

AUTOMATIC DETECTION AND DIMENSIONAL MEASUREMENT OF MINOR CONCRETE CRACKS WITH CONVOLUTIONAL NEURAL NETWORK

Y. Guo^{1,2}, Z. Wang², X. Shen^{1,*}, K. Barati¹, J. Linke²

¹ School of Civil and Environmental Engineering, University of New South Wales, Sydney, NSW 2052, Australia –
(youheng.guo, x.shen, khalegh.barati)@unsw.edu.au

² Linke & Linke Surveys, Sydney, NSW 2019, Australia – jadynwang@gmail.com, j.linke@llsurveys.com.au

Commission IV, WG IV/9

KEY WORDS: Convolutional Neural Network, Semantic Segmentation, Digital Single-Lens Reflex Camera, Crack Detection, Crack Measurement.

ABSTRACT:

The increasing number of aging infrastructures has drawn attention among the industry as the results caused by critical infrastructure failure could be destructive. It is essential to monitor the infrastructure assets and provide timely maintenance. However, one of the crucial problems is that the budget allocated to the maintenance stage is much less than that for the designing and construction stages. The cost of labor, equipment, and vehicles are significant. Therefore, it is impossible to perform a thorough inspection by human inspectors over each asset. A more efficient method will be needed to solve this problem. This paper aims to provide an automatic approach to detecting and measuring the dimensions of minor cracks that appear on concrete structures with a noisy background. This research also investigates the relationship between image pixel size, accuracy, detection rate of cracks, and shooting distance of images. The proposed method will be able to reduce the cost and increase accuracy. A case study was performed on a concrete sewer with cracks distributed on the surface in Sydney, New South Wales, Australia.

1. INTRODUCTION

The problem with aging infrastructure has become one of the most concerning issues globally, especially in the developed countries. Infrastructures vary from roads, rails, and public buildings to bridges, power plants, and dams. Any potential failure would cause severe impacts to both human life and economics. As most of the crucial infrastructures are made of reinforced concrete structures, such as bridges, power plants and water dams, the inspection of these structures is considered in priority. With the increasing age, defects such as cracking, spalling, and corrosion could inevitably appear. Once a certain amount of defects is identified, a decision must be made to repair the defects or even abandon the asset by balancing the cost. A comprehensive inspection is indispensable to support the decision-making.

The traditional inspection conducted by human workers is risky when accessing dangerous areas (such as heights or tunnels). Another shortcoming of manual inspection is the difficulty of recording. It is hard to record the defect's exact location on a curved surface like a power plant or chimney, leading to record inconsistency. Besides, it is time-consuming for one inspector to conduct the examination. If multiple inspectors are involved, errors might appear due to the difference in recognition. The potential market for solving this problem is enormous with low efficiency and high cost.

Recently, some research has been conducted to perform a crack assessment using unmanned aerial vehicles (Liu et al., 2020) and apply deep learning techniques for crack segmentation (Wang et al., 2022). Semantic segmentation has been one of the emerging

technologies and applied to various industries. With the help of semantic segmentation in the asset inspection industry, the computer will automatically complete the time-consuming part of the work by labeling and measuring the defects. Instead of hanging on a wall to capture the data, an experienced inspector will only need to sit in the office, review the distribution, analyze results, and provide professional opinions. Furthermore, by creating a 3D mesh model using photogrammetry and machine vision, the geo-location of the defects can be recorded in digital format.

This paper aims to provide an automated approach to detecting and measuring the dimensions of minor cracks that appear on concrete structures with a noisy background. The relationship between actual pixel size, accuracy, detection rate, and shooting distance is also revealed. It starts with a literature review focusing on convolutional neural network (CNN)-related semantic segmentation technology and previously used methods for crack measurement. Next, a thorough methodology is presented to introduce a practical workflow to conduct crack measuring utilizing semantic segmentation and computer vision, including data capturing, crack detection, and crack measuring. Based on the proposed methodology, a case study over the cracks on a concrete sewer in Sydney, New South Wales, Australia, was conducted to prove the automatic workflow's applicability, and the relationship between accuracy and shooting distance was discussed.

* Corresponding author

2. LITERATURE REVIEW

The development of different CNN models is reviewed in this section. Different approaches to obtaining the actual crack dimensions are also examined.

2.1 CNN Models

Manual inspection is still widely used in recent years, as experience is the critical issue for this work. However, introducing artificial intelligence will help reduce the burden by scaling down the area of interest. Artificial intelligence has been developing rapidly in recent years, especially in two dimensions. Semantic segmentation algorithms are generally based on a convolutional neural network, a relatively mature computer science technology. Krizhevsky et al. (2012) trained a deep convolutional neural network with 60 million parameters and achieved a significant score in the ImageNet contest. Zeiler and Fergus (2014) revealed the mechanism of AlexNet and developed a new model that outperformed AlexNet. Szegedy et al. (2015) proposed an Inception network that optimized the utilization of computing resources. Szegedy et al. (2016) provided principles to design a high-performance network with low computation cost. Ronneberge et al. (2015) proposed an efficient strategy to use available samples with annotations fully.

Based on CNN, many researchers have developed different structures for different purposes; for example, Tesla used a 2D AI detection system to identify various objects on the road and enable automatic driving, AI facial identification, and motion capturing (Qassim et al., 2018). Defect detection is one of the semantic segmentation applications in the Architecture, Engineering, and Construction (AEC) industry. Kim et al. (2020) stated that semantic segmentation-driven crack detection is more objective and reliable than traditional manual inspection.

2.2 Crack Detection with CNN

Regarding crack detection, most algorithms have similar structures, including encoder and decoder. The encoder is usually trained with many annotated datasets, including everyday objects, and training would be very time-consuming. The decoder is trained with a smaller dataset for a particular project. Oliveira and Correia (2012) proposed an automatic system for crack detection and characterization, and the proposed algorithm could detect multiple cracks from 56 images in about 2 min. Chen et al. (2019) suggested a simple and improved structure of convolutional neural networks achieving high accuracy. Chen et al. (2019) believed a large convolution and pooling methodology with fewer network layers could be utilized to get a better result for simple crack identification. By setting the learning rate to 0.01, Li and Zhao (2019) developed an algorithm with high accuracy based on CNN structure and AlexNet. Liu et al. (2019b) adopted U-Net for high efficiency and robustness. Dung (2019) proposed a crack detection method based on FCN for semantic segmentation on concrete crack images. The automated crack identification and visualization algorithm used by Jang et al. (2019) are enabled by transfer learning from GoogLeNet. Qu et al. (2020) applied LeNet-5 to classify the cracks and optimized VGG16 to extract concrete crack characteristics. Chow et al. (2020) provided an artificial intelligence-based inspection workflow for anomaly detection and reduced the search space of defects up to 80% for minor defect regions. Wang and Su (2022) suggested a SegCrack model including a hierarchically structured transformer encoder to output features and a top-down pathway with lateral connections to upsample and fuse features. Although

much research has been conducted on crack detection, the labeled area is still not accurate enough for minor cracks.

2.3 Crack Measurement with Image Processing

With a captured crack image, measuring the actual dimensions will be a crucial step as it can quantify the defects and enable inspectors to evaluate the status of the result. Some researchers have performed experiments to extract the information from images. Cho et al. (2018) proposed an edge-based crack detection technique. To measure the average width of the cracks in pixels, Feng et al. (2020) divided the area by the length of the crack skeleton. Vashpanov et al. (2019) developed a method to determine the crack's dimension based on the image's pixel intensity distribution and achieved an accuracy of less than $\pm 15\%$. Existing research adopted a fixed distance to calculate the actual size. However, less effort was put into the relationship between shooting distance and measured results.

3. METHODOLOGY

The proposed methodology will present a practical workflow to conduct crack measuring. The proposed neural network is adjusted for minor crack detection in a noisy background. In this section, data capturing, crack detection, and crack measuring will be presented separately.

3.1 Data Capturing

It is easier to detect tiny cracks on the concrete surface from images. Therefore, in this study, two-dimensional images are the primary data source. The device and technique for capturing images will be reviewed in the following two sections.

The primary type of dataset will be 2D images. Therefore, a digital single-lens reflex (DSLR) camera will be needed to capture the photos. The reason to use DSLR is because of the fixed focal length, which will be used in the later process. Moreover, it should be noted that the smartphone's camera is not recommended as the parameters could be processed in the background, which might lead to errors. The distance between the camera and the object must be measured by tape. Alternatively, using laser measure will be more convenient.

In this research, a simple routine is considered. The camera should point at the target object perpendicularly. Furthermore, the target crack should be located on a flat surface. More complicated situations, such as cracks found on curved or bumpy surfaces, will be discussed in future research.

3.2 Crack Detection

How to accurately detect and annotate cracks is a crucial step in this research. The backbone of this function is a CNN which has been widely applied in semantic segmentation. The proposed neural network is adjusted for minor crack detection.

3.2.1 Structure of the Model: CNN is the leading technology used to realize crack detection. Regarding crack detection, most models have a similar structure: encoder and decoder. As for encoders, the primary purpose is to extract features and patterns that are either highly abstracted or detailed. It is usually trained with many annotated datasets, including everyday objects (such as trees, cars, and light poles). The reason for using a broad dataset is to train the ability of the encoder to code the objects into vectors. However, training this would be very time-consuming. Moreover, for decoder should be trained with a

specific dataset for a particular object. The proposed convolutional neural network model is presented below (see Figure 1).

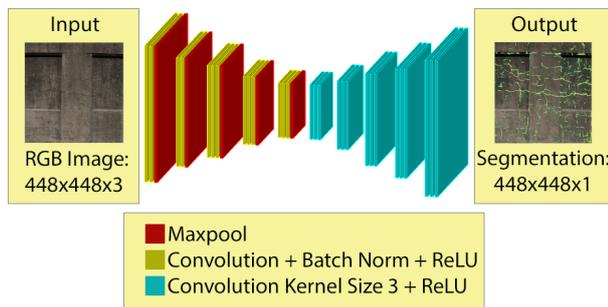


Figure 1. The structure of the proposed CNN model.

3.2.2 Training Dataset: The first task is to capture enough images containing concrete cracks. According to the learning curve provided by Banko and Brill (2001), the algorithm's accuracy increased dramatically with the sample size when the sample is under 100 million inputs. From 100 million to 1 billion, the increase is much smaller. Therefore, a good number of images will achieve satisfactory accuracy while using the minimum resources.

Ideally, the images are taken from a similar environment to the project. However, this does not always work due to the limited access to similar areas. Alternatively, there are some available datasets online with thousands of images. The training dataset used in this paper includes CRACK500 (Zhang et al., 2016), CRACK500 (Yang et al., 2019), GAPS384 (Eisenbach et al., 2017), CFD (Shi et al., 2016), AEL (Amhaz et al., 2016), cracktree200, DeepCrack (Liu et al., 2019a), and CSSC (Yang et al., 2017).

However, the sizes and distributions of images in different datasets might vary considerably. The algorithm prefers similar size images so that the same weight can be assigned to each image. Therefore, a preprocessing step will be needed to unify the image sizes. Furthermore, it also needs to be noted that the size should not be too small as it might lose the shape of cracks. If the size is too big, processing the image will take a long time. Iyer and Sinha (2005) mentioned a method to increase the contrast or brightness of the image in the preprocessing stage. However, this approach did not make much difference in this research.

The next step is annotating the cracks with a set of edited images. The labeled cracks are the guidance to tell the algorithm the essential features. As this step is quite time-consuming, so the advantages and disadvantages of different approaches should be considered thoroughly before carrying out. Two types of annotation methods were tried. The original image is displayed in Figure 2. The ideal annotation method is drawing a mask on top of the pixel-level cracks, as shown in Figure 2. It will give the best result for training due to the accuracy. However, this method usually needs a considerable amount of time. Alternatively, the second approach is bounding boxes (see Figure 2) which require the operator to draw several boxes along the crack to locate the location approximately. Compared with the former way, this method can reduce the time. However, the accuracy will not be as good as a detailed annotation. Therefore, selecting the technique will be a trade-off depending on the project's requirement. If there are too many images, outsourcing

the job to a platform such as Amazon Mechanical Turk (MTurk) could be an option.

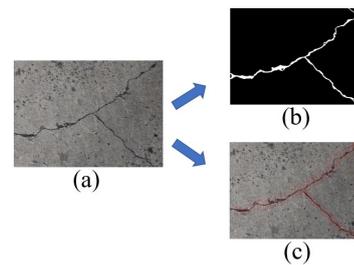


Figure 2. Methods for annotation: (a) the original crack, (b) pixel-level annotation, and (c) bounding box annotation.

The next step is to separate the dataset for different purposes. The suggested division are 70% for training, 20% for validating, and 10% for testing. However, if the database is large enough, for example, 40,000 photos. In this case, 40%, 20%, and 40% are also reasonable. During the training stage, hyper parameters should be adjusted carefully according to the results, which is crucial to improving performance iteratively.

3.3 Crack Measurement

Crack measuring is based on the labeled crack image created from the last step. Since the pixels in the crack area are set to 1 and the rest of the pixels are labeled as 0 for non-crack. Skimage will be applied to the cracked area and firstly depict the border of the cracked area. Then, a skeleton of the crack will be created at the center of two longitudinal lines. After that, the width line will be created perpendicularly to the skeleton, as proposed by Cho et al. (2018) (see Figure 3).

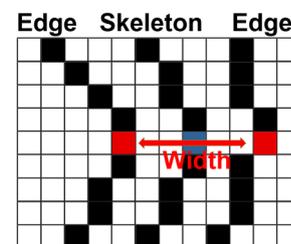


Figure 3. The widths of the crack along with the skeleton (in pixel).

Scale factors will be needed to obtain the actual dimension of the cracks. In this research, scale factors will be obtained from an experiment. The last step is to multiply the pixel dimension by the scale factor. Validation will be conducted by comparing the calculated and actual measurements to test the performance.

4. CASE STUDY

A case study is presented in this section to validate the process of crack measuring. A field experiment was carried out on a concrete sewer with cracks distributed on the surface in Sydney, New South Wales, Australia.

4.1 Data Capturing

The first step is examining the structure and planning a proper route to capture data. As the shape of the concrete sewer is a long box, images were taken along the sewer.

A simplified pattern is applied because the shooting angle and distortion affect the result. Only images perpendicular to the surface will be analyzed in the following procedures. Sony Alpha 7 was used to take photos. The focal length is 24mm, and the image size is 9,504×6,336.

4.2 Semantic Segmentation Process

This section presents a practical workflow for the implementation of two-dimensional artificial intelligence.

4.2.1 The Optimum Size for Processing: An optimum size should be selected due to the limitation of computer power. As the pixel size of the raw image is 9,504×6,336, it can be cut into 3,584×3,584, 1,792×1,792, 896×896, and 448×448. After running the algorithm on each dimension, the size 448×448 gives the best result.

The encoder is VGG16, referenced from Simonyan and Zisserman (2014). It initially has 16 weight layers, and each layer consists of Maxpool and Convolution + Batch Norm + ReLU. After testing, it was found that adding Batch Norm between Convolution and ReLU could improve the performance. The input image is cut into 448x448 and fed into the algorithm as a [448×448×3] matrix. Furthermore, the dimensions for each layer are listed below (see Table 1).

No. of Layer	Dimension
1	224 × 224 × 64
2	112 × 112 × 128
3	56 × 56 × 256
4	28 × 28 × 512
5	14 × 14 × 512

Table 1. The dimensions of each layer in the encoder.

The decoder is designed for crack detection. Each layer consists of Bilinear Interpolate and Convolution Kernel Size 3 + ReLU. The dimensions for each layer are listed below (see Table 2).

No. of Layer	Dimension
5	28 × 28 × 256
4	56 × 56 × 256
3	112 × 112 × 64
2	224 × 224 × 32
1	448 × 448 × 32

Table 2. The dimensions of each layer in the decoder.

The final layer consists of Convolution (kernel size 3) + ReLU, Convolution Kernel Size 1, and Log SoftMax. The dimension of the output image is [448×448×1], and the value of each pixel is either 0 or 1, representing crack or non-crack.

Since the dataset is biased (the number of cracks and non-crack images is not equal), the traditional Binary Cross Entry (BCE) loss does not work as it tends to regard the picture as not having cracks. Instead, a focal loss method based on the structure proposed by Lin et al. (2017) is applied for classification. Then, backpropagation will be performed to adjust the parameters. The VGG16 + Focal Loss model was trained on a smaller dataset with fewer epochs. ResNet has been commonly used for crack detection. U-Net (Liu et al., 2019b) is one of the most recent developments based on ResNet. A comparison between VGG16 + Focal Loss, VGG16 + BCE Loss, and Baseline U-Net will be conducted (see Table 3). The overall performance of

VGG16+Focal Loss with an F1 Score of 0.613 is better than the other two models.

	Average Precision	Average Recall	F1 Score
Baseline U-Net	0.616	0.582	0.598
VGG16+ BCE Loss	0.432	0.603	0.503
VGG16+ Focal Loss	0.566	0.670	0.613

Table 3. The comparison between three models.

The results of annotated images for some evident and thin cracks are presented in Figures 4 and 5.

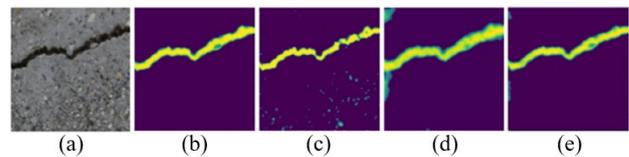


Figure 4. The results of different models for obvious cracks: (a) original image, (b) Baseline U-Net, (c) VGG16 + BCE Loss, (d) VGG16 + Focal Loss, and (e) VGG16 + Focal Loss with more training datasets.

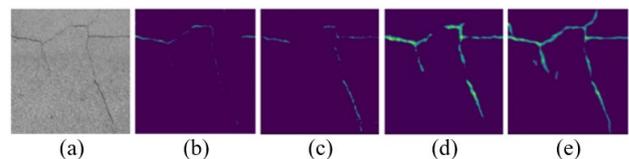


Figure 5. The results of different models for thin cracks: (a) original image, (b) Baseline U-Net, (c) VGG16 + BCE Loss, (d) VGG16 + Focal Loss, and (e) VGG16 + Focal Loss with more training datasets.

As can be seen, for obvious cracks, the performance of the crack is the same. Some noises were introduced in the VGG16 + BCE Loss model. However, only the VGG16 + Focal Loss model could identify most thin cracks. With more datasets fed to train the algorithm, the performance kept improving. The latest model was trained with 855 images, and the results are shown in Figures 4 and 5.

4.2.2 Statistical Results for the Accuracy: The first test dataset contains 1,695 images with 1,483 crack images and 212 non-crack images. The algorithm's performance was assessed using equation (1) to calculate the accuracy. The tested accuracy for this dataset is 95.81%.

As the first test dataset has more crack images, the portion of images in the second test dataset was adjusted to make sure both situations were tested. Therefore, the number of crack images was reduced from 1,483 to 56, and the number of non-crack photos remained the same. The accuracy for the second dataset is 91.79%.

4.2.3 Results of Automatic Detection on Target Dataset: The Semantic segmentation algorithm was run on a dataset containing ten sizes of the cracks from 0.1mm to 1.0mm from distances of 1.0m, 1.5m, 2.0m, 2.5m, and 3.0m. The labels indicating different sizes of cracks are shown in Figures 7 and 8, which

affect the detecting result. Therefore, the tags are removed to achieve a better result.

For small cracks (0.1mm-0.5mm) shot from 1m, only a tiny part of the cracks was detected (see Figure 6). While shot from 3m, most of the cracks were marked, but the area is much larger than the actual one.

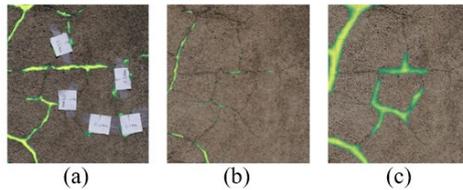


Figure 6. Marked cracks (0.1mm-0.5mm): (a) 1m with the label, (b) 1m without a label, and (c) 3m without a label.

For larger cracks (0.6mm-1.0mm) shot from 1m, most of the cracks were marked in Figure 7. While the photo from 3m, although most of the cracks were drawn, the area is still more extensive than the actual crack.

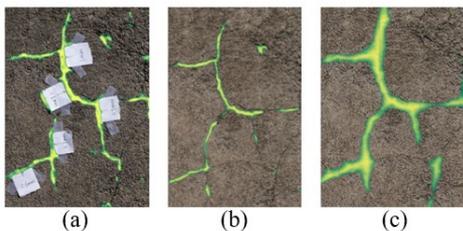


Figure 7. Marked cracks (0.6mm-1.0mm): (a) 1m with label, (b) 1m without label, and (c) 3m without label.

The detected results for cracks under 0.5mm are unsatisfactory, which could be caused by many reasons. Cracks are very tiny, the edges are rough, and the shadow cast around the crack could make the algorithm hard to identify the shape of the crack accurately.

4.3 Crack Measurement

As mentioned in 3.3, the first step is annotating the crack area using semantic segmentation. The cracked and the non-crack regions are colored white and black to gain maximum contrast. Then, the shape of the crack is outlined using skimage, and the skeletons of the cracks are located at the center of two lines. The step of processing is shown in Figure 8.

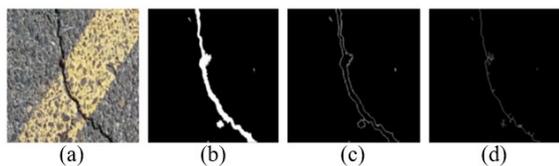


Figure 8. The step of processing for crack measurement: (a) the original crack, (b) the detected crack area, (c) the edge of the crack, and (d) the skeleton of the crack.

After that, the pixel width along the skeleton will be counted. The distance from the pixel to the edge will be measured perpendicularly for every pixel on the skeleton. Furthermore, it will be applied to both sides of the skeleton.

4.4 The Validation of Results

This section is set to quantify the performance of the proposed methodology from three aspects: the rate of recognition, the accuracy of manually derived scale factor, and the accuracy of automatic measurement.

In the first part, the numbers of correctly detected cracks and undetected were manually counted for each image. Moreover, rates of recognition could be calculated. The second part is to determine the scale factor. The process is manual, where the number of pixels for gauges and cracks were manually counted. Error rates were also calculated and compared to find an optimal condition to apply the proposed methodology. The third part is an automated way to calculate the actual crack size based on the previously mentioned methodology. Furthermore, error rates are estimated to validate the applicability of the whole process.

4.4.1 Rate of Recognition: For each processed image, correctly detected cracks are labeled with red dots, and undetected cracks are labeled with blue dots, as shown in Figure 9.

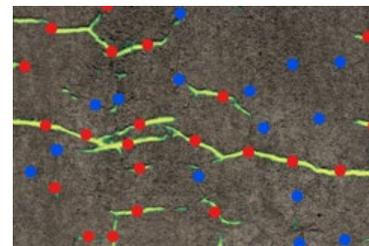


Figure 9. Categorized cracks.

The manually counted process was repeated on each image. The recognition rate can be calculated using Equation 1, and the statistical results are summarized in Figure 10.

$$R = C/(C+U) \times 100\% , \quad (1)$$

Where R = rate of recognition
C = number of correctly detected cracks
U = number of undetected cracks

From Figure 10, it can be concluded that the recognition rate increases as the shooting distance increases.

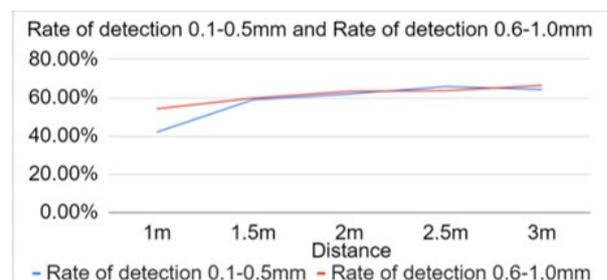


Figure 10. The chart for the rate of recognition.

4.4.2 Manually Derived Scale Factor: As small cracks are the most concerning problems in the industry, a manual experiment was conducted over the captured dataset with ten sizes of cracks from 0.1mm to 1.0mm from distances of 1.0m, 1.5m, 2.0m, 2.5m, and 3.0m. The test was repeated five times to gain a statistical result. As shown in Figure 11, cracks from 0.6mm to 1.0mm were labeled. A crack gauge was stuck next to the cracks.



Figure 11. Crack measuring gauge and labeled cracks.

The gauge's function is to obtain each pixel's actual dimension. Gauges including 0.6mm, 0.8mm, 1.0mm, 1.2mm, and 1.4mm were selected. As the actual width of the gauge is known and the number of pixels could be counted from the image, the pixel size could be calculated using the actual gauge dimension divided by the counted pixel number (see Table 4).

Distance (m)	Number of Pixels for 1.4mm	Pixel Size (mm/pixel)
1	10	0.140
1.5	7	0.200
2	6	0.233
2.5	4	0.350
3	3	0.467

Table 4. The calculated pixel sizes.

The pixel numbers for every size of the cracks were counted manually from different distances (see Table 5).

Manually Counted Pixels Number for Each Crack					
Distance	0.6mm	0.7mm	0.8mm	0.9mm	1.0mm
1m	5	6	9	10	11
1.5m	3	4	6	7	8
2m	2	2	4	4	5
2.5m	2	2	3	3	4
3m	1	2	2	2	3

Table 5. The number of manually counted pixels.

Then, the sizes of the cracks could be derived by multiplying the pixel sizes from Table 4 and the numbers contained in each crack from Table 5. The calculated results are displayed in Table 6.

Calculated Width for Each Crack (mm)					
Distance	0.6mm	0.7mm	0.8mm	0.9mm	1.0mm
1m	0.700	0.840	1.260	1.400	1.540
1.5m	0.600	0.800	1.200	1.400	1.600
2m	0.467	0.467	0.933	0.933	1.167
2.5m	0.700	0.700	1.050	1.050	1.400
3m	0.467	0.933	0.933	0.933	1.400

Table 6. The calculated width for each crack.

After that, the error rate can be calculated using the Equation 2.

$$\text{Absolute Error Rate} = \frac{|\text{Calculated Width} - \text{Measured Width}|}{\text{Measured Width}} \times 100\% \quad (2)$$

The absolute error rate is displayed in Table 7.

Absolute Error Rate					
Distance	0.6mm	0.7mm	0.8mm	0.9mm	1.0mm
1m	17%	20%	58%	56%	54%
1.5m	0%	14%	50%	56%	60%
2m	22%	33%	17%	4%	17%
2.5m	17%	0%	31%	17%	40%
3m	22%	33%	17%	4%	40%

Table 7. The absolute error rates.

Line charts of the average error rate with standard deviation are presented in Figure 12. The average error for the measurement of 0.1mm crack is much higher than other sizes. As can be seen, the error rate gets higher as the shooting distance increases. Moreover, as the size increases, the error rate presents a decreased tendency. A conclusion can be drawn that the cracks under 0.5mm cannot be accurately measured with the current setup.

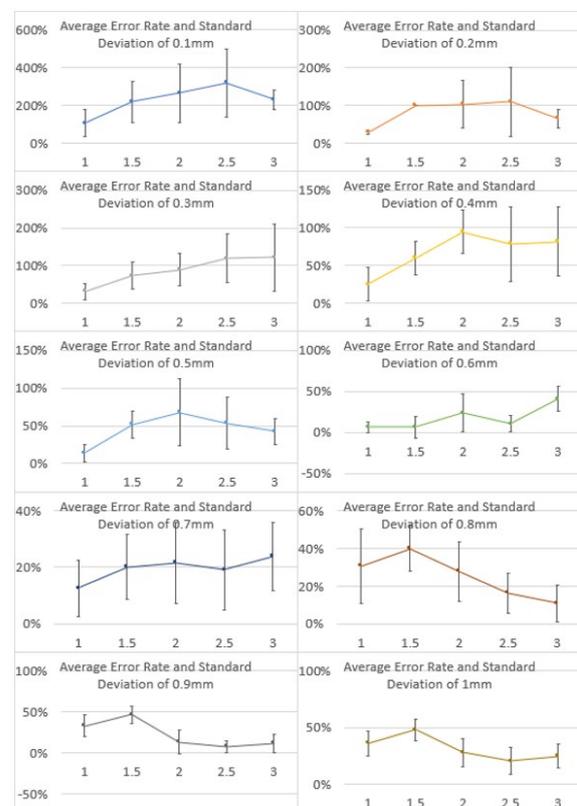


Figure 12. Individual charts of average error rate with standard deviations for each crack.

4.4.3 Automatically Detected Crack Dimensions: The accuracy for 0.6mm to 1.0mm is relatively high, and most cracks are labeled. The dataset used in this section will be labeled cracks of 0.6mm to 1.0mm and shoot from 1m and 3m. The average pixel sizes from different distances are shown in Table 8.

Distance (m)	Pixel Size (mm/pixel)
1	0.134
1.5	0.200
2	0.258
2.5	0.293
3	0.380

Table 8. Average pixel sizes from different distances.

As shown in Figure 13, the automatically measured pixel widths are labeled. The actual width can be automatically calculated by multiplying a scale factor and annotated in the image.

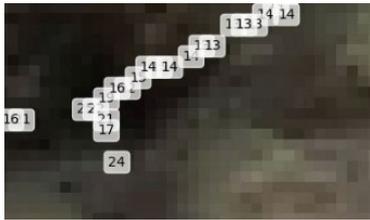


Figure 13. The number of measured widths along the crack.

The automatically detected number of pixels, calculated widths, and absolute error rates are listed in Tables 9 to 11.

Automatically Detected Pixels Number for Each Crack					
Distance	0.6mm	0.7mm	0.8mm	0.9mm	1.0mm
1m	11	6	15	5	10
1.5m	8	7	18	13	13
2m	9	13	14	12	15
2.5m	7	12	14	11	14
3m	11	11	13	10	15

Table 9. The number of automatically detected pixels.

Calculated Width for Each Crack (mm)					
Distance	0.6mm	0.7mm	0.8mm	0.9mm	1.0mm
1m	1.470	0.802	2.005	0.668	1.337
1.5m	1.600	1.400	3.600	2.600	2.600
2m	2.322	3.354	3.612	3.096	3.870
2.5m	2.053	3.520	4.107	3.227	4.107
3m	4.180	4.180	4.940	3.800	5.700

Table 10. The calculated widths.

Absolute Error Rate					
Distance	0.6mm	0.7mm	0.8mm	0.9mm	1.0mm
1m	145%	15%	151%	26%	34%
1.5m	167%	100%	350%	189%	160%
2m	287%	379%	352%	244%	287%
2.5m	242%	403%	413%	259%	311%
3m	597%	497%	518%	322%	470%

Table 11. The absolute error rates for each crack.

As seen in Tables 9 to 11, the error rates generated 1 meter away from the cracks are relatively small. However, the overall performance is still not ideal. Although the proposed algorithm's accuracy should be further improved, the applicability of the proposed workflow has been proved in this paper.

5. CONCLUSIONS

This paper presented an automatic approach to detecting and measuring the dimension of concrete cracks with CNN method. The relationship between image pixel size, accuracy, and detection rate of cracks, and shooting distance of images is investigated. The core technology used for crack detection is CNN-based semantic segmentation. The encoder adopts VGG16, and the decoder is adjusted for crack detecting. Compared to the baseline model, this model added a focal loss layer to improve

the performance. The overall F1 score is improved from 0.598 to 0.613. And the stability of the model is tested on biased datasets with an accuracy over 91%. After that, a crack measuring function based on the semantic segmentation processed image was introduced. Furthermore, a field experiment was conducted to test the accuracy of crack measurement and revealed the relationship between shooting distance and accuracy. It is found that with the decreasing shooting distance, automatic crack measurement error rate drops from 65% to 32%. While the crack recognition rate drops approximately from 65% to 50%. As the average error rate for cracks under 0.5mm (102%) is significantly larger than the cracks over 0.5mm (23%), the proposed measurement method is not reliable for cracks under 0.5mm.

As the current research simplified some conditions, everyday situations will be considered in a future study, e.g., the image was taken perpendicularly to the cracks, and the detected cracks are distributed on a flat surface. The geometric distortion correction should be considered for precise measurements. More datasets will also be needed to train the proposed semantic segmentation algorithm and increase the accuracy to meet the industrial requirement. Furthermore, by creating a 3d mesh model using photogrammetry and machine vision, the geo-location of the defects can be recorded in digital format, which will facilitate the analysis of structural health conditions.

ACKNOWLEDGMENTS

This research is supported by an Australian Government Research Training Program (RTP) Scholarship. We sincerely thank Dr. Yincai Zhou (University of New South Wales) for providing instructions on photogrammetry. We thank Mr. Alastair Linke (Linke & Linke Surveys), and Mr. Samuel Yu (Linke & Linke Surveys) for providing equipment and suggestions for image capturing.

REFERENCES

- Amhaz, R., Chambon, S., Idier, J. & Baltazart, V. 2016. Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection. *IEEE Transactions on Intelligent Transportation Systems*, 17, 2718-2729.
- Banko, M. & Brill, E. Scaling to very very large corpora for natural language disambiguation. Proceedings of the 39th annual meeting of the Association for Computational Linguistics, 2001. 26-33.
- Chen, K., Yadav, A., Khan, A., Meng, Y. & Zhu, K. 2019. Improved crack detection and recognition based on convolutional neural network. *Modelling and simulation in engineering*, 2019.
- Cho, H., Yoon, H.-J. & Jung, J.-Y. 2018. Image-based crack detection using crack width transform (CWT) algorithm. *IEEE Access*, 6, 60100-60114.
- Chow, J. K., Su, Z., Wu, J., Li, Z., Tan, P. S., Liu, K.-f., Mao, X. & Wang, Y.-H. 2020. Artificial intelligence-empowered pipeline for image-based inspection of concrete structures. *Automation in Construction*, 120, 103372.
- Dung, C. V. 2019. Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, 99, 52-58.

- Eisenbach, M., Stricker, R., Seichter, D., Amende, K., Debes, K., Sesselmann, M., Ebersbach, D., Stoeckert, U. & Gross, H.-M. How to get pavement distress detection ready for deep learning? A systematic approach. 2017 international joint conference on neural networks (IJCNN), 2017. IEEE, 2039-2047.
- Feng, C., Zhang, H., Wang, H., Wang, S. & Li, Y. 2020. Automatic pixel-level crack detection on dam surface using deep convolutional network. *Sensors*, 20, 2069.
- Iyer, S. & Sinha, S. K. 2005. A robust approach for automatic detection and segmentation of cracks in underground pipeline images. *Image and Vision Computing*, 23, 921-933.
- Jang, K., Kim, N. & An, Y.-K. 2019. Deep learning-based autonomous concrete crack evaluation through hybrid image scanning. *Structural Health Monitoring*, 18, 1722-1737.
- Kim, J. J., Kim, A.-R. & Lee, S.-W. 2020. Artificial neural network-based automated crack detection and analysis for the inspection of concrete structures. *Applied Sciences*, 10, 8105.
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Li, S. & Zhao, X. 2019. Image-based concrete crack detection using convolutional neural network and exhaustive search technique. *Advances in Civil Engineering*, 2019.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision, 2017. 2980-2988.
- Liu, Y., Yao, J., Lu, X., Xie, R. & Li, L. 2019a. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338, 139-153.
- Liu, Y. F., Nie, X., Fan, J. S. & Liu, X. G. 2020. Image - based crack assessment of bridge piers using unmanned aerial vehicles and three - dimensional scene reconstruction. *Computer - Aided Civil and Infrastructure Engineering*, 35, 511-529.
- Liu, Z., Cao, Y., Wang, Y. & Wang, W. 2019b. Computer vision-based concrete crack detection using U-net fully convolutional networks. *Automation in Construction*, 104, 129-139.
- Oliveira, H. & Correia, P. L. 2012. Automatic road crack detection and characterization. *IEEE Transactions on Intelligent Transportation Systems*, 14, 155-168.
- Qassim, H., Verma, A. & Feinzimer, D. Compressed residual-VGG16 CNN model for big data places image recognition. 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018. IEEE, 169-175.
- Qu, Z., Mei, J., Liu, L. & Zhou, D.-Y. 2020. Crack detection of concrete pavement with cross-entropy loss function and improved VGG16 network model. *Ieee Access*, 8, 54564-54573.
- Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention, 2015. Springer, 234-241.
- Shi, Y., Cui, L., Qi, Z., Meng, F. & Chen, Z. 2016. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17, 3434-3445.
- Simonyan, K. & Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015. 1-9.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the inception architecture for computer vision. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. 2818-2826.
- Vashpanov, Y., Son, J.-Y., Heo, G., Podousova, T. & Kim, Y. S. 2019. Determination of geometric parameters of cracks in concrete by image processing. *Advances in Civil Engineering*, 2019.
- Wang, J. J., Liu, Y. F., Nie, X. & Mo, Y. L. 2022. Deep convolutional neural networks for semantic segmentation of cracks. *Structural Control and Health Monitoring*, 29, e2850.
- Wang, W. & Su, C. 2022. Automatic concrete crack segmentation model based on transformer. *Automation in Construction*, 139, 104275.
- Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X. & Ling, H. 2019. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 21, 1525-1535.
- Yang, L., Li, B., Li, W., Liu, Z., Yang, G. & Xiao, J. Deep concrete inspection using unmanned aerial vehicle towards cssc database. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2017. 24-28.
- Zeiler, M. D. & Fergus, R. Visualizing and understanding convolutional networks. European conference on computer vision, 2014. Springer, 818-833.
- Zhang, L., Yang, F., Zhang, Y. D. & Zhu, Y. J. Road crack detection using deep convolutional neural network. 2016 IEEE international conference on image processing (ICIP), 2016. IEEE, 3708-3712.