Point Cloud-Based Concrete Surface Defect Semantic Segmentation Using Modified PointNet++

Bolourian N., Hammad A., Ghelmani A.
Concordia University, Canada
amin.hammad@concordia.ca

Abstract. Structural inspection is essential to improve the safety and sustainability of infrastructure systems, such as bridges. Therefore, several technologies have been developed to detect defects automatically and accurately. For example, instead of using naked eye for bridge surface defect detection, which is subjective and risky, Light Detection and Ranging can collect high-quality 3D point clouds. This paper presents the Surface Normal Enhanced PointNet++ (SNEPointNet++), which is a modified version of the well-known PointNet++ method applied to the task of concrete surface defect detection. To this end, a point cloud dataset from three bridges in Montreal was collected, annotated, and classified into the three classes of cracks, spalls, and no-defects. Based on the comparison between the results (IoU) from the proposed method and similar research done on the same dataset, there are at least 54% and 13% performance improvements in detecting cracks and spalls, respectively.

1 Introduction

Structural inspection is essential to improve the sustainability and safety of infrastructure systems (e.g., bridges). Therefore, several technologies have been recently developed to detect defects automatically and accurately. Light Detection and Ranging (LiDAR) scanners have proven their benefits in identifying surface defects of bridges, such as cracks and spalls. Olsen et al. (2010) proposed a slicing analysis to quantify the spalling volume of large-scale structural elements based on the cross-sectional slices of the point cloud data. Other approaches based on damage-sensitive features such as curvature (Teza et al., 2009), distance and gradient (Liu et al., 2010), and surface normal (Guldur and Hajjar, 2017) have also been proposed. For localization and quantification of concrete surface spalls, Kim et al. (2015) introduced a new automated technique using Terrestrial Laser Scanner (TLS), which was applicable for defects larger than 4 mm in depth. Guldur et al. (2015) first used both graph-based and surface normal-based methods considering RGB and XYZ coordinates of points to detect the defects, and then applied clustering to group the defect points into individual defects. Those methods are sensitive to noise, uneven density, and complicated structures (Te et al., 2018).

Recently, the use of Deep Learning (DL)-based methods has been increasing in the construction industry. DL is applied for classification, semantic segmentation, and instance segmentation of point clouds. In terms of bridge inspection, DL classification aims to investigate the existence of defects without giving further information about their location or boundary. Compared to classification, semantic segmentation is more detailed requiring each point to be labeled individually and each class can be visualized with a specific color. Instance segmentation provides an even more detailed analysis by segmenting the points and assigning an ID to each instance in a class (Hafiz and Bhat, 2020).

Qi et al. (2017a) presented the first novel point-based method, called PointNet, to learn the features of each point using shared Multilayer Perceptrons (MLPs), which is a supplement of feedforward network, and global features using symmetrical pooling functions. However, PointNet has two main shortcomings: (1) lack of local context learning and (2) translation invariance limitation (Qi et al., 2017b). To overcome these limitations, in another work Qi et

https://doi.org/10.7146/aul.455.c227

al. (2017b) proposed PointNet++, which has hierarchical feature learning by using multiple vanilla PointNet learners with different scales.

This paper aims to take advantage of both point cloud data and DL-based semantic segmentation to detect concrete surface defects by applying a modified version of a well-known DL method, called PointNet++ (Qi et al., 2017b). The proposed method is called Surface Normal Enhanced PointNet++ (SNEPointNet++).

2 Methodology

PointNet++ architecture can be categorized into two main sections: (1) Sampling and grouping, (2) feature propagation. The goal of the sampling and grouping section is to address the two main limitations of lack of local context and translation invariance in the original PointNet paper. To this end, first a set of center points are sampled from input data using the furthest-point sampling algorithm, then neighboring points around each sampled center point are determined and grouped using query ball. While this approach includes the local context, since the density of point clouds is not unified in all parts of a real dataset, the resulting neighborhood may vary depending on the selected center points (translation variance). To address this problem, PointNet++ employs multi-scale and multi-resolution grouping. Finally, each center point and its corresponding neighborhood are input into the Vanilla PointNet to obtain local features. After obtaining the desired features, in the second section the features are propagated back into the original input size so as to obtain the label of each point (segmentation). To this end, interpolation layers are required, corresponding to the sampling and grouping layers used in the previous section. Furthermore, the output of each interpolation layer is first concatenated with the input of the corresponding sampling and grouping layer through skip connections, for improved performance, before being fed into a unit PointNet. Finally, the extracted features are passed through two fully connected PointNet units to obtain the label of each point.

Considering that PointNet++ semantic segmentation method was originally designed to detect indoor building elements, it cannot be applied in an off-the-shelf manner to the task of concrete bridge defect detection. As such, four main aspects differentiate this study from the original one, as explained below.

1. Creating a large high-quality point cloud dataset

A sufficient point cloud dataset is a key issue in the point-based semantic segmentation of surface defects. Unlike point clouds, many images of concrete cracks and spalls are available online, which can be used for training a DL model. Strict safety regulations, availability, and accessibility complications in scanning a bridge are the main reasons for the lack of point cloud datasets. Therefore, this study aims to provide a high-quality point cloud dataset to be used in surface defect detection by different groups of researchers. Furthermore, several data augmentation approaches such as shifting, flipping, and rotating can be applied to generate a bigger dataset based on the available one. In this work, the collected point clouds were tripled by flipping vertically and horizontally, however, rotation was not applied because it can alter the characteristics of some defects (e.g. orientation of shear cracks)

2. Redefining the attributes of points to better capture the main features of defects

Unlike the original PointNet++, finding two identical elements in one class is impossible, which causes some difficulties in the learning process. However, almost all cracks and spalls have two main characteristics; they are deeper and darker than their adjacent areas. Therefore, if trained properly, the network can learn to use these defining characteristics to distinguish between different types of defects and improve its performance.

This study proposes using normal vectors, as an input feature for training SNEPointNet++, in addition to the point location and color features, which are collected during scanning. The use of normal vectors for spall localization has also been suggested in other works such as Kim et al. (2015), which developed a methodology based on the changes in the depth and normal vectors of defects. Ideally, the normal vector of the no-defect area and the deepest point of a defect are perpendicular to the XZ plane, and the deviation between two adjacent normal vectors, identifying the potential presence of a defect. In the original PointNet++ work, the normalized coordinate values of (*X'*, *Y'*, *Z'*) are also used, which provide the network with the relative location of each point. However, since the changes in depth indicate defect existence and Y' dedicates the depth of each point, only this feature is employed in SNEPointNet++. It is computed based on Equation 1 (Qi et al., 2017a).

$$Y' = y_i / y_{max} \tag{1}$$

where y_{max} is the maximum value of y_i in each segment. As shown in Figure 1, a reference plane matching the damaged surface is used to calculate the depth of the points of the defect (y_i) with respect to that surface. It can be seen that the value of Y' increases with the depth of the defect. As such, in the SNEPointNet++ method, each point is represented by a 10-dimensional vector with values of *x, y, z, R, G, B,* N_x, N_y, N_z, and *Y'*.

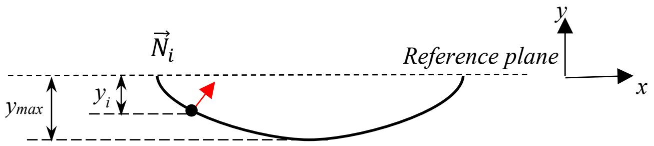


Figure 1: Cross Section of a Sample of Normal Vector and Depth of a Defect Point.

It should be noted that unlike the original PointNet++, in SNEPointNet++, 2D convolving for wrapping the dataset is performed on the XZ surface using the blocks with the given size.

3. Addressing the issue of imbalanced data for priority classes

The imbalanced class distribution is one of the main challenges in this research. Because not only the number of defects in each class but also their size are much different. Moreover, the largest part of the dataset belongs to the no defect class, which has the least priority. As a result, a weighted softmax cross-entropy loss function is applied to increase the contribution of the minority classes (spalls and cracks), which have priority.

4. Applying sensitivity analysis to adjust the hyperparameters in order to better capture the features of small defects.

The hyperparameters related to the network architecture and the dataset should be adjusted in case of using different datasets and attributes. The range of each hyperparameter is selected to cover the most promising values. The two main hyperparameters, which are considered in this paper are number of sub-layers and sampling size of each sublayer. Sub-layers in PointNet++ are responsible for extracting the features from the sampled and grouped points. The learning capacity of a neural network increases exponentially with its depth (i.e., number of sub-layers) and polynomialy with its width (i.e., number of nodes) (Montufar et al., 2014). Therefore, a sensitivity analysis on the number of sub-layers is performed to obtain the optimal performance.

On the other hand, multiple sampling sizes is one of the advantages of PointNet++. Deeper network makes the opportunity of increasing the variety of sampling sizes. Although all defects are small, their sizes vary in a wide range. Therefore, the sampling sizes should satisfy large spalls as well as small cracks.

The best combination of hyperparameters are selected based on the following three performance metrics, which are computed using Equations 2-4.

$$Precision = TP/(TP + FP) \quad (2)$$

$$Recall = TP/(TP + FN) \quad (3)$$

$$IoU = TP/(FP + TP + FN) \quad (4)$$

where TP, FN, TN, and FP represent True Positive, False Negative, True Negative, and False Positive, respectively.

3 Case Study

3.1 Data Collection and Preparation

The input of the proposed method, which is the 3D point cloud dataset, was scanned from four reinforced concrete bridges in Montreal using a FARO Focus3D scanner (FARO Technologies Inc., 2012). The 3D Faro laser scanner is equipped by a camera, which automatically captures images during scanning and detects the color of each point. CloudCompare (Girardeau-Montaut, 2020) is a 3D point cloud processing software. Afterwards, the scanned point clouds were registered and the irrelevant points (i.e., buildings, trees) were removed. Then, several parts were segmented out of the scanned bridge surface, and the *Z*-axis of the canonical coordinate system was set in the vertical direction. Normal vectors were calculated and three attributes (N_x, N_y, and N_z) were added to each point. Finally, the point cloud was manually annotated and labeled into three classes: cracks, spalls, and no-defect. The statistical information of the dataset is summarized in Table 1. Figure 2 demonstrates the distribution of the dataset based on segment density.

Table 1: Statistics of the Annotated Dataset Before Augmentation.

Dataset	**Number of segments**	**Number of points**	**Cracks**		**spalls**		**No-defect**
			Number of cracks	**Number of points**	**Number of spalls**	**Number of points**	**Number of points**
Training & Evaluation	81	21,313,285	475	182,430	588	1,252,551	19,878,304
Testing	21	5,628,620	120	64,269	185	682,614	4,881,737
Total	102	26,941,905	595	246,699	773	1,935,165	24,760,041

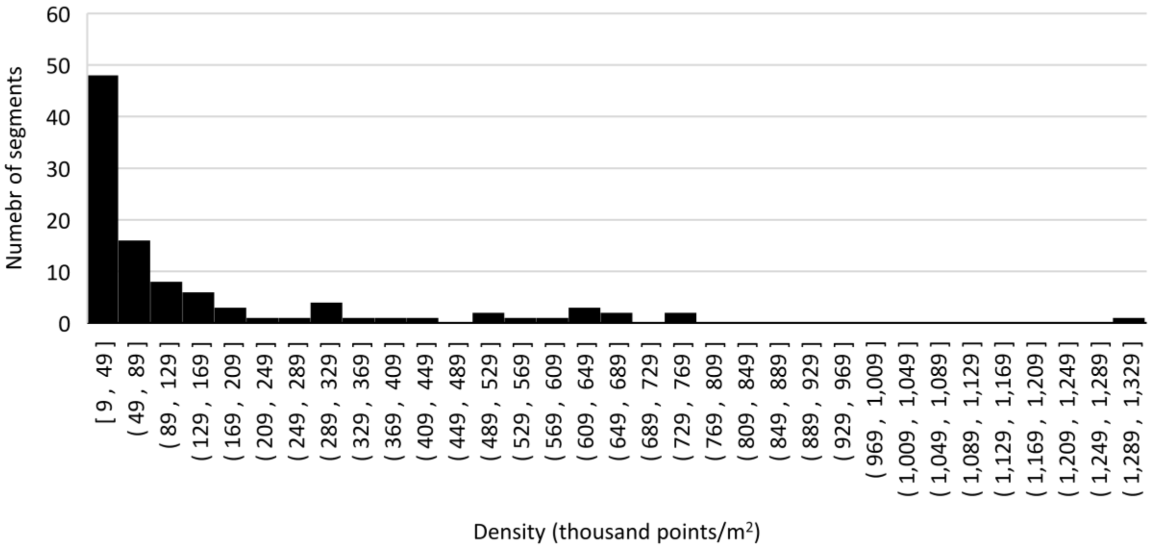
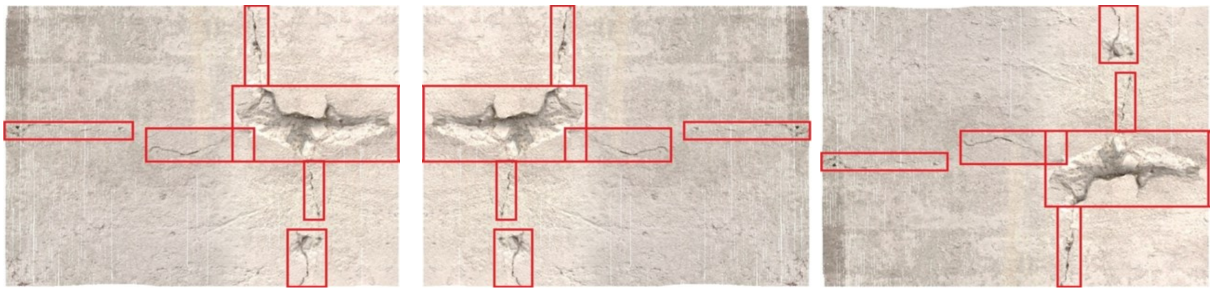


Figure 2: Distribution of Dataset Segments Based on Density.

The prepared dataset is further tripled in size by flipping the original dataset once horizontally and once vertically (Figure 3). The implementation and model training were performed using TensorFlow-GPU 1.15.1, python 3.6, and Cuda 11.0. The point cloud dataset attributes and the corresponding labels were concatenated and stored into *NumPy* format files. Then, the annotated point clouds were wrapped and normalized inside the blocks and saved in HDF5 format. The number of points in each block and block sizes variables can be adjusted in this step.



(a) Original annotated segment (b) Horizontally flipped segment (c) Vertically flipped segment

Figure 3: Data Augmentation.

3.2 Training and Testing

Data processing was performed on a LAMBDA workstation, with 3 NVIDIA RTX A6000 GPUs and an AMD Ryzer Threadripper 3960x 24-core CPU. Most of the algorithms are developed in Python 3.8. The number of batches and initial learning rate were assumed 24 and 1e-3, respectively. The learning rate decayed 50% every eight epochs until the minimum value of 1e-5 was reached.

The original network includes two sets of 4-layer networks. In this study, four, five, and six sublayers were considered to find the effective depth. The widths of defects vary between 0.2 cm (i.e., hairline cracks) and 50 cm (i.e., severe spalls). Therefore, a relatively wide range of 2.5 cm to 50 cm was considered for the sampling size of each sub-layer. The five sublayers with the sampling sizes of 2.5 cm, 5 cm, 10 cm, 20 cm, and 30 cm resulted in the most efficient network performance in terms of both crack and spall semantic segmentation. The testing results and the architecture of SNEPointNet++ are shown in Table 2 and Figure 4, respectively.

Table 2: Testing Results (%).

Sampling size in each layer (cm)					**Cracks**			**Spalls**		
1	**2**	**3**	**4**	**5**	**Precision**	**Recall**	**IoU**	**Precision**	**Recall**	**IoU**
2.5	5	10	20	30	73.3	93.0	69.2	89.9	92.0	82.5

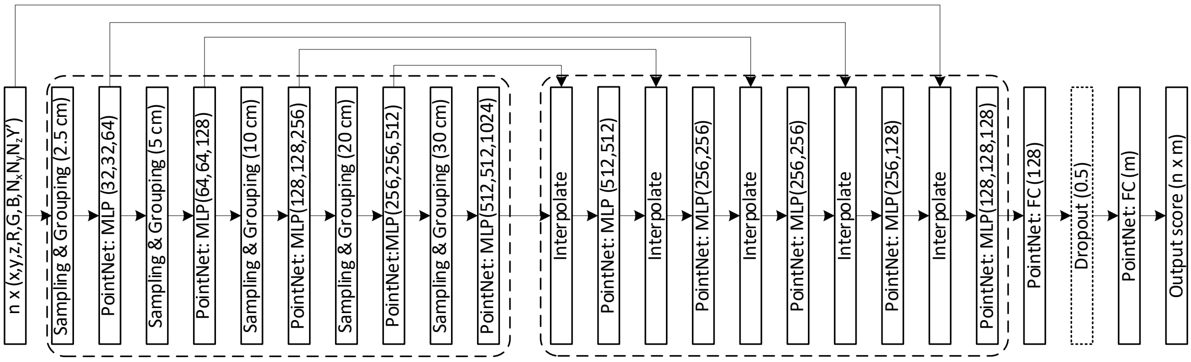


Figure 4: Architecture of SNEPointNet++.

4 Discussion

The recent image-based methods have reached around 98% (Le et al., 2021) to 99% (Vignesh et al., 2021) recalls in concrete surface defect classification. However, classification is not the appropriate approach to find the semantic information of each point individually, which is the objective of this research.

Compared to the recent image-based semantic segmentation methods, SNEPointNet++ results in 1-6% (Hoskere et al., 2020; Fu et al., 2021; Wang et al., 2022) and 1% higher IoU (Hoskere et al., 2020) in terms of crack and spall detection, respectively. Although Lee et al. (2019) trained a model with the highest precision in crack semantic segmentation, SNEPointNet++ leads to 19% higher recall.

To compare point cloud-based surface defect detection methods, they are categorized into three groups based on the types of defects (i.e., cracks, spalls, and both). The crack detection methods are incomparable because of using different scales (i.e., error) or visualization to show the results. Although the dataset of Valença et al. (2017) included the cracks with 0.1 mm to 4 mm thickness, the minimum detectable crack width was 1 mm. Turkan et al. (2018) proposed the only DL-based method in this category, which is not comparable due to showing the results based on error.

In the case of spall detection, the best result of Kim et al. (2017) was 92% average recall and 97 % average precision for ten samples with the sizes between 10 mm × 10 mm × 4 mm and 100 mm × 100 mm × 7 mm. Due to scanning larger spalls in an ideal situation in terms of incidence angle and distance, better results are expected.

The last group, which belongs to spall and crack detection methods, includes two studies. Unlike Guldur and Hajjar (2017) method, adapted DGCNN (Bahreini and Hammad, 2021) and SNEPointNet++ were evaluated using the same dataset. The IoU results show that the efficiency of SNEPointNet++ is 10% and 24% higher than adapted DGCNN in terms of spall and crack detection, respectively.

5 Conclusions and Future Work

This paper presents a DL-based automated method for detecting two types of concrete surface semantic segmentation, including cracks and spalls, using point clouds. The basic benefits of PointNet++ are: (1) considering the features related to both images (i.e., RGB) and point cloud (i.e., x, y, z); (2) not converting point clouds into other representations (i.e., voxels, images); (3) considering both local and global features; and (4) using multi-scale and multi-resolution samples. However, applying the original PointNet++ does not result in high performance, and fundamental modifications are required based on the nature of defects and collected dataset.

The contributions of this paper are: (1) Creating a large high-quality point cloud dataset, available upon request, for future research in concrete surface defect detection; and (2) Developing SNEPointNet++, which is a DL point-based semantic segmentation method for concrete surface defect detection. The proposed method is a modified version of PointNet++ to focus on two main characteristics of surface defects: normal vector and depth.

The proposed technique is successfully validated through a dataset of four bridges in Montreal, Canada, which was tripled by augmentation. This dataset includes 1,785 cracks and 2,319 spalls with a minimum width of 2 mm and 5 mm, respectively. SNEPointNet++ can detect cracks and spalls with 93% (IoU: 69%) and 92% (IoU:83%) recalls, respectively. It can be concluded that considering the nature, size, and patterns of the classes is vital to train a high-performance model. Using different settings for the scanning quality and resolution during the data collection, and having the segments with different densities make SNEPointNet++ invariant to these factors. However, a high-quality dataset is still required to have an accurate defect annotation.

As mentioned above, the lack of a huge dataset, which is crucial to train an accurate model, can be considered as the main limitation of this research. Moreover, the available dataset is limited to the surfaces of the bridge piers and abutments. Therefore, the model is not trained for curved surfaces. Despite using the weighted loss function for learning, the effect of the class-imbalanced dataset on the results cannot be fully eliminated. As future work, the dataset can be expanded by scanning more bridges, using other augmentation approaches, and generating synthetic point cloud datasets. Furthermore, instance segmentation can be applied to extract the individual defects. Future work could also define more classes based on the level of severity, such as small, medium, and severe spalls. In addition, future work will aim to investigate the effect of RGB on the performance of SNEPointNet++ by training the network without color feature and comparing the results with the network outputs of this paper.

References

Bahreini, F., Hammad, A. (2021). Point Cloud Semantic Segmentation of Concrete Surface Defects Using Dynamic Graph CNN. In: 38th International Symposium on Automation and Robotics in Construction. pp. 379–386.

FARO Technologies Inc. (2012). Faro Laser Scanner Focus 3D X 130 -NEO-Tech.

Fu, H., Meng, D., Li, W., Wang, Y. (2021). Bridge Crack Semantic Segmentation Based on Improved Deeplabv3+, J. Mar. Sci. Eng. 9, 671.

Guldur, B., Hajjar, J.F. (2017). Laser-based surface damage detection and quantification using predicted surface properties, Autom. Constr. 83, pp. 285–302. https://doi.org/10.1016/j.autcon.2017.08.004

Guldur, B., Yan, Y., Hajjar, J.F. (2015). Condition Assessment of Bridges Using Terrestrial Laser Scanners. In: Structures Congress. pp. 355–366.

Hafiz, A.M., Bhat, G.M. (2020). A survey on instance segmentation: state of the art., International journal of multimedia information retrieval 9, pp. 171–189. https://doi.org/doi.org/10.1007/s13735-020-00195-x

Hoskere, V., Narazaki, Y., Hoang, T.A., Spencer Jr., B.F. (2020). MaDnet: multi-task semantic segmentation of multiple types of structural materials and damage in images of civil infrastructure, J. Civ. Struct. Health Monit. 10, pp. 757–773.

Kim, M.K., Sohn, H., Chang, C.-C. (2015). Localization and Quantification of Concrete Spalling Defects Using Terrestrial Laser Scanning, J. Comput. Civ. Eng. 29, pp. 1–12. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000415

Le, T.-T., Nguyen, V.-H., Le, M.V. (2021). Development of Deep Learning Model for the Recognition of Cracks on Concrete Surfaces, Appl. Comput. Intell. Soft Comput., e8858545. https://doi.org/10.1155/2021/8858545

Lee, D., Kim, J., Lee, D. (2019). Robust concrete crack detection using deep learning-based semantic segmentation, Int. J. Aeronaut. Space Sci. 20, pp. 287–299.

Liu, W., Chen, S., Boyajian, D., Hauser, E. (2010). Application of 3D LiDAR scan of a bridge under static load testing, Mater. Eval. 68, pp. 1359–1367.

Montufar, G.F., Pascanu, R., Cho, K., Bengio, Y. (2014). On the number of linear regions of deep neural networks, Adv. Neural Inf. Process. Syst. 27, pp. 1–9.

Olsen, M.J., Kuester, F., Chang, B.J., Hutchinson, T.C. (2010). Terrestrial Laser Scanning-Based Structural Damage Assessment, J. Comput. Civ. Eng. 24, pp. 264–272. https://doi.org/10.1061/(asce)cp.1943-5487.0000028

Qi, C.R., Su, H., Mo, K., Guibas, L.J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Computer Vision and Pattern Recognition (CVPR). IEEE, Honolulu, HI, USA, pp. 4–23. https://doi.org/10.1109/CVPR.2017.16

Qi, C.R., Yi, L., Su, H., Guibas, L.J. (2017b). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, Adv. Neural Inf. Process. Syst. pp. 1–14.

Te, G., Hu, W., Zheng, A., Guo, Z. (2018). Rgcnn: Regularized graph CNN for point cloud segmentation. In: The 26th ACM International Conference on Multimedia. Seoul, Republic of Korea, pp. 746–754. https://doi.org/10.1145/3240508.3240621

Teza, G., Galgaro, A., Moro, F. (2009). Contactless recognition of concrete surface damage from laser scanning and curvature computation, Non-Destr. Test. Eval. Int. 42, pp. 240–249. https://doi.org/10.1016/j.ndteint.2008.10.009

Turkan, Y., Hong, J., Laflamme, S., Puri, N. (2018). Adaptive wavelet neural network for terrestrial laser scanner-based crack detection, Autom. Constr. 94, pp. 191–202. https://doi.org/10.1016/j.autcon.2018.06.017

Valença, J., Puente, I., Júlio, E., González-Jorge, H., Arias-Sánchez, P. (2017). Assessment of cracks on concrete bridges using image processing supported by laser scanning survey, Constr. Build. Mater. 146, pp. 668–678. https://doi.org/10.1016/j.conbuildmat.2017.04.096

Vignesh, R., Narenthiran, B., Manivannan, S., Arul Murugan, R., RajKumar, V. (2021). Concrete Bridge Crack Detection Using Convolutional Neural Network. In: Materials, Design, and Manufacturing for Sustainable Environment. Springer, pp. 797–812.

Wang, J., Liu, Y., Nie, X., Mo, Y.L. (2022). Deep convolutional neural networks for semantic segmentation of cracks, Struct. Control Health Monit. 29, e2850.