At present, most of the hole wall crack detection methods only detect whether the crack exists, but cannot quantify the extent and location of the crack. In addition, the traditional method relies heavily on the experience and knowledge of the algorithm designer, which leads to certain technical requirements for the detection staff, and hence, a low detection accuracy [3–6]. Some detection systems focus on the speed of image recognition [7,8], while others focus on accuracy [9,10]. Other algorithms analyze the direction and composition of cracks based on the information of the front and rear frames, and cannot implement end-to-end detection [11]. Based on the nonlinear mapping capability of deep convolution networks, there is a series of methods that can be used to eliminate image distortion using the neural network [12].

Due to the limitations of the shooting environment, most of the deep ground shooting adopts a cone-shaped reflection. Therefore, the image itself has a large amount of distortion [13]. Considering that the field of view of refraction shooting is too small, it takes too long to build the whole model. Therefore, some people use the direct shooting method, which involves distortion. Original pixels at different positions may map to one pixel and one pixel may spread to multiple pixels [14]. Usually, the inner hole is shot as video information; frames are extracted and splicing is carried out at a later stage. Therefore, the problem of splicing is involved in the crack identification, and how to identify the crack after splicing is also a problem. It is also necessary to identify the direction of the crack, which requires a lot of labor and time [15]. Deep ground micropores and other similar types of crack detection and quantitative analysis are not a single task. Therefore, if multiple systems or models are used to solve the problem while professional staff conduct calibration, the cost and time taken for detection will be greatly increased.

The identification of cracks is a classification problem; therefore, the label accuracy of data itself is very important to the learning of a neural network. Some of the models can perform semantic analysis on the images and have a lower time cost due to using texture regression analysis to measure the accuracy of crack identification [16]. However, such an operation is complicated and has a lot of uncertainties. It also needs a lot of manual identification of the video at a later stage.

Based on the above reasons, we proposed an end-to-end model based on a cascade neural network. It simultaneously deals with surface distortion and patch-search-based crack recognition, and coordinates with the inertial measurement unit (IMU) to calibrate the crack direction of side-wall images. Moreover, the ground truth is used to guide the data set, filter the ambiguous data, and enhance the accuracy of the system. The first goal of this paper was to build a mapping relationship to achieve a high robustness. Our system needed to be enhanced under the influence of light, shadow, blur, and other noises, and has a wide range of adaptability. The second step was to construct a patch-based crack recognition and to combine the inertial measurement unit and meter information to visualize the three-dimensional crack distribution. At the same time, it combined the ratio relation between the patch and the frame to determine the extent of the crack.

After the distorted image was traversed by the patch, the coordinates of each patch in each side-wall image were obtained by combining inertial navigation information. The patch in the splicing was used to obtain the crack coordinates and crack graphics; the ground truth was established through the original image, and the model system was evaluated using intersection-over-union (IOU). The three-dimensional image and coordinate coefficients of the crack were obtained using continuous splicing. The convolution neural network adjusted the model to obtain the most accurate model system for the task.

Several experiments showed that the system could objectively and accurately identify the crack. This greatly improves the efficiency of related work, such as deep crack detection, steel tube inner wall detection, groove inner wall detection, and other aspects with good results. It saves labor costs and time.

2. Related Works and Foundations

Because the environments of most deep ground water pressure crack stress detection methods are very narrow and the camera field of view is also very narrow, there is an urgent need for the fast processing of small field images, as well as a detection method. Our network structure is divided into two parts: the front part is the mapping reconstruction network that is used to solve the distortion; and the back part is the detection network for the identification and crack splicing. Using information from LPMS-ME1 IMU (Alubi, Guangzhou, China), we spliced the coordinates of the pipe cracks with corresponding frames in the time and spatial dimensions. We spliced them with patches and finally formed a 3D system for flaw detection and crack trajectory generation. In order not to be disturbed by the magnetic field, we shielded the magnetic field information in the Kalman filter [17]. Because crack

detection itself is a classification problem, after traversing using a patch mechanism, we evaluated its quality with a specific ground truth and finally got the system to accurately identify the crack trend.

To ensure the speed of shooting and avoiding overlapping when obtaining the field of view of each frame, we adopted the method of direct shooting. With a fixed-speed propulsion device, we could quickly and accurately obtain the hole-wall information of the field of view of the front view. Based on the shooting principle of a CCD HD-2M camera (Imperx, Boca Raton, FL, USA), the corresponding sizes of pixels at different positions in the field of view were deviated. The exposure curve will also be distorted due to the different distances from the light source. Using the nonlinear fitting features of the neural network and our homemade feature stickers, we trained a visualization network to achieve radial distortion correction of the in-hole wall-facing image to become a plane image. In order to shoot effectively, we made a corresponding hardware system. With a large amount of calibrated data, the training model had a higher objectivity than a single subjective discrimination. To ensure that the probe rod entered the probe site at a constant speed, a lifting and limiting mechanism was designed. Figure 1 shows the overall and partial introduction of the hardware of our entire system. To consider the actual size relation of each pixel in the image and prevent large errors in the process of the distortion correction, we used a constant-speed propulsion device to propel the whole system. In order to keep the whole system stable, we designed two kinds of stabilizing devices, as shown in Figure 1. An H1S2T motor (NiMotion, Beijing, China) was used to push the screw and drive the lifting platform to feed the test rod evenly. A 485 transmission protocol was adopted for coordinate and visual signal modulation of this system, which was conducive to long-distance transmission. Figure 2 shows a schematic diagram of the actual operation of the system.

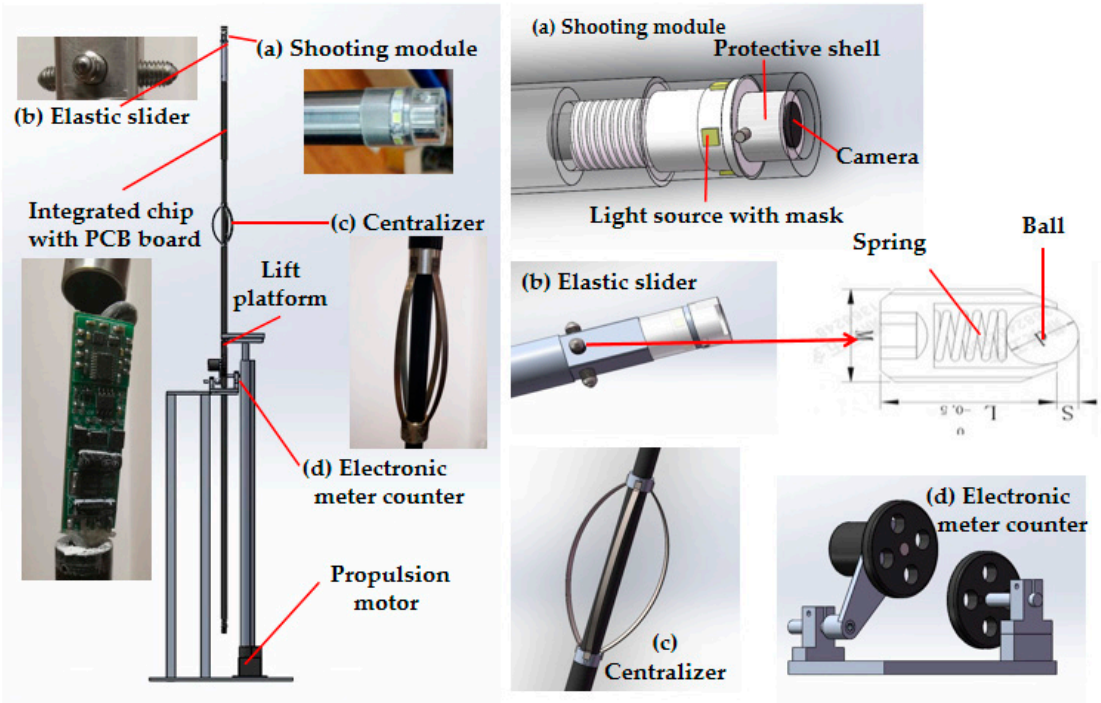


Figure 1. System partial function introduction.

At present, the crack detection in the inner wall of a pipeline is mainly done through manual data analysis, and it often takes a long time to take pictures inside a pipeline. The influence of different cracks is also different. In the previous article, we mentioned some mainstream crack detection methods, but the detected cracks are not further processed and their morphological characteristics are not separated. If the crack distribution in the hole wall can be obtained succinctly and quickly, the analysis of internal holes will be very convenient. The purpose of this study was to detect the distribution of stress cracks

due to water fracturing on the inner wall of a microbore and the distribution trend of stress in deep ground at a low cost.

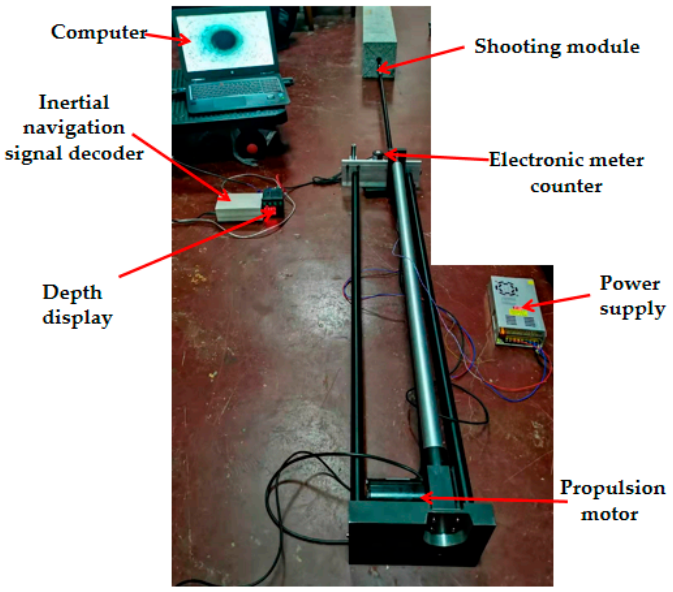
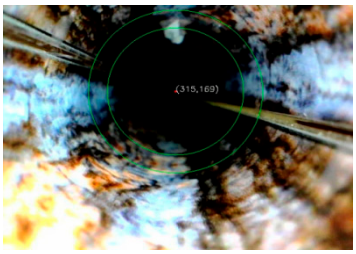


Figure 2. Image of the working system.

Based on the background of the appeal, therefore, a kind of intelligent and convenient small hollow interior wall crack detection scheme that can effectively reduce the workload of operators and the quantitative basis for further analyzing the shooting data is urgently needed. Our system was divided into two parts. One part was responsible for the image of the front view distortion correction and sampling. The other part is responsible for the identification of the crack and to reassemble them into continuous crack images. Through correlation analysis of patch images and ground-truth-based data guidance, a three-dimensional crack coordinate map was established from IMU information.

In part one, to ensure that the up-sampling in the distortion correction process could accurately restore the detailed information of the hole wall in the elevation diagram, we used the depth concatenation mechanism [18,19]. In the second part, we used the confidence analysis of the patch and the data guidance mechanism of the ground truth to quantitatively analyze the direction and confidence level of the crack. Figure 3 shows the schematic diagram of the image reconstruction of the rock pattern wallpaper on the inner wall of the microsplit cavity.



(**a**)

Figure 3. *Cont.*

(**b**)

(**c**)

(**d**)

(**e**)

Figure 3. (**a**) Image of a fixed focus segment (in green circle) was captured from the elevation view, (**b**) corresponding distortion correction diagram in the focal segment, and (**c**,**d**) splicing using the coordinate value, and (**e**) corresponding three-dimensional visualization.

The system was composed of a motor, meter counter, IMU, miniature CCD camera, ring light source, and alignment device. Here, we used the framework of OpenCV 4.1 and TensorFlow 1.3. The inertial navigation instrument and video signal were collected using the single-chip microcomputer, and the final output was produced through the Python interface. The software system was programmed in Python 3.6. The detection algorithm was developed in the OpenCV and TensorFlow 1.5 deep learning platform.

3. Proposed Method

The whole algorithm was divided into two parts and three processing processes. Figure 4 provides an overview of the processing workflow, which displays the components of our proposed vision system. Our system was roughly divided into two parts: a distortion correction network and a crack detection network. In the distortion correction network branch, the distortion caused by the expansion of the frontal annular interface to the rectangular image is mapped by using the nonlinear fitting ability

between convolution layers of the neural network. In the crack detection network branch, the distorted images were sampled up. Moreover, a simple crack detection network was used to traverse the image, and the patch with the detected crack was passed to the position of the original image to finally form the overall crack map. Combined with the information from the IMU and meter, a 3D crack map was formed. The image we captured was between 3 and 5 cm in the focal section of the front lens, and when we extracted the image, we used a gamma correction to correct the difference in brightness due to different shooting distances.

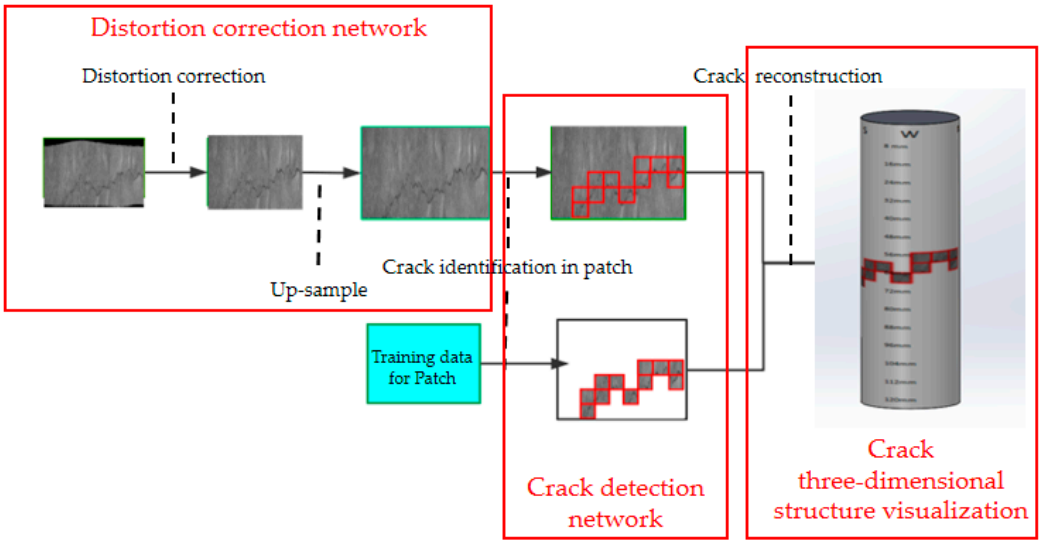


Figure 4. Flowchart of the entire method.

Considering that this system is not a nonlinear fitting of a single task, we created a special processing method for the training data. Real stickers were used in combination with geotechnical internal photographs, and data enhancement was used in the learning process. The original signal we obtained was the video signal because the radial feed we shot was controlled by the motor. Therefore, we only needed to adjust the feed speed and the corresponding length in the extracted frame to obtain relatively stable and non-repetitive area pictures.

We provide the theoretical background for the model in Sections 3.1–3.3. We briefly describe the patch recognition process and auxiliary method of data calibration in our work in Section 3.4.1. Finally, the details of our experiments are given in Section 3.4.2.

3.1. Distortion Correction Network

We grayscaled all image patches used for training. The picture directly taken by the camera was a three-channel color picture. It contained the information of an RGB (red, green, and blue) three-channel, while the gray image we want only needed to retain the brightness information of the image; therefore, the color information was discarded. For the bridge crack images to be processed in this paper, the brightness information of grayscale images could fully express the overall and local features of the image; furthermore, the grayscale processing of the image greatly reduced the amount of calculation for subsequent work.

$$Gray = R \times 0.299 + G \times 0.587 + B \times 0.114 \quad (1)$$

Since the distortion correction itself is a reconstruction process, we constructed a full convolution neural network for this task. Because we needed to maintain the details in the whole image while reconstructing the image, we used the method of cascading detail layers to fuse the low-dimensional detail features with the high-dimensional features [20]. As shown in the figure, after weighing the operation speed and reconstruction accuracy, we adopted the following network structure as the

distortion correction terminal. The receptive field of the neural network should be large enough to cover the entire field of view (FOV), and the network should be deep enough.

The input image size of our system was 408 × 568. All the network structures in this system were determined through the tradeoff between accuracy and speed. The first layer was a convolution layer that filtered the input with 64 kernels and each filter was of size 7 × 7 × 1. The output image, which was as large as the input image, could be obtained using the transposed convolution decoding region [21]. The structure of the network is shown in Figure 5. The method of skipping connections could be simply realized with backpropagation without gradient vanishing. Although models with a higher resolution may lead to higher simulation accuracy, a longer computation time is required. Table 1 shows the detailed structure of the distortion correction network.

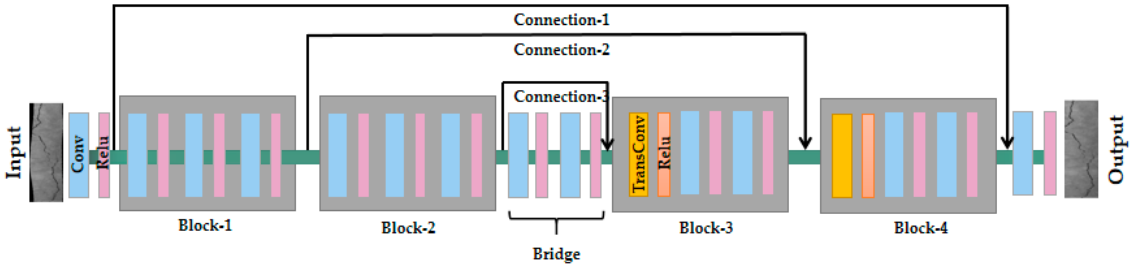


Figure 5. Diagram of the distortion correction network. Blue(Conv) represents the convolution layer, yellow(TransConv) represents the transposed convolution layer, and red and orange(Relu) represent the corresponding activation functions.

Table 1. The detailed structure of the distortion correction network. Tag * means that the convolution information there was extracted, then directly superimposed with the information of the symmetric transposed convolution channel, and finally fused by convolution.

Layers (Size, Kernel Size, Stride)	Step	Type	Input	Output
ImageInputLayer (408 × 568)	-		1	64
Conv_ Relu (7 × 7,1)	-	Pad (same)	64	64
Max_Pooling	Block_1	Pad (0)	64	64
* Conv_ Relu_Block_1 (5 × 5,1)	Block_(1–2)	Pad (same)	64–128	64–128
Conv_ Relu_Block_1 (5 × 5,1)	Block_(1–2)	Pad (same)	64–128	64–128
Conv_ Relu_Block_1 (5 × 5,1)	Block_(1–2)	Pad (same)	64–128	64–128
Drop_Out (204 × 284)	Block_2	50%	128	128
Max_Pooling (102 × 142)	Block_2	Pad (0)	256	256
Conv_ Relu_Bridge (5 × 5,1)	Bridge	Pad (same)	256	256
Conv_ Relu_Bridge (5 × 5,1)	Bridge	Pad (same)	256	256
Drop_Out	Bridge	50%	256	256
* TransConv_ UpRelu(2 × 2,2)	Block_(3–4)	Crop (0,0)	128–64	128–64
Conv_ Relu_Bridge (5 × 5,1)	Block_(3–4)	Pad (same)	128–64	128–64
Conv_ Relu_Bridge (5 × 5,1)	Block_(3–4)	Pad (same)	128–64	128–64
Conv_ Relu (3 × 3,1)	-	Pad (same)	64	1
Final_Reg_Out	-		1	Pixel-Pr

Based on the relationship between the pixels, we defined a set of loss functions. Here, we used two groups of loss functions to represent the learning effect of the distortion correction network, and these loss functions were as follows.

The reconstruction loss function was calculated using:

$$L_{pixel} = \frac{1}{hw}\sum_{i}^{h}\sum_{j}^{w}\rho(I_1(i,j) - I_2(i + V_{i,j}^{x}, j + V_{i,j}^{y})), \quad (2)$$

where i, j represent the values in the horizontal and vertical coordinates of the pixels in the time spectrum diagram I, respectively, V^x and V^y are the estimated pixels in the horizontal and vertical

directions. To reduce the influence of outliers, we used the Charbonnier penalty $\rho(x) = \left(x^2 + \epsilon^2\right)^{\alpha}$. h and ω are the height and width of the images (I_1 and I_2), respectively.

Because most of the distortion will cause a non-closed interval, we used the smoothness loss to manage the aperture problem that caused ambiguity in estimating details of the distortion in non-textured regions. It was calculated using:

$$L_{smooth} = \rho(\nabla V_x^x) + \rho\left(\nabla V_y^x\right) + \rho\left(\nabla V_x^y\right) + \rho\left(\nabla V_y^y\right), \tag{3}$$

where ∇V_x^x and ∇V_y^x are the gradients of the estimated V^x in each direction, and ∇V_x^y and ∇V_y^y are the same as V^y.

By comparing the characteristic graphs I_{1n} and $I\prime_{1n}$ before and after the partition reconstruction, we can know the degree of correction within the network.

Finally, we combined several loss functions to form an objective function, as shown in Equation (4):

$$L_S = L_{pixel} + L_{smooth} \tag{4}$$

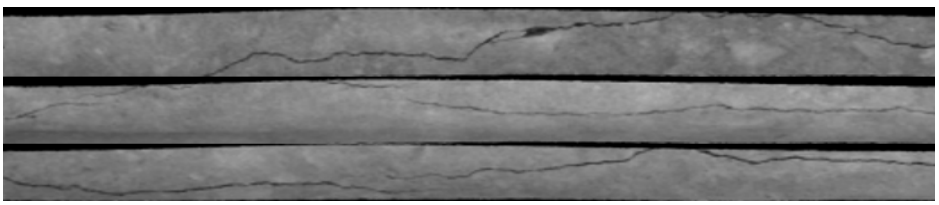
In order to evaluate the distortion performance of the system, we used the image similarity coefficient as the reference index. For this similarity evaluation, we used a comparison parameter to evaluate the reconstruction quality. *SSIM* represents the structural similarity between the target and the predicted image [22], as follows:

$$SSIM\left(I_{p1}, I_{p2}\right) = \frac{\left(2\mu_{p1}\mu_{p2} + c_1\right)\left(2\sigma_{p1p2} + c_2\right)}{\left(\mu_{p1}^2 + \mu_{p2}^2 + c_1\right)\left(\sigma_{p1}^2 + \sigma_{p2}^2 + c_2\right)}. \tag{5}$$

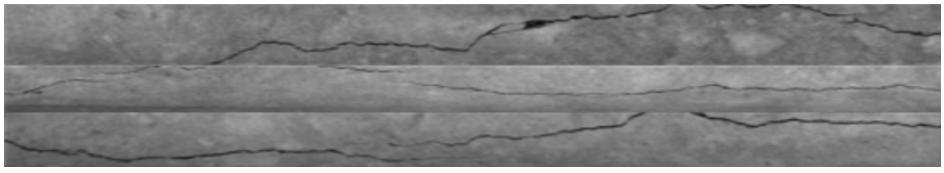
To calculate the efficiency, we split and compared the whole graph, and finally obtained the similarity coefficients, where μ_{p1} and μ_{p2} are the mean values of the input images. σ_{p1} and σ_{p2} are the variance of the images, and σ_{p1p2} is the covariance of these inputs. c_1 and c_1 are constants used to stabilize the division by a small denominator, which were 0.0001 and 0.001, respectively.

$$K_{ssim} = \frac{1}{N}\sum_{n}^{N}\left(1 - SSIM(I_{1n}, I'_{1n})\right) \tag{6}$$

Figure 6 shows the schematic diagram before and after the distortion correction.



(**a**)



(**b**)

Figure 6. (**a**) Expanded image of the circular elevation. (**b**) The corresponding distortion correction image.

After reconstruction, it was necessary to conduct crack detection and the patch traversal on the image. Therefore, it was necessary to conduct up-sampling on the distorted image. For example, as Rong et al. [23] did, we processed the image. The schematic diagram of the up-sampling network is shown in Figure 7. The detailed structure of the distortion correction network is shown in Table 2. Because the lower sampling was different from the upper sampling, the lost information was abstract and high-dimensional. As a result, we adopted the method of bicubic down-sampling to conduct lower sampling on the collected original large images to make labels. Using the learned up-sampling filters effectively suppressed reconstruction artifacts caused by the bicubic interpolation. The crack characteristics in different scales could be learned by comparing the label values after each sampling rate reduction. Since the L2 loss function made the image smooth and lost many features of details when learning the mapping of super pixels, we used the Charbonnier function to learn the mapping, which is defined as:

$$\pounds(\overline{I'_{1n}}, I'_{1n}, \theta) = \frac{1}{N}\sum_{i=1}^{N}\sum_{S=1}^{L}\rho(\overline{I'^{(i)}_{1nS}} - I'^{(i)}_{1nS}), \tag{7}$$

where $\overline{I'_{1n}}$ represents the input frame, I'_{1n} represents the up-sampling frame, and θ represents the nonlinear function relationship between the two frames. $\rho(x) = \sqrt{x^2 + \epsilon^2}$ is the Charbonnier penalty function. N is the number of training samples in each batch, and L is the number of levels in our pyramid (L = 3, in our paper). We empirically set ε to 1×10^{-3}.

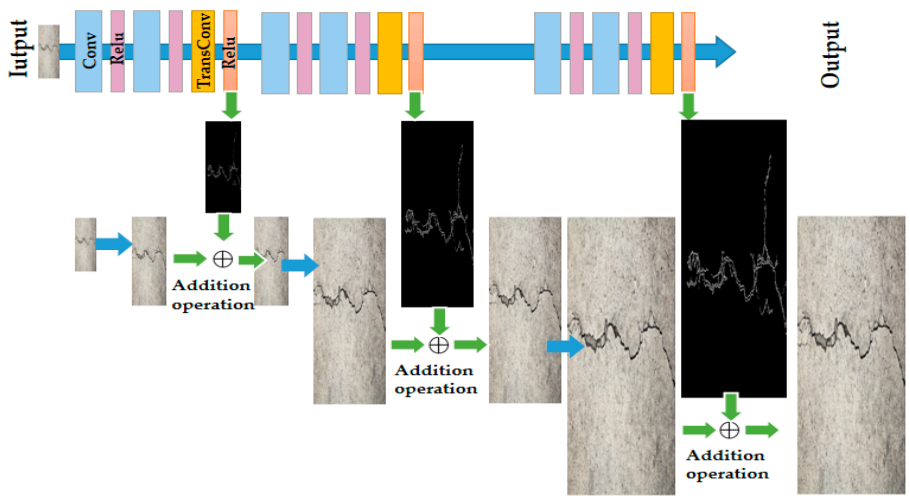


Figure 7. Schematic diagram of the up-sampling network.

Table 2. The detailed structure of the distortion correction network.

Layers (Size, Kernel Size, Stride)	Step	Type	Input	Output
ImageInputLayer (408 × 568)	-		1	64
Conv_ Relu (3 × 3,1)	(1–3)	Pad (same)	64	64
Conv_ Relu (3 × 3,1)	(1–3)	Pad (same)	64	64
Drop_Out	(1–3)	50%	64	64
* TransConv_ UpRelu (2 × 2,2)	(1–3)	Crop (0,0)	64	64
Final_Conv_Reg_Out (3264 × 4544)	-		1	Pixel-Pr

Tag * means that the characteristics of this layer were compared with the corresponding label layer by layer, and the deviation value was returned.

We used the method of Lai et al. [24] to up-sample the distorted image and used the loss function. The output of each transposed convolution layer was connected to two different layers:

(1) Convolution layer 1, which was used to reconstruct residual images at this level;

(2) Convolution layer 2, which was used to extract features at the finer s+1 level.

3.2. Crack Detection Network

We used the method of scanning the cracks and finally spliced the images [25]. Because the processed image was very large, we used a shallow classification network to identify the distorted image. To evaluate the stability of the system, we transformed the classification problem into a regression problem by using the ground truth labels. To effectively evaluate the relationship with the real crack distribution, we did some processing on the pixel of the label boundary. Figure 8 shows the network structure for identifying the cracks in a patch, which was simple and fast. The details of the structure of the crack detection neural network is shown in Table 3. In this article, the image resolution of 3264 × 4544 was collected and it was necessary to directly manage such a large image and its associated large computational burden. A picture that is too small, however, could not identify cracks or the overall characteristics of the reaction. After several experiments, we selected 64 × 64 pixels per image cut into small images. This size could guarantee the accurate identification of the fracture and capture the overall layout of the cracks.

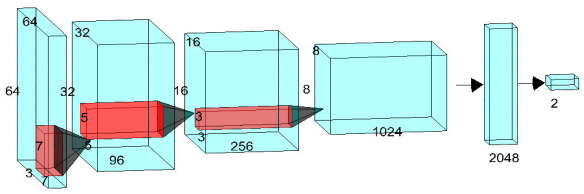


Figure 8. Schematic diagram of the crack detection neural network.

Table 3. The details of the structure of the crack detection neural network.

Layers (Kernel Size, Stride)	Type	Input	Output
ImageInputLayer (64 × 64)	-	-	1
Conv_ Relu (7 × 7,1)	Pad (same)	1	96
Conv_ Relu (5 × 5,1)	Pad (same)	96	256
Conv_ Relu (3 × 3,1)	Pad (same)	256	1024
Conv_ Relu (3 × 3,1)	Pad (same)	1024	2048
Fully Connection	-	2048	2
Drop_Out	50%		
Final_Class_Out		2	2

By combining the inertial navigation information, we obtained the coordinates of each patch, combined the real-time coordinates with the propulsion distance, and referred them to the initial coordinates. We could also obtain the real-time coordinates of each crack in the video, and constructed the 3D image. This could also easily and efficiently construct the defect environment inside the detected object, forming an intuitive detection system.

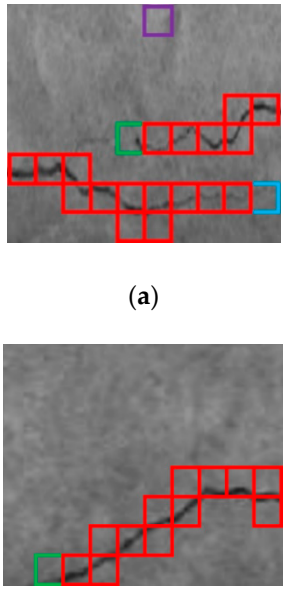
3.3. Three-Dimensional Visualization of the Crack Structure

Because the required result in this paper was not a simple classification problem, and we used patch traversal to map the image of the same height, we needed a mechanism to evaluate the confidence of the patch to reduce the system noise. We determined whether the recognition result in the patch adjacent to it was the wrong judgment of the neural network through the discriminant result in a patch and the position relation. Based on the identification of the presence of cracks in a patch, whether two cracks were adjacent determined the recognition confidence. If there were cracks in the three positions to the left of a crack, while there were none on the right, the crack was at the clockwise rotating end. The discriminant method was also applicable to the identification of the right terminal crack. If the

patch was an isolated patch, it was considered to be an interference or an excessively short crack, and the following decision rules were obtained by analogy:

(1) If there were zero identical cracks in the eight patches around the current patch, the cracks at the current patch location were invalid and the results were suppressed.
(2) If there was an identical judgment in the current position, the crack in the current position was valid and the patch was at the reverse end of the crack.
(3) If there were cracks all around the patch, the crack at the current location was valid and was located on a section of the crack in the middle.
(4) If there were three or four identical judgments at the current position, the crack in the current position was valid and was located at the intersection position.

Figure 9 shows the identification of cracks in our system.



(**a**)

(**b**)

Figure 9. (**a**) Two separate cracks and (**b**) a continuous crack. The red box was the normal crack patch, and green and blue represent the left end and right end, respectively. Purple was an isolated crack patch; therefore, it was excluded.

The system structure of this section is shown below. After the recognition achieved in the above manner, we combined the information of the inertial navigation and electronic meter with the corresponding image to obtain the expansion diagram of the radial extension. The simplified version of the crack detection network was composed of the following four parts:

1. Identification of crack: By using a 64×64 sliding window, the convolution neural network could identify the crack.
2. The patch map: The patch identified as a crack was stored in a blank page. It was then combined with other images in the patch identified through traversal to obtain the image of the whole crack, which was in the same image frame.
3. Coordinate calculation: With the information from the meter counter and the IMU, it recorded the corresponding position of the extracted patch and pasted the patch to the corresponding position of the original picture.
4. The output direction: According to the relationship between the relative position of the patch and the adjacent frames, the spatial direction of the inner wall crack could be obtained.

Figure 10 shows the expanded image extracted from different frames after the ring distortion. The recognition result diagram was based on a patch of the crack after the loss of distortion and the 3D image visualized by combining the IMU signal and depth data of the meter.

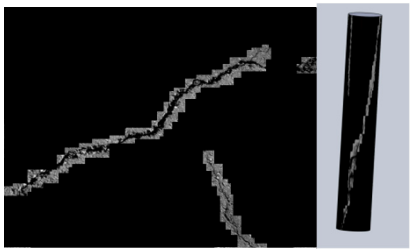


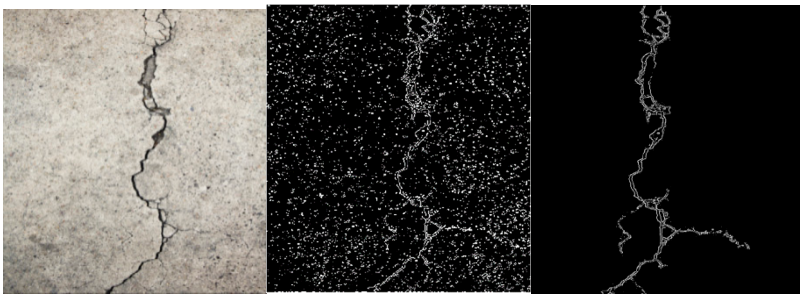
Figure 10. Three-dimensional visualization of cracks corresponding to spatial coordinates.

Previous data were manually labeled to identify the existence of cracks and were then described and observed. Through the above mechanism of crack confidence, we could get a more accurate crack distribution structure. Through the above spatial visualization mapping operation, the end-to-end module of the crack direction identification was completed. Because the system directly outputted the spatial information of the crack, it greatly reduced the workload of operators and analysts and the technical difficulty for testers.

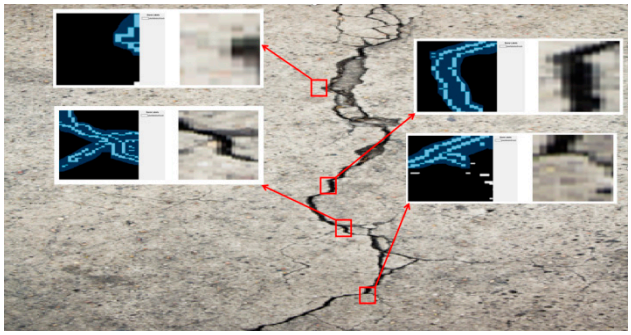
3.4. Experiments Details

3.4.1. The Dataset

Because a deep learning data set is quite important, when calibration was carried out on the collection of data, we needed to accurately measure and emphasize the singular data and ambiguous data. The corresponding size in the image we used was 0.1 mm for the actual width of each pixel. In the actual observation, the hole-wall crack was not observed if the width was less than 2 mm, and the pixel width needed to be 20 pixels. At the same time, in the actual ground cracks, there was relatively complete structural information in a crack that was 0.2-cm long. Therefore, the crack pixel length in the cut image block was 64 pixels long. The system's light would not change; therefore, the gamma error of the light was the same and it was easy to coordinate. Considering the actual situation of long and wide cracks, the patch size was set as 64×64 pixels. As for whether there was a crack in the divided patch and the marking judgment was carried out, the collected patch was binarized with a dynamic threshold value. Through adaptive local binarization processing (LPB) [26], the processing of the pixel depth in the image was segmented to obtain the high-gray value crack. The connected domain operation was used to extract the crack information. In order to ensure the confidence of the crack image data in the patch, we proposed the use of a ground truth calibration method using a patch for guidance. When more than 20% of the pixels in the ground truth marker patch were read, the image in the patch was learned anyway. If it was less than 20% and classified as ambiguous data image, it will not be learned. As shown in Figure 11, this was the method used to process the contour image and mark the patch with the local binarization method. Through the pre-screening of the data in the above IOU, we transformed what was originally a classification problem into a regression problem without reducing the efficiency and speed of learning. As for the final trained model, we also evaluated it by comparing the image in the patch with the ground truth.

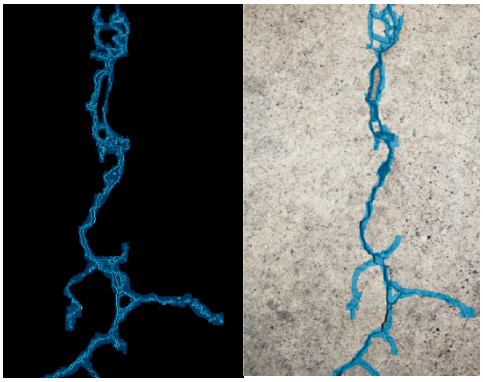


(**a**) (**b**) (**c**)



(**d**)

(**e**) (**f**)



(**g**)

Figure 11. (**a**) Expansion diagram of the inner wall of the pipe. (**b**) The effect diagram of the mean value binarization. (**c**) The effect diagram of local binarization (range 5 × 5). (**d**) The schematic diagram of data label is made according to the ground truth in the patch. (**e**) The ground truth accounted for more than 20% of all pixels in the patch, which was denoted as a crack patch. (**f**) The ground truth and all pixels in the patch accounted for less than 20%, this was recorded as suspected data and not sent to the network for learning. (**g**) Ground truth mapping for the whole image.

According to the label mechanism assisted by the IOU, we could get a good performance of connect domain area segmentation of the crack data and produced the effect diagram. Through this method, we labelled the data used to identify the crack in the patch. To make up for the shortage of measurement data, we searched many cracked image data from the Internet and adjusted the corresponding size. We selected pictures with long continuous cracks in the whole inner wall and adjusted the size of all test pictures to 3264 × 4544. We used 6000 images with cracks for distortion learning and randomly divided them into 258 patches. In addition, cracks were detected with 64 × 64 patch sizes from the image data.

In order to synthesize and enhance our data as authentically as possible, we enhanced the data learned in the recognition network and selected random parameters to process the image. We rotated the images in the patch in equal proportion such that the learning speed would be slower than the reasoning speed. At the same time, it was flipped horizontally and upside down to increase the number of images, to make the cracks clearer, and to improve the accuracy of the training model. Figures 12 and 13 show how we augmented the data.

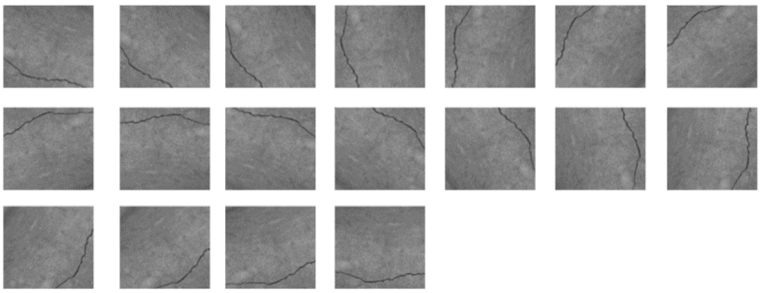
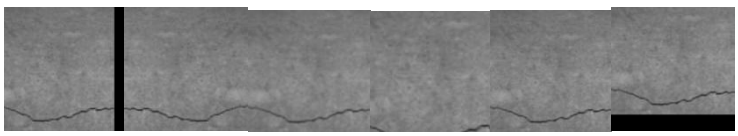
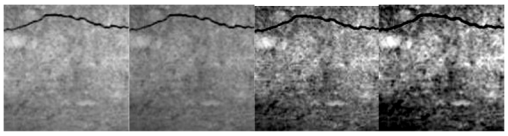


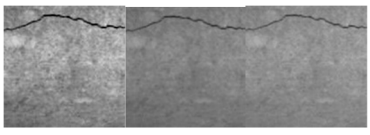
Figure 12. After the same proportional scaling, the 360-degree rotation in the same patch is augmented.



(**a**)



(**b**)



(**c**)

Figure 13. (**a**) Images with different jittering parameters, showing that the range was 10% of the whole picture. (**b**) The state of the picture after adding Gaussian noise at different intensities (10%, 20%, 30%, and 40%). (**c**) Different gamma curves.

In this paper, our dataset was collected using the device shown in Figure 12 and we evaluated our method on this dataset.

We used the original network trained under a large amount of crack data and used a fine-tuning method for the network by shooting and calibrating the crack data ourselves.

3.4.2. The Implementation Process

Our model was trained on four NVIDIA GTX2080 GPUs with 16 GB memory and a PC with a 3.4 GHz CPU for roughly 10 h. Experiments were implemented based on the deep learning framework TensorFlow. The operating system was Windows 10.

The first part was the nonlinear mapping of the full convolution network; therefore, the selection of the size of the convolution kernel will have a great impact on the whole system. The patch size in the recognition end of the second part was 64; therefore, we directly used the convolution kernel size of 3×3 for fitting. To learn how the size of kernels in the first convolution layer affected the performance of the network, we trained and tested the network with different sizes of kernels and fixed the rest of the parameters. Table 4 shows the influence of the size of the kernels. It can be noted that the tested kernel sizes within the set {3, 5, 7, 9, 11} showed different performances, while the kernel sizes beyond that range resulted in a higher mean squared error.

Table 4. The network of different steps in this system was affected by the size of the convolution kernel.

Kernel Size	Accuracy (%)	*SSIM*
Distortion correction Network (7×7, 9×9)	-	0.713
Distortion correction Network (11×11)	-	0.833
Distortion correction Network (Ours)	-	0.892
Up-sampling network (7×7)	-	0.757
Up-sampling network (9×9)	-	0.786
Up-sampling network (Ours)	-	0.824
Crack detection network (3×3)	93.1	-
Crack detection network (5×5)	96.1	-
Crack detection network (Ours)	98.3	-

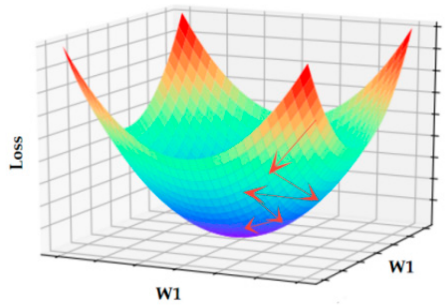
In order to prevent excessive detail loss in the pooling process and to fuse the features of high and low dimensions to a certain extent, we used the depth concatenation mechanism to learn the distorted images. Skipping one or more layers could accelerate the training speed and prevent gradient vanishing. The comparison of reconstruction errors with and without a depth concatenation is shown in Table 5. We can see that reconstruction took less time to converge and the final error was smaller with depth concatenation.

Table 5. The network of different steps in this system was affected by the size of the convolution kernel. PSNR—Peak Signal to Noise Ratio; IFC—Information Fidelity Criterion.

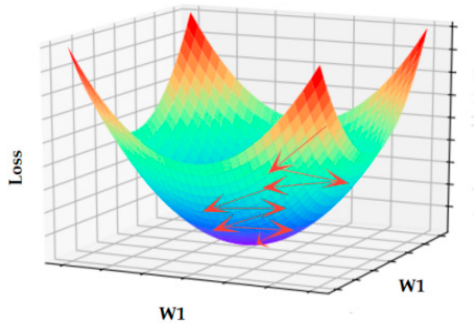
Method	PSNR	*SSIM*	IFC
Distortion correction network (without depth Concatenation)	34.1	0.813	2.312
Distortion correction network (depth concatenation)	37.5	0.892	3.145
Up-sampling network (without multiscale label)	44	0.742	1.781
Up-sampling network (multiscale label)	48.2	0.863	2.124

Our system consisted of two parts. One was the regression part of the distortion correction and up-sampling, and the other was the patch crack recognition part guided by the ground truth. We set up a loss function based on mean squared error (MSE) for pixel correction to show the change of the corresponding relationship between pixels before and after the distortion included up-sampling. For the classified structure, in each patch, we simply used an L2 function to identify the crack. Furthermore, during the training process, the stochastic gradient descent (SGD, momentum = 0.9) with mini-batches of 64 samples was applied to update the weight parameters. When we trained one step, we froze the weight of the other branch. Until the performance was stable, we carried out the whole weight

training. The distortion correction and up-sampling were complementary, as their fusion significantly improved both. As He et al. [27] proposes, the learning rate based on cosine attenuation should be used for image learning with high detail requirements. The learning rate was initially set to 10^{-2}, and then decreased according to a fixed schedule. The rate was changed to 10^{-3} after 40k iterations, then to 10^{-4} after 60k iterations, and stopped after 80k iterations. Figure 14 shows the convergence under the state of variable learning rate, which could effectively avoid the zigzag phenomenon in the process of regression [28]. Table 6 shows the results of several steps in our system using different learning rates and optimizers.



(**a**)



(**b**)

Figure 14. Because distortion mapping and up-sampling are processes that go from coarse to fine, the varying learning rate had a better convergence effect on our system. (**a**) The learning rate of cosine decay is used. (**b**) Constant learning rate.

Table 6. The effects of different training parameters.

Method	Accuracy (%)
Constant step size learning rate	92.1
Schedules vary the learning rate	92.8
Cosine changes the learning rate	94.2
Stochastic gradient descent optimizer	94.2
Adams optimizer	93.5

4. Results

In our system, a method of distortion correction and post-correction image up-sampling of the visual image with the crack detection in the patch was integrated. The detection data was guided by the ground truth, and the patch was spliced using the confidence mechanism to improve the robustness and efficiency of the system. Finally, combined with the IMU and meter data in the system,

the three-dimensional shape of the internal crack was visualized. To verify the effectiveness of the distortion correction, we used the *SSIM* index to evaluate the crack images from different perspectives after the distortion correction and used a confusion matrix to evaluate the crack recognition. To evaluate the performance of the crack detection, we validated our module on a dataset and achieved a final detection accuracy of 96.3%. Figure 15 shows the effect after 3D visualization.

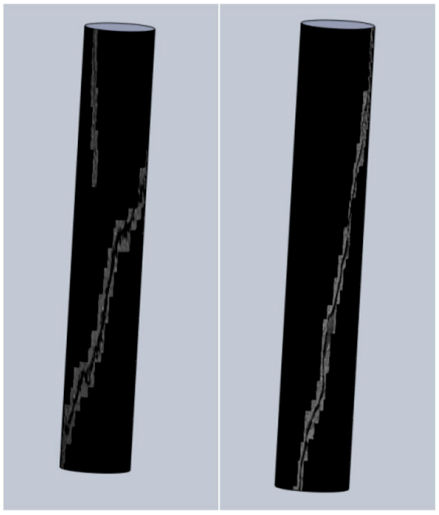


Figure 15. The resultant renderings.

For the crack identification part, we constructed an evaluation system to evaluate the robustness of the system. According to whether the extracted area in the patch contained the crack characteristics of corresponding labels, the accuracy of this model was constituted by true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN) values according to:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{8}$$

Furthermore, because we have introduced the confidence mechanism to patch, the accuracy was further improved in the actual test. Because the quality of shooting in the actual test process was affected by various factors, we augmented the images with the ground truth during the patch extraction. The experiments showed that our system was efficient and accurate, and greatly reduced the complexity faced by the operators. Table 7 shows the different parameters and indicators, and the comparison between the existing methods and our system. As can be seen from the table, although the traditional mathematical modeling [29] was faster than the deep-learning-based mapping, its accuracy was not high because it could not adapt to most nonlinear distortion relations. If the distortion correction was carried out without the down-sampling operation, we can see that a large amount of processing time was lost, while the accuracy and *SSIM* exponential increase was not high. By comparing the two models with the same parameters of the patch traversal mechanism without the ground truth and with the ground truth, it can be seen that the accuracy and speed were significantly improved [30,31]. After adding the confidence guiding mechanism, the recognition speed of the model was further improved.

Table 7. Model comparisons were made with the same size and the same frame of the images.

Method	Accuracy (%)	*SSIM*	Time (s)
Function mapping correction [29]	74.2	0.613	3.4
Detection without down-sampling	92.5	0.843	5.0
Detection with down-sampling	90.2	0.821	2.9
Labeled without the ground truth [30]	82.4	0.645	2.6
DenseNet30 [31] + patch (ground truth)	94.8	0.851	1.6
Our model without the patch confidence mechanism	95.4	0.856	1.2
Ours	96.3	0.863	0.6

5. Conclusions

In this study, we used a two-stage model to manage 3D crack detection and calibration. We used a microcamera to collect video in real time, and a motor to drive a constant speed feed to extract the image in the video. We corrected the distortion, detected the crack, and finally visualized the 3D detection map of the recognition area. Combined with a stable feed and IMU information, the accurate information of each patch could be extracted, which facilitated the subsequent damage and quantitative analysis, and greatly reduced the operation complexity for the staff. We used the ground truth to guide the learning of the patch, and used the *SSIM* index system for the evaluation. We found that the effect was significantly improved. After many experiments, our end-to-end system finally achieved very good results. This method can greatly shorten the artificial observation of each crack position, and can manually construct the crack trend chart, which improves the work efficiency and reduces the operation difficulty of the detection personnel.

Our research shows that combining some simple algorithms with the nonlinear fitting ability of the neural network can replace part of the work of humans in the field of detection, and has certain promotional value.

However, there are still many difficulties in our approach, such as the high cost of dataset production and having too many training model parameters. In addition, the speed of the model itself was not fast, and the real-time performance was poor. In the future, we hope to optimize this system, and make a lightweight end-to-end model that can complete a 3D visualization of the aperture.

Author Contributions: X.F. and Y.W. conceived and designed the experiments; project administration, J.W.; resources, Y.W.; Y.L. and L.Z. completed the experiments. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the intelligent manufacturing project, and the platform Chinese Sichuan provincial science and technology department key research and development fund, the fundamental Research Funds for the Sichuan Universities (2019YFG0356 and 2018GZDZX0015). And The APC was funded by Sichuan Universities.

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

References

1. Zhang, X.; Li, T.; Gao, H.; Hao, J. Research on vision inspection system for drainage pipelines damage based on pattern recognition. In Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO), Sanya, China, 15–18 December 2007; Volume 1–5, pp. 697–701.
2. Dutagaci, H.; Cheung, C.P.; Godil, A.; Dutağacı, H. Evaluation of 3D interest point detection techniques via human-generated ground truth. *Vis. Comput.* **2012**, *28*, 901–917. [CrossRef]
3. Zhu, J.; Zhang, C.; Qi, H.; Lu, Z. Vision-based defects detection for bridges using transfer learning and convolutional neural networks. *Struct. Infrastruct. Eng.* **2019**, 1–13. [CrossRef]
4. Lin, C.-S.; Chen, S.-H.; Chang, C.-M.; Shen, T.-W. Crack Detection on a Retaining Wall with an Innovative, Ensemble Learning Method in a Dynamic Imaging System. *Sensors* **2019**, *19*, 4784. [CrossRef] [PubMed]

5. Khani, M.M.; Vahidnia, S.; Ghasemzadeh, L.; Ozturk, Y.E.; Yuvalaklioglu, M.; Akin, S.; Ure, N.K. Deep-learning-based crack detection with applications for the structural health monitoring of gas turbines. *Struct. Health Monit.* **2019**. [CrossRef]
6. Hao, X.-L.; Liang, H. A multi-class support vector machine real-time detection system for surface damage of conveyor belts based on visual saliency. *Measurement* **2019**, *146*, 125–132. [CrossRef]
7. Wu, S.; Wu, Y.; Cao, D.; Zheng, C. A fast button surface defect detection method based on Siamese network with imbalanced samples. *Multimed. Tools Appl.* **2019**, *78*, 34627–34648. [CrossRef]
8. Zhang, Y.; Cui, X.; Liu, Y.; Yu, B. Tire Defects Classification Using Convolution Architecture for Fast Feature Embedding. *Int. J. Comput. Intell. Syst.* **2018**, *11*, 1056–1066. [CrossRef]
9. Schneider, D.; Holtermann, T.; Merhof, D. A traverse inspection system for high precision visual on-loom fabric defect detection. *Mach. Vis. Appl.* **2014**, *25*, 1585–1599. [CrossRef]
10. Chen, Y.-J.; Tsai, J.-C.; Hsu, Y.-C. A real-time surface inspection system for precision steel balls based on machine vision. *Meas. Sci. Technol.* **2016**, *27*, 074010. [CrossRef]
11. 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]
12. Zhang, G.; Wei, Z. A position-distortion model of ellipse centre for perspective projection. *Meas. Sci. Technol.* **2003**, *14*, 1420–1426. [CrossRef]
13. Yoshizawa, T.; Wakayama, T. Development of an inner profile measurement instrument using a ring beam device. *Photonics Asia 2010* **2010**, *7855*, 78550.
14. Islam, M.M.M.; Kim, J.-M. Vision-Based Autonomous Crack Detection of Concrete Structures Using a Fully Convolutional Encoder-Decoder Network. *Sensors* **2019**, *19*, 4251. [CrossRef] [PubMed]
15. Yang, X.; Chen, S.; Jin, S.; Chang, W. Crack Orientation and Depth Estimation in a Low-Pressure Turbine Disc Using a Phased Array Ultrasonic Transducer and an Artificial Neural Network. *Sensors* **2013**, *13*, 12375–12391. [CrossRef] [PubMed]
16. Steckenrider, J.J.; Furukawa, T. A Probabilistic Superpixel-Based Method for Road Crack Network Detection. *Adv. Intell. Syst.* **2020**, *943*, 303–316.
17. Garcia, R.; Pardal, P.; Kuga, H.; Zanardi, M. Nonlinear filtering for sequential spacecraft attitude estimation with real data: Cubature Kalman Filter, Unscented Kalman Filter and Extended Kalman Filter. *Adv. Space Res.* **2019**, *63*, 1038–1050. [CrossRef]
18. Fang, C.; Shang, Y.; Xu, N. MUFOLD-SS: New deep inception-inside-inception networks for protein secondary structure prediction. *Proteins Struct. Funct. Bioinform.* **2018**, *86*, 592–598. [CrossRef]
19. Lee, Y.; Kim, H.; Park, E.; Cui, X.; Kim, H. Wide-residual-inception networks for real-time object detection. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 758–764.
20. Pouyanfar, S.; Chen, S.-C.; Shyu, M.-L. An efficient deep residual-inception network for multimedia classification. In Proceedings of the 2017 IEEE International Conference on Multimedia and Expo (ICME), Hong Kong, China, 10–14 July 2017; pp. 373–378.
21. Im, D.; Han, D.; Choi, S.; Kang, S.; Yoo, H.-J. DT-CNN: Dilated and Transposed Convolution Neural Network Accelerator for Real-Time Image Segmentation on Mobile Devices. In Proceedings of the 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 26–29 May 2019; pp. 1–5.
22. Wang, S.Q.; Rehman, A.; Wang, Z.; Ma, S.W.; Gao, W. SSIM-Motivated Rate-Distortion Optimization for Video Coding. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 516–529. [CrossRef]
23. Rong, J.; Huang, S.; Shang, Z.; Ying, X. Radial Lens Distortion Correction Using Convolutional Neural Networks Trained with Synthesized Images. *Comput. Vis.* **2017**, *10113*, 35–49.
24. Lai, W.-S.; Huang, J.-B.; Ahuja, N.; Yang, M.-H. Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5835–5843.
25. Cha, Y.; Choi, W.; Büyüköztürk, O. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Comput. Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [CrossRef]
26. Bataineh, B.; Abdullah, S.N.H.S.; Omar, K. An adaptive local binarization method for document images based on a novel thresholding method and dynamic windows. *Pattern Recognit. Lett.* **2011**, *32*, 1805–1813. [CrossRef]

27. He, T.; Zhang, Z.; Zhang, H.; Zhang, Z.; Xie, J.; Li, M. Bag of Tricks for Image Classification with Convolutional Neural Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–21 June 2019; pp. 558–567.
28. Zhan, L.; Wang, Y. Stable and Refined Style Transfer Using Zigzag Learning Algorithm. *Neural Process. Lett.* **2019**, *50*, 2481–2492. [CrossRef]
29. Wu, F.; Li, Q.; Li, S.; Wu, T. Train rail defect classification detection and its parameters learning method. *Measurement* **2020**, *151*, 107246. [CrossRef]
30. Chen, Y.; Liang, X.; Zuo, M.J. An improved singular value decomposition-based method for gear tooth crack detection and severity assessment. *J. Sound Vib.* **2020**, *468*, 115068. [CrossRef]
31. Hartl, R.; Landgraf, J.; Spahl, J.; Bachmann, A.; Zaeh, M.F. Automated visual inspection of friction stir welds: A deep learning approach. In Proceedings of the Multimodal Sensing: Technologies and Applications, Munich, Germany, 26–27 June 2019; SPIE-Intl Soc. Optical Eng.: London, UK, 2019; Volume 11059, p. 1105909.