Sub-dataset Generation and Matching for Crack Detection on Brick Walls using Convolutional Neural Networks

Mehedi Hasan Talukder[1], Shuhei Ota[2], Masato Takanokura[2] and Nobuaki Ishii[2]
[1]*Course of Industrial Engineering and Management, Graduate School of Engineering, Kanagawa University, Yokohama, Japan*
[2]*Department of Industrial Engineering and Management, Faculty of Engineering, Kanagawa University, Yokohama, Japan*

Keywords: Deep Learning, Maintenance of Brick Walls, Color Information.

Abstract: Crack detection is an issue of significant interest in ensuring the safety of buildings. Conventionally, a maintenance engineer performs crack detection manually, which is laborious and time-consuming. Therefore, a systematic crack detection method is required. Among the existing methods, convolutional neural networks (CNNs) are more effective; however, they often fail in the case of brick walls. There are several types of bricks and some may appear to have cracks owing to their structure. Additionally, the joining points of bricks may appear as cracks. It is theorized that if sub-datasets are generated based on the image attributes, and a proper sub-dataset is selected by matching the test image with the sub-datasets, then the performance of the CNN can be improved. In this study, a method consisting of sub-dataset generation and matching is proposed to improve the crack detection in brick walls. CNN learning is conducted with each sub-dataset, and crack detection is performed using a proper learned CNN that is selected by matching the test images with the images in the sub-datasets. Four performance metrics, namely, precision, recall, F-measure, and accuracy, are used for performance evaluation. The numerical experiments show that the proposed method improves the crack detection in brick walls.

1 INTRODUCTION

Crack detection during maintenance is an essential task to ensure the safety of concrete structures (Liu et al., 2019). The number of old buildings, roads, and bridges is increasing worldwide. Numerous buildings, highways, and bridges in Japan, the USA, and other countries have been in service for over 50 years (Road Bureau Japan, 2015; American Society of Civil Engineers, 2017). In Japan, there are approximately 700,000 bridges of which 43% are 50 years or older (Road Bureau Japan, 2015). Cracks may occur in old structures and are considered to be the initial signs of deterioration. The detection of cracks in early stages can help prevent the collapse of buildings, especially after natural disasters. Conventionally, crack detection is performed by a maintenance engineer by manual process which is a time- and labor-intensive task (Dais et al., 2021). However, 50% of towns and 70% of villages in Japan have no technicians available for maintenance (Road Bureau Japan, 2015). In the USA, approximately 40% of bridges have been in service for approximately 50 years; 24% of buildings and one out of every five miles of highways are in poor condition (ASCE, 2017). In these old structures, cracks and sometimes extensive damage can be observed. Therefore, to ensure safety, crack detection is of vital importance in the maintenance of concrete structures (Cha et al., 2017; Ozgenel et al., 2018).

Through advanced digital image technology, several image processing and vision based techniques have been implemented to inspect and detect defects in civil infrastructure (Choi et al., 2014; Neogi et al., 2014; Ozgenel et al., 2018; Qader et al., 2003; Wu et al., 2008; Yeum et al., 2015). The performance of the image processing depends on the environment. Current image processing techniques can detect wall cracks under certain environmental conditions (Talukder et al., 2020; Talukder et al., 2021). Presently, various convolutional neural network (CNN) methods are being used for crack detection in concrete structures (Dung et al., 2019; Huyan et al., 2019; Jacob et al. 2019; Li et al. 2019; Li et al. 2020; Mahtab et al. 2019). CNN methods have proven to

Talukder, M., Ota, S., Takanokura, M. and Ishii, N.
Sub-dataset Generation and Matching for Crack Detection on Brick Walls using Convolutional Neural Networks.
DOI: 10.5220/0010615501910197
In *Proceedings of the 2nd International Conference on Deep Learning Theory and Applications (DeLTA 2021)*, pages 191-197
ISBN: 978-989-758-526-5

be more effective for crack detection than other methods (Cha et al., 2017; Ozgenel et al., 2018). However, their performance is poor due to various environmental conditions, such as shadows, inadequate brightness, and stains (Cha et al., 2017; Talukder et al., 2021; Hoang et al., 2018; Andrushia et al., 2018). With changes in the environmental conditions, the image quality also changes; hence, current CNN methods are less effective. Crack detection in brick walls is especially a major challenge (Ozgenel et al., 2018). The structure of some bricks and the joining points of bricks may appear as cracks, which poses a challenge to current methods.

Current CNN methods use a large dataset consisting of images with different properties for the learning process. There are several types of bricks and some may appear to have cracks owing to their structure. Additionally, the joining points of the bricks may appear as cracks. Hence, it is difficult to detect cracks in brick walls effectively using current CNN methods. It is theorized that if sub-datasets are generated based on the image properties and a proper sub-dataset is selected for crack detection by matching the test image with the images of sub-datasets, then the performance of the CNN can be improved.

The objective of this study is to develop methods of sub-dataset generation and matching which are used for CNN-based crack detection in brick walls. Sub-dataset generation algorithm is used to generate small sub-datasets from a large dataset based on the image features; CNN learning is done by these sub-datasets. After CNN learning, matching algorithm is used to select proper learned CNN by matching the features of input image with those of the sub-datasets. The key novelty of the proposed approach lies in the sub-dataset generation from a large dataset and matching of the test image with the sub-datasets to select a proper sub-dataset for crack detection. Using the proposed method, sub-datasets are generated based on the images having similar attributes. Color models, texture analysis, histogram analysis, etc. are used for image classification (Bianconi et al., 2011; Varma et al., 2005). RGB color provides strong indications and increases the image classification accuracy (Bianconi et al., 2011). The images of brick walls contain sufficient information of R, G, and B values. Hence, the R, G, and B values of images are considered as attributes for the sub-dataset generation and for the matching algorithm. Owing to the sub-dataset generation and matching, a proper sub-dataset can be selected for crack detection. Therefore, the proposed method improves the performance of crack detection in brick walls.

2 RELATED WORKS

Numerous image processing, edge detection, vision-based, CNN-based, and other methods have been developed by researchers for the purpose of crack detection. Ozgenel *et al.* (2018) performed crack detection using vision-based, image processing-based, and CNN-based crack detection methods. Their study showed that shadow and noise remain as challenges in using CNN methods. Cha *et al* .(2017) used the vibration method, image processing, and the CNN method for crack detection; the study states that the crack detection results are substantially affected by noise created from lighting and distortion. Huyan *et al.* (2019) used a fast-region CNN based crack deep network for the purpose of crack detection of sealed unsealed cracks with complex road background. They show that, image surface have a strong influence on the performance of crack detection. According to this study, the performance of CNN is poor for the crack detection for some background conditions such as images with shadow, images with noise, and so on.

Talukder *et al.* (2021) proposed a method consisting of a sub-dataset generation and matching-based CNN for crack detection in concrete walls. Their study used the brightness value of images for the sub-dataset generation and matching as image properties. However, images of brick walls contain sufficient information of RGB colour; hence, in our study, the R, G, and B values of images are used for the sub-dataset generation and matching.

Dais *et al.* (2021) used CNN and transfer learning for the crack detection of brick masonry. CNNs are comparatively effective for crack detection; however, performance is low for brick masonry surface. This is because, there are several types of bricks and the joining points of brick walls are also different in different walls. Therefore, crack detection of brick walls is challenging. A few sample images of different brick walls are presented in Figure 1. The first and second rows of the figure show sample crack and no-crack images of the brick walls, respectively, which were collected from the different buildings of the Kanagawa University, Japan.

Figure 1: A few sample images of different brick walls.

3 PROPOSED METHOD FOR CRACK DETECTION

3.1 Structure of the Proposed Method

In this study, a CNN-based method consisting of sub-dataset generation and matching is proposed for the crack detection in brick walls. The overall structure of the proposed method is illustrated in Figure 2. The mechanism of sub-dataset generation from a large dataset and the matching algorithm that selects a CNN learned using the proper sub-dataset according to a test image is shown in Figure 3.

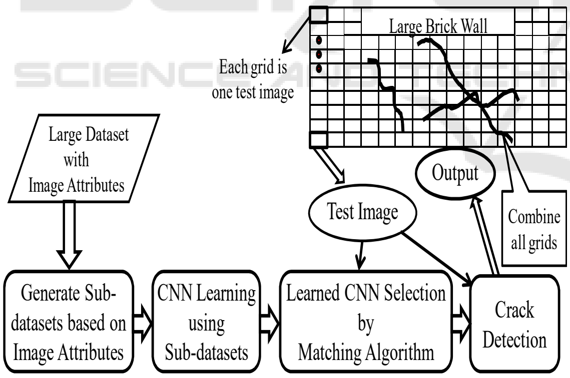

Figure 2: Structure of the proposed crack detection method.

In the proposed crack detection method, a large dataset of images is divided into small sub-datasets based on the attributes of the images, as shown in Figure 2. Following the sub-dataset generation, CNN learning is performed using the sub-datasets. Thereafter, the CNN learned based on the proper sub-dataset is selected using the proposed matching algorithm. The proper learned CNN is selected by matching the attributes of the test image with the attributes of the sub-datasets, and then the crack detection is performed using the selected learned CNN. Generally, the performance of a CNN for crack detection in brick walls is poor when utilizing a large dataset consisting of images of different types of brick walls. To improve the performance of CNN for crack detection of brick walls, the process of sub-dataset generation and matching is performed in the proposed method.

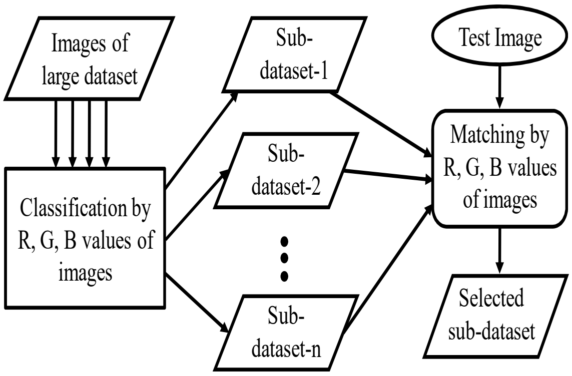

Figure 3: Mechanism of the sub-dataset generation and matching.

In general, the images of brick walls contain sufficient information of R, G, and B values; hence, these values are considered as the image attributes for the sub-dataset generation and the matching algorithm. Figure 3 shows the generation of the small sub-datasets from a large dataset based on the images with similar RGB values. Using the proposed method, a proper learned CNN is selected by matching the RGB values of a test image with the RGB values of the images of sub-datasets used for learning.

3.2 Sub-dataset Generation and Matching

To generate small sub-datasets from a large dataset and to match the test input image with the sub-datasets, the color distances of the R, G, and B values between the images are calculated. In this process, the averages of the R, G, and B values of the images are first calculated. The average R, G, and B values of the *i*-th image are calculated using Equation (1) as follows:

$$\begin{bmatrix} \overline{R}_i \\ \overline{G}_i \\ \overline{B}_i \end{bmatrix} = \frac{1}{n_x n_y} \sum_{x=1}^{n_x} \sum_{y=1}^{n_y} \begin{bmatrix} r_{xy} \\ g_{xy} \\ b_{xy} \end{bmatrix}, \quad (1)$$

where x and y represent the spatial coordinates, n_x and n_y represent the size of the images, and r_{xy}, g_{xy}, and b_{xy} represent the R, G, and B values of the corresponding pixel position in the image, respectively. The average R, G, and B values of two images are calculated using Equations (2) and (3) given below.

$$I_1 = \{\overline{R_1}, \overline{G_1}, \overline{B_1}\} \tag{2}$$

$$I_2 = \{\overline{R_2}, \overline{G_2}, \overline{B_2}\}. \tag{3}$$

Further, the *Color Distance* between these two images can be calculated using Equation (4) as shown below:

$$Color\ Distance = \sqrt{(\bar{R}_1 - \bar{R}_2)^2 + (\bar{G}_1 - \bar{G}_2)^2 + (\bar{B}_1 - \bar{B}_2)^2} \tag{4}$$

To generate small sub-datasets from a large dataset, the *Color Distance* between *0* and C_D was taken for each sub-dataset, where C_D was used as the threshold value for the *Color Distance* parameter. The algorithmic steps of the sub-dataset generation and matching algorithms are explained below.

Algorithmic Steps of Sub-dataset Generation:

1) Calculate the $\bar{R_i}, \bar{G_i}, \bar{B_i}$ values for all images of the large dataset using Equation (1).
2) For $i = 1$ to N // N is the size of the large dataset.
   Select image I_i of the dataset.
3) For $j = i + 1$ to N
   Calculate the *Color Distance* between I_i and I_j using Equation (4).
   If the *Color Distance* is <= C_D, include in the same sub-dataset and remove I_j from the large dataset.
   Repeat steps 2 to 3 until no images remain in the large dataset.
4) Exit when there are no more images in the large dataset.

Algorithmic Steps of Matching:

1) Calculate the $\bar{\bar{R_i}}, \bar{\bar{G_i}}, \bar{\bar{B_i}}$ values, which are the average values of the $\bar{R}, \bar{G}, \bar{B}$ values of the images of sub-dataset *i*.
2) Calculate the $\bar{R}, \bar{G}, \bar{B}$ values of the test image using Equation (1).
3) Calculate the *Color Distance* between the test image and each sub-dataset using Equation (4).
4) The sub-dataset with the minimum *Color Distance* is selected for the crack detection.

4 NUMERICAL EXPERIMENTS

4.1 Dataset Preparation for CNN Learning

To train the CNN, a total of 400 images (200 crack and 200 no-crack images) of brick walls with a size of 227 × 227 pixels were generated by manually cropping from 100 raw images. The raw images of the brick walls were collected from the Internet. A total of 30 images (15 crack and 15 no-crack images) were used for the testing process to evaluate the performance of the proposed method for the crack detection of brick walls.

4.2 Experimental Design

The crack detection was performed using the proposed method in MATLAB. For the experiments, the CNN architecture was designed with eight layers (Cha et al., 2017; Talukder et al. 2021). Table 1 lists the detailed dimensions of each layer and their operations.

In Table 1, *L* represents the layers, and *C* and *P* represent the convolution and pooling, respectively. For the experiments, the number of epochs was set to*10*, and the batch size was *32*.

Table 1: Dimensions of the layers and operations of the CNN.

Layers	Height & Width	Depth	Operator	Height & Width	Depth	No.	Stride
Input	227	3	C1	15	3	24	2
L1	107	24	P1	7	-	-	2
L2	51	24	C2	11	24	48	2
L3	21	48	P2	5	-	-	2
L4	9	48	C3	9	48	96	2
L5	1	96	ReLU	-	-	-	-
L6	1	96	C4	1	96	2	1
L7	1	2	Softmax	-	-	-	-
L8	1	2	-	-	-	-	-

Two experiments, namely, experiments 1 and 2, were performed to evaluate the performance of the proposed crack detection method. Experiment 1 was conducted to evaluate the performance of the CNN for the crack detection in brick walls using a large dataset consisting of the different types of images for the purpose of training the CNN. Experiment 2 was conducted to evaluate the effectiveness of the sub-dataset generation using the RGB values of images as image attributes and the matching algorithm for crack detection.

Images with similar RGB values were obtained for each sub-dataset. Subsequently, the generated sub-datasets were used to perform the CNN learning. The properly learned CNN was selected using the matching algorithm and then used to perform the crack detection.

For the sub-dataset generation and matching in experiment 2**,** the results were checked for different values of C_D (C_D = *0, 20,* and *40*) to select the appropriate threshold value. At $C_D = 20$, the best results were observed. Thus, the threshold value was set at $C_D \geq 20$ for experiment 2. The threshold value may change with a change in the images of the training dataset.

4.3 Results

Four performance metrics, namely, precision, recall, F-measure, and accuracy (Talukder et al., 2021; Baratloo et al. 2015), were calculated to compare the performances of experiments 1 and 2. Four types of data, i.e., true positives (TP), true negatives (TN), false negatives (FN), and false positives (FP), were used to calculate the values of the performance metrics, which are defined as follows:

$$\text{Precision} = TP / (TP + FP) \tag{5}$$

$$\text{Recall} = TP / (TP + FN) \tag{6}$$

$$\text{F-measure} = (2 \times \text{precision} \times \text{recall}) / (\text{precision} + \text{recall}) \tag{7}$$

$$\text{Accuracy} = (TP + TN) / (TP + FN + TN + FP). \tag{8}$$

where TP indicates a crack detected for a real crack image, FN indicates no crack detected for a real crack image, TN indicates no crack detected for a no-crack image, and FP indicates a crack detected for a no-crack image (Baratloo et al. 2015; Talukder et al., 2021).

When the crack detection method is accurate, the values of precision, recall, F-measure, and accuracy are all close to *1.0*, whereas they are almost *0.0* when the model is improper (Talukder et al., 2021; Baratloo et al. 2015).

The quantitative results of the experiments are presented in Tables 2, 3, and 4, which compare the performances of experiments 1 and 2.

Table 2: Results of experiment 1.

	Positive (True)	Negative (True)
Positive (Estimate)	TP 15	FP 3
Negative (Estimate)	FN 0	TN 12

Table 3: Results of experiment 2.

	Positive (True)	Negative (True)
Positive (Estimate)	TP 15	FP 1
Negative (Estimate)	FN 0	TN 14

Table 4: Comparison of the performance metrics of the experiments.

	Precision	Recall	F-measure	Accuracy
Experiment 1	0.833	1.0	0.908	0.900
Experiment 2	0.938	1.0	0.968	0.967

From Table 2, it can be observed that a total of three false results (FP) were obtained in experiment 1**.** In contrast, only one false result (FP) was obtained in experiment 2 as shown in Table 3.

Upon comparing the performance metrics of both experiments, it was observed that the values of the performance metrics improved in experiment 2 as compared to experiment 1, as shown in Table 4**.** The values of the performance metrics improved; namely, precision, F-measure, and accuracy improved by 93.8%, 96.8%, and 96.7% from 83.3%, 90.8%, and 90.0%, respectively. From these results, it is evident that the proposed method can improve the performance of crack detection in brick walls.

5 DISCUSSION

From the results of experiment 1 in Table 2, three false results were obtained because some brick structures appeared to have cracks. In some cases, the joining points of bricks also appeared as cracks. To clarify this, an example is shown in Figure 4 of an image taken from Kanagawa University, Japan. For this image type, experiment 1 provided false results; however, experiment 2 provided correct results for crack detection by selecting a proper sub-dataset.

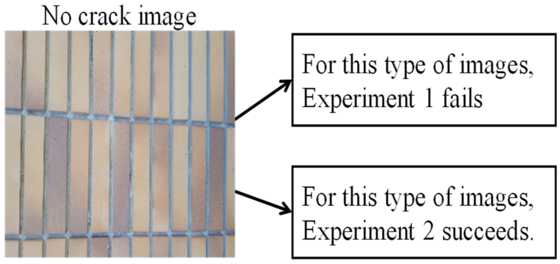

Figure 4: Example of image type for which experiment 1 fails.

For these types of images of brick walls of Figure 4, experiment 1 failed because the joining points of the bricks appeared as cracks; however, in reality, those were not cracks. The joining points of the bricks are dark and may appear similar to cracks. When using a large dataset for CNN training and for crack detection, then most similarity between the crack images of large training dataset and these types of images of Figure 4 was observed, rendering experiment 1 a failure.

Experiment 2 was successful for these types of brick walls, as shown in Figure 4, because sub-datasets were generated, using which the CNN learning was performed. For crack detection, the learned CNN was selected by matching the test image with the generated sub-datasets. The selected CNN was trained using the sub-dataset that contained only the images that were similar to the test image. Thus, experiment 2 was a success.

The advantage of the proposed method is that the values of the performance metrics are improved for the test images of brick walls. However, a limitation of the method is that the threshold value (C_D) used for the *Color Distance* parameter changes with a change in the images of the training dataset.

6 CONCLUSIONS

In this study, a new method consisting of sub-dataset generation and matching was proposed to improve the performance of CNN for the crack detection in brick walls. The proper learned CNN was selected for crack detection by matching the attributes of the sub-datasets used for learning with those of the test image. The results show that the proposed method improves the performance of crack detection in different types of brick walls.

In this study, the images of the training dataset were prepared manually, with 400 images being prepared for CNN learning. The dataset generation by manual process is laborious and time consuming. For this reason, manual dataset generation is difficult in industrial practices.

In future research, we plan to develop a systematic method for dataset preparation with a capacity to produce a large number of images (as high as 10,000 images) for CNN learning. In detail, we plan to develop datasets generation method not only for brick walls but also for concrete walls which will be used for the purpose of maintenance. Systematic method of datasets generation will reduce the required time for datasets generation as well as reduce the cost of maintenance.

REFERENCES

American Society of Civil Engineers (ASCE), (2017). Infrastructure Report Card.

Andrushia, A. D., Anand N., Godwin, I. A. (2018). Analysis of edge detection algorithms for concrete crack detection. *International journal of mechanical engineering and technology*. 9(11), 689–695.

Baratloo, A., Hosseini, M., Negida, A., Ashal, G. (2015). Simple definition and calculation of accuracy, sensitivity and specificity. *Emergency.* 3 (2), 48–49.

Bianconi, F., Harvey, R., Southam, P., Fernandez, A. (2011). Theoretical and experimental comparison of different approaches for colour texture classification. *Journal of electronic imaging*, 20 (4), 1–20.

Cha, Y. J., Choi, W. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Computer-aided civil and infrastructure engineering*. 32, 361–378.

Choi, D., Jeon, Y., Lee, S. J., Yun J. P., Kim, S. W. (2014). Algorithm for detecting seam cracks in steel plates using a Gabor filter combination method. *Applied Optics*. 53 (22), 4865–4872.

Dais, D., Bal, I., E., Smyrou, E., Sarhosis, V. (2021). Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. *Automation in construction*. 125, 1–18.

Dung, C. V., Anh, L. D. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in construction*. 99, 52–58.

Hoang, N. D., Nguyen, Q. L., Tran, V. D. (2018). Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network. *Automation in construction*, 94, 203–213.

Huyan, J., Li, W., Tighe, S., Zhai, J., Xu, Z., Chen, Y. (2019). Detection of sealed and unsealed cracks with complex backgrounds using deep convolutional neural network. *Automation in construction*. 107, 1–14.

Jacob, K., Mark, D. J., Peter, B., Mike, M., Gordon, M. (2019). A convolutional neural network for pavement surface crack segmentation using residual connections and attention gating. *2019 IEEE International Conference on Image Processing, ICIP.*

Li, G., Ma, B., He, S., Ren, X., Liu, Q. (2020). Automatic tunnel crack detection based on u-net and a convolutional neural network with alternately updated clique. *Sensors. 20*, 1–23.

Li, S., Zhao, X. (2019). Image-based concrete crack detection using convolutional neural network and exhaustive search technique. *Advances in civil engineering*. 2019, 1–12.

Liu, Z., Cao, Y., Wang, Y., Wang, W. (2019). Computer vision-based concrete crack detection using u-net fully convolutional networks. *Automation in construction.* 104,129–139.

Mahtab, M. K., Sahand, V., Leila, G., Ozturk, Y. E., Mustafa, Y., Selim, A., Nazim, K. U. (2019). Deep-learning-based crack detection with applications for the structural health monitoring of gas turbines. *Structural health monitoring*. 19 (5), 1440–1452.

Neogi, N., Mohanta, D. K., Dutta, P. K. (2014). Review of vision-based steel surface inspection systems. *Journal on image and video processing*. 50, 1–19.

Ozgenel, C. F., Sorguc, A. G. (2018). Performance comparison of pretrained convolutional neural networks on crack detection in buildings. *International Symposium on Automation and Robotics in Construction*, ISARC.

Qader, I. A., Abudayyeh, O., Kelly, M. (2003). Analysis of edge detection techniques for crack identification in bridges. *Journal of computing in civil engineering*. 17 (4), 255–263.

Road Bureau Japan (2015). Road maintenance in Japan: Problems and solutions. Ministry of land, infrastructure, transport and tourism, Roads in Japan.

Talukder, M. H.,Ota, S., Takanokura, M., Ishii, N. (2021). Crack detection in concrete structures under varied environmental conditions using CNN. *Journal of the society of plant engineers Japan*. 33 (1), 14–21.

Talukder, M. H., Ota, S., Takanokura, M., Ishii, N. (2020). Crack detection of concrete walls by CNN using sub-datasets. *The 2020 Spring National Conference of Operations Research Society of Japan*, 82–83.

Varma, M., Zisserman, A. (2005). A statistical approach to texture classification from single images. *International journal of computer vision*. 62 (1), 61–81.

Wu, X., Xu, K., Xu, J. (2008). Application of undecimated wavelet transform to surface defect detection of hot rolled steel plates. *2008 Congress on Image and Signal Processing*.

Yeum, C., Dyke, S. (2015). Vision-based automated crack detection for bridge inspection. *Computer aided civil infrastructure engineering*. 30, 759–770.