CSCE Annual Conference

Growing with youth – Croître avec les jeunes

Laval (Greater Montreal)

June 12 - 15, 2019

CONCRETE SURFACE DEFECT DETECTION USING DEEP NEURAL NETWORK BASED ON LIDAR SCANNING

Nasrollahi, M.[1,3], Bolourian, N.[1,4] and Hammad, A.[2,5]
[1] Department of Building, Civil, and Environmental Engineering, Concordia University, Canada
[2] Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Canada
[3] m_nasro@encs.concordia.ca
[4] n_bolour@encs.concordia.ca
[5] hammad@ciise.concordia.ca

Abstract: Structural inspection of bridges is essential to improve the safety of the infrastructure systems. Visual inspection is the principal method of detecting surface defects of bridges. In order to automate the process of structural inspection, it is important to collect proper datasets and use an efficient approach to analyze them and find the defects. Light Detection and Ranging (LiDAR) scanners can collect high-quality 3D point cloud datasets. Furthermore, Deep Neural Networks (DNNs) have been recently used for detecting 3D objects within 3D point clouds. This paper aims to develop a method for detecting concrete surface defects using a DNN based on LiDAR scanning. PointNet is a Convolutional Neural Network (CNN) used extensively for analyzing 3D point sets. PointNet has three parts: classification, part segmentation and semantic segmentation. The semantic segmentation part is originally designed to detect indoor building elements. In this paper, PointNet is adapted to detect surface defects using point cloud datasets from scanning bridge surfaces. The reason for selecting PointNet is that it is robust to missing and corrupted data. Training and testing datasets are collected from concrete bridges and annotated manually. Five point cloud datasets are prepared in five areas, which contain 3,572 annotated segments. They are classified into two classes, defect and non-defect. About 15% of the prepared point cloud data are assigned as a testing dataset. Promising initial results have been obtained in spite of the small size of the training dataset. In order to increase the accuracy, our future work aims to prepare a bigger dataset to train the model.

1 INTRODUCTION

Structural inspection of bridges is essential to improve the safety of infrastructure systems, such as bridges. Visual inspection is the traditional method of detecting surface defects of bridges. This process can be unsafe, time consuming, expensive and subjective to human errors (Guldur et al. 2015). In order to automate the process of structural inspection, it is important to collect proper datasets and use an efficient approach to analyze them and find the defects. Recently, 3D Light Detection and Ranging (LiDAR) scanners (Liu et al. 2011) are used for detecting surface defects (e.g. cracks) using computer vision methods. LiDAR scanners can collect high-quality 3D point cloud datasets.

The objective of this paper is to develop a method for detecting concrete surface defects using a Deep Neural Network (DNN) based on LiDAR scanning. The remaining contents of this paper are described as follows. Section 2 is a review of the main approaches of surface defect detection, specially the state-of-the-art Convolutional Neural Network (CNN). Section 3 provides the proposed methodology. Then,

Section 4 explains the case study, which includes the process of collecting and preparing the datasets, training the model, and testing the trained model. Section 5 has the conclusion and the future work.

2 LITERATURE REVIEW

There are two main approaches for detecting concert surface defects. The first is based on geometry analysis and the second is based on supervised machine learning. In the geometry analysis approach, volume losses were calculated using Gaussian curvature distribution (Teza et al. 2009) and crossing section method (Olsen et al. 2009). Armesto-Gonzalez et al. (2010) proposed an automated classification algorithm in order to detect moisture based defects. Girardeau-Montaut et al. (2005) detected volume changes in excavation using an octree-based comparison between average and Hausdorff distance. Liu et al. (2011) presented an algorithm to detect defects based on distance and gradient criteria of concrete bridge surface. Tsai et al. (2012) detected cracks in asphalt paving based on dynamic optimization and linear buffered Hausdorff scoring. Laefer et al. (2014) contributed a mathematical basis for using Terrestrial Laser Scanning (TLS) to detect cracks in unit-block masonry (i.e. stone, brick, or concrete masonry units). A new and automated technique that can simultaneously localize and quantify spalling defects on the concrete surface using TLS was proposed by Kim et al. (2014). Guldur et al. (2015) developed a strategy to define an appropriate threshold considering Red, Green, and Blue (RGB) color values of point clouds or intensity values.

Machine learning is a field of artificial intelligence, which gives computer systems the “learning” ability from data without explicit programming (Michalski, et al. 2013). Deep learning is one type of machine learning. In deep learning algorithms, first data are split into three non-equal datasets; the largest one is used for training, and the others are used for validation and testing. Several hyper-parameters are chosen for training, validation, and testing.

Convolutional Neural Network (CNN) is a class of deep forward networks, which is inspired by biological processes in the connectivity patterns between neurons (LeCun et al. 1995). A convolutional neural network (CNN) contains input, convolutional, subsampling and output layers; and each layer receives inputs by means of local receptive fields from the previous layer. Using local filters, neurons extract elementary geometric features, such as edges, boundaries, and corners. By convolving a filter using a unique weight vector over the entire image, a feature map is generated. Typically, a convolutional layer generates several feature maps in order to extract several features from the input image (LeCun et al. 1995). The process of multiplying a weight vector by the input data by convolving a kernel and creating linear activations in a feature map is the first stage of a convolutional neural network (Goodfellow et al. 2016). Nonlinear activation and subsampling the feature map by applying pooling functions are the other two stages. Pooling (subsampling) makes the model invariant to small geometric transformations or distortions of object’s features in the input (LeCun and Bengio 1995), which significantly improves the efficiency of the network (Goodfellow et al. 2016). Max pooling extracts the maximum elements of a rectangular neighborhood of feature map elements. Scherer et al. (2010) evaluated that max pooling works better than mean pooling in CNN for object detection.

The quality of data plays an important role in finding a proper fit function for any neural network. Therefore, collecting adequate datasets is necessary to achieve an accurate model. The datasets should represent the appropriate parameters and include different cases based on the requirements.

A point cloud is a 3D dataset which contains geometric information of sparse points captured from a 3D space. Point cloud datasets may also contain RGB and density information. Based on the 3D data representation, CNN approaches are classified into three main groups: pixel-based approaches, voxel-based approaches, and 3D point-based approaches. In pixel-based approaches (Su et al. 2015), 3D data are converted to 2D projection. Voxels are generated from the 3D points (Brock et al. 2016). In 3D point-based approaches, Qi et al. (2016) presented the very first novel approach to point cloud processing using 3D CNN and applying 3D recognition tasks containing object classification, part segmentation, and semantic segmentation.

Due to the popularity of the pixel- and voxel-based methods, 3D point cloud datasets are typically transformed into 3D-voxel grids or images in order to use in deep learning. The transformation result is voluminous data with unclear invariances in some cases. Moreover, learning from point cloud sets is easier than meshes due to their simplicity and unified structure. Meshes are complex and have combinatorial irregularities (Qi et al. 2017)

PointNet is a CNN which can be applied in classification, part segmentation, and semantic segmentation directly based on 4-Dimensional (4D) tensors data. This network was proposed in 2017 in order to prevent the problem related to voxelization and rendering point clouds (Qi et al. 2017). The input of PointNet is a subset of points, which has three main characteristics. First, these points are unordered. Interaction among neighbored points is the second important characteristic of these datasets. The points are not isolated and local structure of the combination of neighboring points affects the semantic information of the point sets. The third characteristic of point clouds is being invariant under transformation. These three characteristics are considered in designing the architecture of PointNet. Qi et al. (2017) used Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) (Armeni et al. 2016) to validate the semantic scene segmentation part of PointNet algorithm.

3 METHODOLOGY

Regular concrete surface defects on bridges are cracks, spalling, scaling, and delamination. Scaling is the flaking or peeling of the surface of concrete as a result of exposure of freezing and thawing. Delamination is separating the surface layer of concrete, which is not completely detached. The prolonged delamination until detaching causes spalling, and a spall is a detached part of a concrete mass (Ministry of Transportation 2008) . Table 1 shows the severity of defects, which is categorized into four levels of light, medium, severe, and very severe, based on the depth and area of the defects.

Table 1. Defects severity based on depth of loss (d), width and height of the affected area (w, h) (Ministry of Transportation 2008)

Type of defect	Light*	Medium*	Severe*	Very Severe*
Scaling	d < 5	5 ≤ d < 10	10 ≤ d < 20	20 ≤ d
Delamination	w, h < 150	150 ≤ w, h < 300	300 ≤ w, h < 600	600 ≤ w, h
Spalling	w, h < 150 d < 25	150 ≤ w, h < 300 25 ≤ d < 50	300 ≤ w, h < 600 50 ≤ d < 100	600 ≤ w, h 100 ≤ d

* All dimensions are in mm

This section explains the steps of applying the CNN approach on 3D point cloud datasets to recognize concrete surface defects. Moreover the details of the CNN architecture are explained in Section 3.4. There are five main steps in this method: (1) data collection, (2) manual annotation, (3) data pre-processing, (4) training and evaluation, and (5) testing (Figure 1).

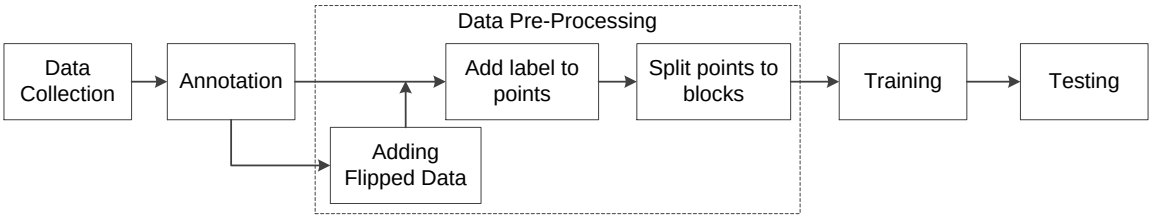


Figure 1. Proposed method

The semantic segmentation of PointNet is used in this method. This part is originally designed to detect indoor building elements. In this paper, the semantic segmentation of PointNet is adapted to detect surface defects using point cloud datasets from scanning bridge surfaces. PointNet is used in this work because it is robust to missing and corrupted data and directly applies on point datasets.

3.1 Data Collection

The process of accurate data collection is an important aspect of reaching valuable output results of a CNN algorithm. The geometric features of defects, especially the depth, play an important role in extracting useful features using the neural network. Terrestrial laser scanning provides high stability and accuracy compared to other methods of laser scanning (Nasrollahi et al. 2018). The position of the LiDAR scanner affects the visibility of defects in the collected point clouds. In addition, the fine registration process results in having reliable datasets to feed the CNN.

3.2 Annotation

In addition to collecting accurate datasets, the annotation process is vital for reaching a satisfactory CNN trained model. The annotation process is manual and is based on the structural definitions of concrete surface defects as shown in Table 1. The datasets are split into two main classes of defect and non-defect.

3.3 Data Pre-Processing

PointNet semantic segmentation is based on feeding the input data with the structure of S3DIS; so the training and testing datasets of this work are prepared in the same structure. Point clouds are distributed manually into different areas and each area has different parts. The *Z* axis of the canonical coordinate system is set in the vertical direction. The *X* axis is along the concrete surface of the bridge, and the *Y* axis is perpendicular to the concrete surface (inside direction). This setting gives the depths of the defects positive *Y* values. The convolutional process is 2D and performs on the *XZ* surface; so the 3D blocks of points are selected on this surface with the specified dimensions in the *X* and the *Z* directions, and depth of datasets as the third dimension. By choosing the dimensions and number of points of the 3D blocks, points are downsampled or upsampled as required, based on the density of the input datasets. The effect of changing the density on the results is discussed in Section 4.5.

In the data pre-processing step, the annotated point clouds are converted to Hierarchical Data Format (HDF). HDF is an abstract data managing and storing model (HDF-Group 2018). Points are wrapped and normalized inside the discussed blocks and saved in HDF5 format. In this process, three channels are added to each point's information. These channels are the normalized local values as to the split part of the whole point cloud. For each part, the point with the minimum coordinate values is set to the origin of a local coordinate system and the normalized values of all of the points are calculated as follows:

$$N_{x_i} = \frac{x_i}{X}\ , N_{y_i} = \frac{y_i}{Y}\ , N_{z_i} = \frac{z_i}{Z} \qquad \text{(Qi et al. 2017)}$$

where *X*, *Y*, and *Z* are the maximum values in the specific part. The normalized values are distributed from 0 to 1. By adding these values, each point is represented as a 9-dimensional vector of *XYZ*, *RGB*, and $N_xN_yN_z$. The base unit of the *XYZ* is meter with 3 precision integers. The RGB values are in the range of 0 to 255.

3.4 Training and Evaluation

The training model has two main parts: (1) classification network and (2) segmentation network, which are explained in the following sections. Three sets of Multi-Layer Perceptron (MLP) are used in these networks. All the hidden layers include batch normalization. Rectified Linear Unit (ReLU) activation function is used in all layers but the last one. The numbers of points in the two classes of datasets of this study are not equally balanced; the number of defect points is much less than the number of non-defect points, which is known as the issue of "unbalanced datasets". In a regular loss function, the effect of miss-predicting a defect point on the learning predictor vector is equal with respect to a non-defect classified point. By using a weighted loss function, the effective weight of points of each class on the correcting process of backpropagation can be adjusted. Weighted sparse softmax cross entropy loss function is used in this study, which is discussed in Section 4.4.

The points of each part in every area are automatically split into different blocks, which are fed into the CNN as input data. The number of points in each block in the original PointNet is 1,024 with block size of 1 m × 1 m for rooms of above 3 m high. This is assumed as a very low density of points for detecting most types of defects (e.g. Medium sized spalls). Therefore, the block size is selected to be less than 50 cm × 50 cm on the XZ surface, with the depth of the defects as the third dimension, which is less than 10 cm. Choosing the same number of points (i.e. 1,024) with the new block size can increase the density of points by more than 100 times.

3.4.1 Classification Network

Classification network has two sets of MLP. The first MLP accepts blocks of points as input and every layer extracts detailed features of points by convolving on the blocks. Every hidden layer includes batch normalization and ReLU activation function. The main goal of this set of MLP is to extract local features per point from the 9-dimensional input vectors of points. The output of the first MLP is a vector of all input points, where every point has a weight. This vector represents the extracted local features of points. A max-pooling layer is applied on the feature vector to down-sample the features, followed by the second set of MLP in order to extract global features of each point.

3.4.2 Segmentation Network

The third set of MLP is fed by concatenation of the extracted local and global features. Each convolutional layer in this MLP is followed by a dropout layer, except the last one. The output of this network is a vector of predicted probabilities of belonging to each class for every point.

3.5 Testing

The testing process uses a part of the datasets, which is not seen by the model in the training and the evaluation processes to validate the accuracy of the model.

4 CASE STUDY

In the case study, several datasets are collected using a terrestrial LiDAR, and annotated manually into the two main classes of defects and non-defects. In this study, all types of defects, regardless of the severity levels, are categorized as defects, in order to get the initial validation of the proposed CNN approach. Assuming the adopted CNN works well on detecting geometrical features of defects, it is hoped that having more detailed classes of defects, instead of only one defect class, may help the CNN train a more detailed model. However, the main aim of this study is validating the adopted CNN; and adding more classes based on the types of defects and their severity levels is part of our future work.

4.1 Data Collection

In CNN, a large dataset is required to train an accurate model. However, in this study, only three reinforced concrete bridges in Montreal are scanned using FARO Focus3D scanner (Faro 2012) in order to collect accurate point cloud datasets. The specifications of this LiDAR are presented in Table 2.

The scanning parameters, such as the fields of view (FoVs), resolution, quality, and number of scanned points are tabulated in Table 3. The scanned point clouds were imported into CloudCompare software (Girardeau-Montaut 2015) to register them and remove the irrelevant points, such as trees and buildings around the bridge.

Table 2. FARO Focus3D LiDAR scanner specifications (Faro 2012)

LiDAR	Points per Second	Field of View		Angular Resolution	Accuracy	Measurement Range
		Vertical	Horizontal			
FARO Focus 3D	976,000	305°	360°	0.009°	±2mm	1.5m - 120m

Table 3. Scanning information

	Scan 1	Scan 2	Scan 3
Horizontal FoV	23° to 259°	23° to 259 °	0° to 360°
Vertical FoV	-42.5° to 71°	-42.5° to 71°	-62.5° to 90°
Resolution	1/4	1/4	1/1
Quality	6x	6x	2x
Number of Points (Mpts)	25.5	25.5	699.1

4.2 Annotation

The point clouds are split into several parts considering the following: (1) The number of points in each segment must be in a specific range (between 150,000 pts to 400,000 pts); (2) Since the blocks have box shape, the scanned surfaces are divided into rectangular parts. (3) The size of each part should be big enough to contain different sizes of defects and small enough not to contain more points than the maximum number, especially in the areas with higher density of points. Moreover, annotating the defects based on their expected patterns leads to more effective learning. Annotation is done in CloudCompare software (Girardeau-Montaut 2015). The prepared datasets contain 55 parts from the scanned bridges that are categorized in 5 areas and contain more than 13.5 million points. 847 defects are annotated in the datasets, which contain 1.8 million points. The statistical information about the annotations is given in Table 4. Figure 2 shows a sample of an annotated segment.

Table 4. Statistics datasets used in training, evaluation, and testing processes

Datasets (%)	Area	Number of Parts	Number of points	Defects		
				Number	Number of points	Percentage
Training (74.9)	1	10	2,991,920	100	257,058	8.6
	1-Flipped	10	2,991,920	100	257,058	8.6
	2	12	3,077,769	91	180,842	5.9
	2-Flipped	12	3,077,769	91	180,842	5.9
	3	10	1,996,611	213	488,593	24.5
	3-Flipped	10	1,996,611	213	488,593	24.5
Evaluation (11.7)	4	12	2,522,790	238	581,791	23.1
Testing (13.4)	5	11	2,892,714	205	356,560	12.3
Total (100)		87	21,548,104	1,251	2,791,337	13.0

4.3 Data Pre-Processing

The annotated datasets are distributed in three categories for training, evaluation, and testing as shown in Table 4. The flipped point clouds of Areas 1 to 3 are added to the training datasets in order to enlarge them. The point clouds are flipped with respect to the YZ surface (Figure 2). The training annotated datasets are flipped using a python code and the flipped datasets are not used in the evaluating and testing. The proportion of evaluation and testing the data over the training data is experimental. In this research, almost 75% of the data were allocated for the training process.

The next steps of data pre-processing are adding the annotated labels (L) to the point's arrays and splitting them into blocks. After adding labels to points, the output files are 2D matrices, with each row having seven parameters (XYZRGBL) for each point (Qi et al. 2017). Every part is split into defined blocks (saved in HDF5 files) and is prepared for the CNN training process. The sizes of blocks of points are defined based on the sizes of structural defects discussed in Section 3 (Table 1).

Surface density of points varies on different parts of point clouds because of the change in angle of incidence. In order to solve this issue, PointNet normalizes the number of points in each block to a unique number. This process requires downsampling or upsampling.

4.4 Training and Evaluation

In order to find an appropriate model for detecting defects in point clouds, several models are trained and tested. A weighted softmax loss function is used and the corresponding weight vector of [0.86, 0.14] is defined based on the point's distribution in the classes. The base learning rate is set as 1e-3. It decreases using an exponential function every 8 epochs and the minimum value is 1e-5.

Adam and momentum optimizers are used for training. Based on the results, Adam optimizer has a better performance and is used for the rest of the analysis. Dropout layers are used in the network to improve the generalization ability of the trained model. A deeper network than the original PointNet semantic segmentation network is used in this study because the diagrams of training and evaluation accuracies showed the model reached the accuracy of 90% soon and got stuck there.

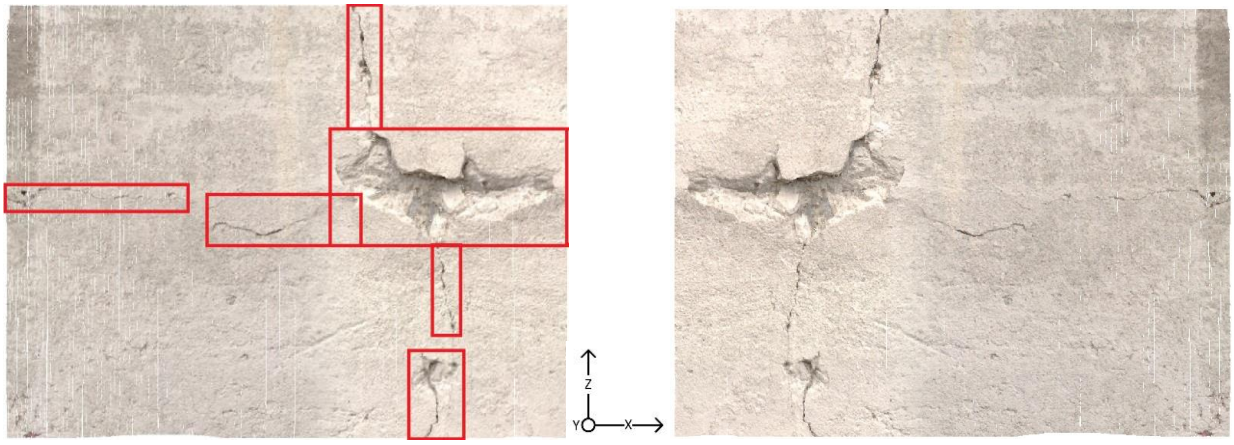


Figure 2. (a) Sample of annotated original points segment, (b) Flipped segment

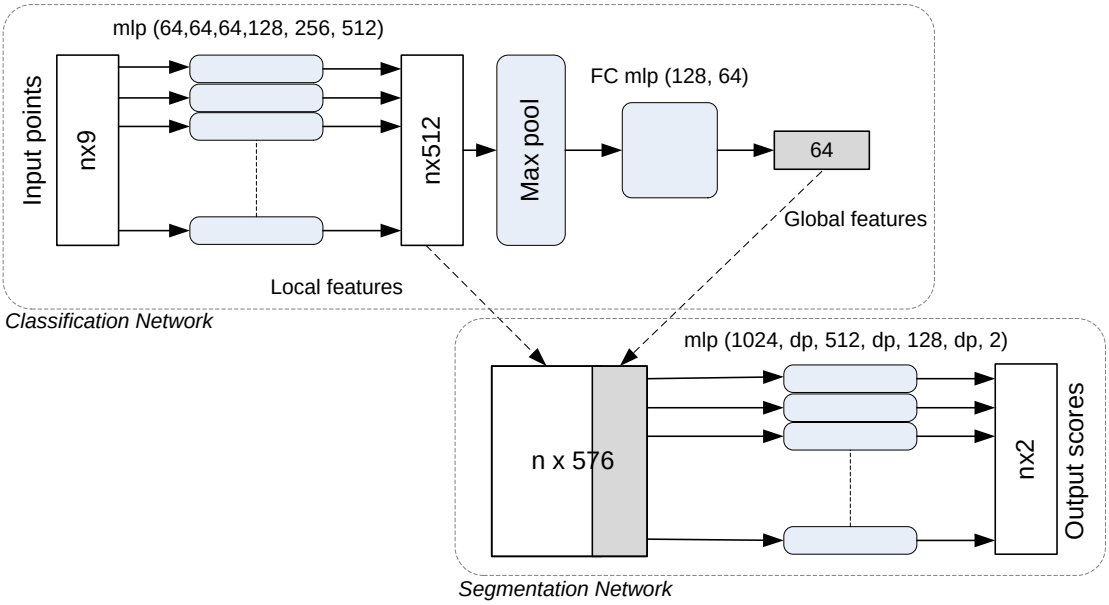


Figure 3. Modified PointNet network

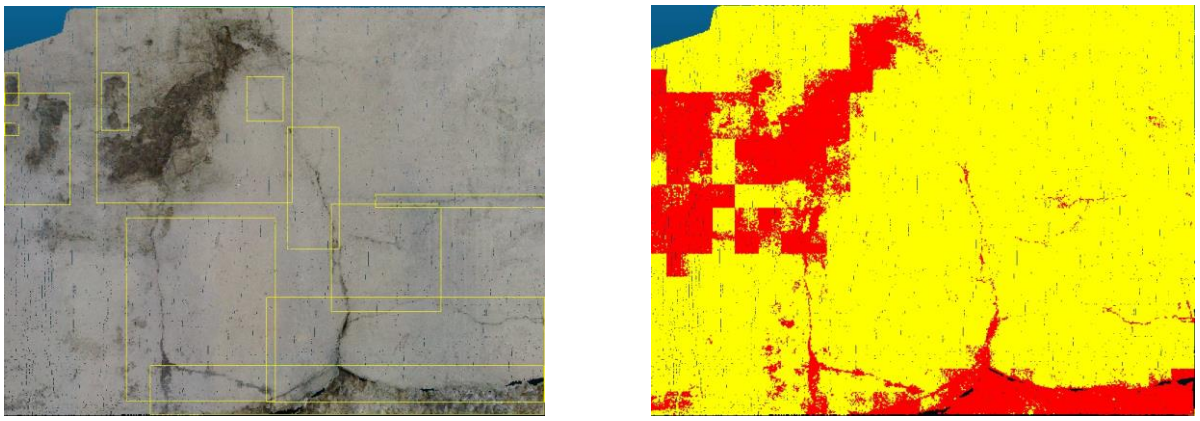
Several adjustments of the network model are applied and the best model is selected based on its performance on testing datasets. In the best model (Figure 3), the first MLP in the classification part has six hidden layers with numbers of neuron 64, 64, 64, 128, 256, and 512. The second MLP has two fully connected layers with numbers of neuron 128, and 64. The local features vector has 512 weights and the global features vector has 64 weights. By concatenating these vectors, an n×576 vector is fed to the segmentation part. The segmentation part of the network is changed to a pyramid neural network and the feature map of dimension n×576 expands to n×1024 to extract more features. This MLP has four convolutional hidden layers and three dropout layers. These modifications of the model increased the averaged overall accuracy of the testing to 86% and the averaged accuracy of defect detection to 66% as discussed in Section 4.5.

There are some hyper-parameters in the training network, such as epoch number, size of blocks, number of points in each block, number of batches for computing at the same time, filter stride size, and learning rate. The number of batches indicates how many blocks of points are analyzed for learning at the same time. A higher number of batches improves the performance of the model, but the available memory on the hardware constraints the possible volume of computation. A Compute Canada cluster is used to implement this case study. Two NVIDIA P100 Pascal GPUs, 32 CPUs, and 120 GB of memory are used for the training process.

4.5 Testing and Results

After training on the datasets, a model is generated, which contains predictor weight vectors for two classes of defect and non-defect. Obviously, the number of points in each block and the block size of the testing process have to be the same as those of the training process. The dataset of area 5 is used for testing the trained model. Figure 4 shows an example of the test outputs. This part is tested by a model with block size of 10 cm ×10 cm. The results of the testing are shown in Table 5. Eleven parts are used as the testing area. Each part has defects with different depths. The depth of the part's bounding box is assumed as the maximum depth of the segment's defects. The accuracies of the detected defects of the 11 parts are shown in Table 5. As expected, the detection results are better in the case of deeper defects.

To find out the impact of density changes on the results, 12 cases with different settings of block's point densities (between 0.6 and 42 pts/cm^2) are defined, trained and tested. Figure 5 shows the impact of the changes of the block densities on the accuracy of defects with respect to the minimum density. By increasing the density, first, the accuracy of the defects extremely fluctuates. After reaching about 11 pts/cm^2, a steady trend appears with 15% higher accuracy. Since the average point density of the raw dataset is 13 pts/cm^2, it can be concluded that increasing the block's density more than the density of the actual dataset (upsampling) does not have considerable effect on the results.



(a) (b)

Figure 4. (a) Annotated segments of the dataset (b) The output of testing (red parts indicate the points predicted as defects)

Table 5. Output results of the testing

Part	Number of defects	Number of points	Area (cm^2)	Depth (cm)	Mean loss	Accuracy	Accuracy of defects
1	30	327,777	6,248	1.5	0.157	0.839	0.315
2	16	195,140	5,957	1.6	0.327	0.858	0.149
3	5	153,353	18,490	2.6	0.034	0.944	0.618
4	24	486,062	6,607	2.8	0.143	0.759	0.605
5	10	166,446	19,338	3.5	0.127	0.897	0.523
6	15	166,667	16,152	3.7	0.146	0.863	0.352
7	19	184,671	20,104	4.1	0.263	0.734	0.584
8	7	369,296	21,459	5.0	0.059	0.911	0.498
9	5	291,776	13,756	5.1	0.053	0.909	0.648
10	13	259,111	42,174	6.0	0.096	0.827	0.867
11	16	278,342	47,597	8.1	0.097	0.858	0.875
Weighted average					0.116	0.857	0.659

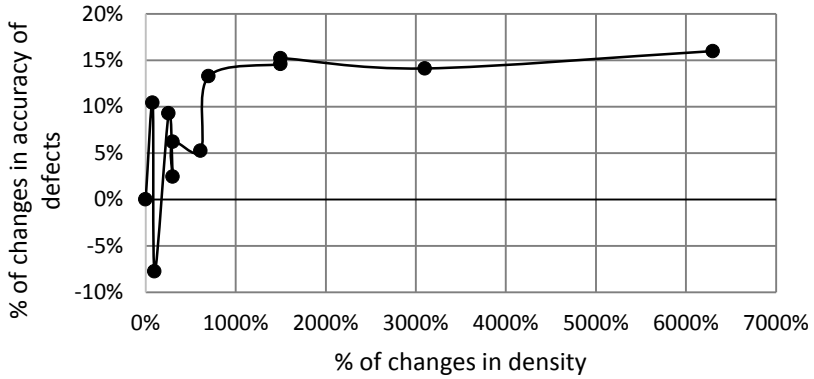


Figure 5. Effect of changing density on accuracy of defects

5 CONCLUSION AND FUTURE WORK

This paper proposed a method for detecting surface defects of concrete bridges using point clouds and DNN. The proposed method is based on PointNet, which is adapted to detect defects in LiDAR scanned datasets. Eight areas of the point cloud datasets (containing three flipped datasets) are used in training, evaluation, and testing. The following conclusions can be stated: (1) The trained model performed better in detecting the deeper defects, and (2) increasing the block's density more than the density of the actual dataset does not affect the results. There are two limitations in this study: (1) The training datasets are small, and (2) all kinds of defects are categorized in only one class. As the future work, preparing more annotated LiDAR scanned point clouds of bridges to expand the training datasets is expected to increase the accuracy of defect detection. In addition, by adding more classes to the training datasets, this method would perform better in detecting the specified defects.

References

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S. 2016. 3d semantic parsing of large-scale indoor spaces. *IEEE Conference on Computer Vision and Pattern Recognition.* Las Vegas Valley,1534-1543.

Armesto-González, J., Riveiro-Rodríguez, B., González-Aguilera, D., Rivas-Brea, M. T. 2010. Terrestrial laser scanning intensity data applied to damage detection for historical buildings. *Journal of Archaeological Science* **37**(12): 3037-3047.

Brock, A., Lim, T., Ritchie, J.M, Weston, N. 2016. Generative and discriminative voxel modeling with convolutional neural networks. arXiv: 1608.04236(2).

Faro. 2012. *FARO Laser Scanner Focus3D X130.* FARO Technologies Inc. https://www.faro.com/products/construction-bim-cim/faro-focus/.
Girardeau-Montaut, D. *CloudCompare.* 2015. http://www.cloudcompare.org/.
Girardeau-Montaut, D., Roux, M., Marc, R., Thibault, G. 2005. Change detection on points cloud data acquired with a ground laser scanner. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* **36**(3).
Goodfellow, I., Bengio, Y., Courville, A. 2016. *Deep learning*. MIT press, Cambridge, MA, USA.
Guldur, B., Yan, Y., Hajjar, J. 2015. Condition Assessment of Bridges Using Terrestrial Laser Scanners. *Structures Congress:* 355-366.
HDF-Group. 2018. HDF5 User's Guide. https://portal.hdfgroup.org/display/HDF5.
Kim, M. K., Sohn, H., Chang, C. C. 2014. Localization and quantification of concrete spalling defects using terrestrial laser scanning. *Journal of Computing in Civil Engineering*, **29**(6): 04014086.
Laefer, D. F., Truong-Hong, L., Carr, H., Singh, M. 2014. Crack detection limits in unit based masonry with terrestrial laser scanning. *NDT & E International*, **62**: 66-76.
LeCun, Y. and Bengio Y. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*. **3361**(10): 255-258.
Liu, W., Chen, S., Hauser, E. 2011. LiDAR-based bridge structure defect detection. *Experimental Techniques* **35**(6): 27-34.
Ministry of Transportation. 2008. Ontario Structure Inspection Manual (OSIM). St. Catharines, ON.
Nasrollahi, M., Bolourian, N., Zhu, Z., Hammad, A. 2018. Designing LiDAR-equipped UAV Platform for Structural Inspection. *International Symposium on Automation and Robotics in Construction (ISARC).* Berlin, Germany, **35**: 1092-1099
Olsen, M. J., Kuester, F., Chang, B. J., Hutchinson, T. C. 2009. Terrestrial Laser Scanning-Based Structural Damage Assessment. *Journal of Computing in Civil Engineering* **24**(3): 264-272.
Qi, C. R., Su, H., Mo, K., Guibas, L. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Computer Vision and Pattern Recognition (CVPR).* Honolulu, HI, USA,: 77-85.
Qi, C. R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L. J. 2016. Volumetric and multi-view CNNs for object classification on 3D data. *IEEE conference on computer vision and pattern recognition.*: 5648-5656.
Scherer, D., Müller, A., Behnke, S. 2010. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. *International Conference on Artificial Neural Networks.* Thessaloniki, Greece.: 92-101
Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E. 2015. Multi-view convolutional neural networks for 3D shape recognition. *IEEE international conference on computer vision.* Santiago, Chile,: 945-953.
Tsai, Y. J. and Li, F. 2012. Critical assessment of detecting asphalt pavement cracks under different lighting and low intensity contrast conditions using emerging 3D laser technology. *Journal of Transportation Engineering,* **138**(5): 649-656.