Semantic Image Retrieval and Clustering for Supporting Domain-Specific Bridge Component and Defect Classification

Peter Cheng-Yang Liu[1] and Nora El-Gohary, A.M.ASCE[2]

[1]Graduate Student, Dept. of Civil and Environmental Engineering, Univ. of Illinois at Urbana-Champaign, Urbana, IL. E-mail: cyliu4@illinois.edu
[2]Associate Professor, Dept. of Civil and Environmental Engineering, Univ. of Illinois at Urbana-Champaign, Urbana, IL. E-mail: gohary@illinois.edu

ABSTRACT

Automatic defect detection and classification from images is becoming increasingly important for bridge deterioration prediction and maintenance decision making. The majority of existing defect detection efforts have developed their datasets for training a machine-learning algorithm for detection/classification. However, the majority of these datasets suffer from two main limitations. First, most of the datasets are relatively small in size, which is not sufficient to build a well-trained, accurate image classifier. Second, most of the datasets lack the needed variety in scenes, angles, and backgrounds, which is not adaptable to different application contexts and environments. To address these limitations, this paper proposes a semantic image retrieval and clustering method to collect a large size of relevant images with various scenes, angles, and backgrounds from the Web and cluster these images for supporting domain-specific bridge component and defect detection. The proposed method includes three primary steps: query formation and image search and retrieval, image representation, and image clustering. First, a set of domain-specific words were extracted from bridge inspection documents and used as queries for retrieving a large number of images from the Web. Second, a transfer learning technique was used to transfer knowledge in a pre-trained model for general image classification to the bridge component and defect-related image clustering task. A deep convolutional neural network (CNN) with pre-trained weights was used to extract the visual features of the images for image representation. Third, a clustering technique was used to cluster the images based on the extracted features. The performance of the proposed method was evaluated using the silhouette coefficient. The evaluation results show that the proposed method is promising.

INTRODUCTION

According to the American Society of Civil Engineers, America's bridges received a C+ grade in 2017 nationwide report card (ASCE 2017). The report shows that over 9 percent of the nation's bridges, 54,000 bridges, are structurally deficient. Visual inspection is still the primary technique used for bridge inspection. For example, the Long-Term Bridge Performance (LTBP) program lists several visual inspection methods for assessing bridge condition and bridge performance, such as steel and concrete elements (Hooks and Weidner 2016). However, manual inspection poses serious safety risks. For example, in 2017, transportation incidents caused more than 2,000 fatal occupational injuries. Moreover, more than 800 workers fell, slipped, and/or tripped during their duties on the jobsites (Bureau of Labor Statistics 2019). Besides, the current inspection practices are costly. The average cost for bridge inspection ranges from $1,000 to $10,000 per bridge (Hong et al. 2012). Hence, automatic visual bridge maintenance approaches are needed to reduce safety risks and inspection costs.

The state-of-the-art automatic visual recognition approaches use supervised deep

convolutional neural networks (CNN). Despite the good performance results of these approaches, they require large-scale annotated training datasets. For example, the ImageNet (Russakovsky et al. 2015), a CNN trained by a fully-annotated dataset, has achieved the best performance in the ImageNet Large Scale Visual Recognition Competition (Russakovsky et al. 2015) since 2014 (Krizhevsky et al. 2012). However, the currently available bridge component and defect-related image datasets are neither large nor labeled.

Recent research efforts (e.g., Sharif Razavian et al. 2014; Yosinski et al. 2014) tried to solve such training data limitations by transfer learning. Transfer learning is a machine-learning technique that aims to reuse the knowledge (weights or features) gained from solving a problem and apply it to a different one. Existing research efforts (e.g., Sharif Razavian et al. 2014; Yosinski et al. 2014) showed that CNN approaches could outperform other approaches, even with a relatively small number of labels, by transferring the model weights learned to the other large datasets.

In addition to the aforementioned limitations of data size and labeling, two gaps exist in the available bridge component and defect-related image datasets. First, existing datasets lack a sufficient number of object classes. Most of the available datasets were used in detecting or classifying less than ten types of bridge components or defects (Koch et al. 2015; Narazaki et al. 2018). Sufficient bridge component and defect categories are required to train a robust classifier. Second, existing datasets lack variety in scenes, angles, and backgrounds, because of the fixed current image acquisition platforms (e.g., camera poses). Training images should cover enough variability, so the trained methods could be adapted to various unseen environments.

To address these limitations, this paper proposes a content-based image retrieval and clustering approach to retrieve a large size of relevant images, cluster similar images, and remove non-related images. The proposed approach includes four primary steps: query formation and image search and retrieval, feature extraction and image representation using transfer learning, image clustering and outlier removal, and evaluation. First, a set of domain-specific words were extracted from bridge inspection documents and used as queries for retrieving a large number of images from the Web. Second, a deep CNN model pre-trained with a large-scale annotated dataset, the ImageNet, was used to transfer the visual knowledge and extract feature vectors for image representation. Third, the images were clustered, based on their feature vectors, using a clustering algorithm. Fourth, the silhouette coefficient was used to evaluate the clustering performance.

LITERATURE REVIEW

Deep CNN Methods

Recently, deep CNN approaches showed noticeable results on several computer vision tasks. For object classification problems, the Visual Geometry Group (VGG) networks (Simonyan and Zisserman 2014) designed simple convolutional blocks and improved the classification accuracy by building a deeper CNN architecture. The following methods achieved state-of-the-art performances by modifying the CNN architectures (He et al. 2016 and Szegedy et al. 2016): Visual Geometry Group (VGG 16 and VGG 19), Residual Network (ResNet 50) (He et al. 2016), and Inception Network (Inspection v3) (Szegedy et al. 2016).

VGG16 and VGG19: The VGG networks introduce simple convolutional blocks to show that the accuracy could dramatically increase by making the CNN architectures deeper. A convolutional block uses 3x3 convolutional filters for each layer and stacks layers on top of each

other to increase depth. Each block ends up with a max-pooling layer to reduce volume size. The CNNs use fully-connected layers at the end of the networks for classification. The difference between the VGG16 and the VGG19 is that the VGG16 has 13 convolutional layers, whereas the VGG19 has 16 convolutional layers.

ResNet50: The residual network aims to solve the degradation problem in deep CNN learning. Contrary to intuition, the experiments showed worse performance results when CNN architectures stack "deeper" to 56 layers compared to 20 layers (He et al. 2016). The ResNet solves the problem by learning the residual of a function instead of the function itself. Learning the residuals is easier, because the residuals of a function have less content than the function itself. The architecture is similar to the VGG networks, but it adds a short connection within each block for residual learning. Traditionally, given an input, x, CNN models try to learn the objective function, $F(x)$. The residual learning, on the other hand, tries to learn the residual function, $g(x)$, by adding input to the end of the residual block. The objective function becomes:

$$F(x) = g(x) + x.$$

Inception V3: the goal of the inception network is to reduce the redundant learning in a deep CNN network. The inception network can achieve similar performance compared to previous networks with fewer computations by introducing the inception block, a multi-level feature extractor. The inception blocks are composed of 1x1, 3x3, and 5x5 convolution filters. The outputs of these filters are concatenated together and then sent to the next layer.

Deep CNN Datasets and Transfer Learning

An essential requirement for deep CNN approaches is large-scale annotated training datasets. Outside of the civil infrastructure domain, the ImageNet dataset (Russakovsky et al. 2015) contains more than 14 million images with 1,000 classes labeled for classification problems; and the COCO dataset (Lin et al. 2014) provides more than 330 thousand images with 1.5 million object instances, 80 object categories, and 91 stuff categories labeled for object detection and segmentation tasks. The insufficiency of datasets in several applications led to the adoption of transfer learning to leverage the features, which were learned from deep CNN models trained on large-scale annotated datasets, for image representation (Yosinski et al. 2014). Deep CNN models trained with large-scale annotated datasets could be directly used as feature extractors (Sharif Razavian et al. 2014) or used for fine-tuning the networks (Girshick et al. 2014). Also, previous research efforts showed high performance results for unsupervised learning by transferring the knowledge learned from solving one task, using one large-scale annotated dataset, to another task (Guérin et al. 2017).

Civil Infrastructure Component and Defect Detection

Civil infrastructure component and defect detection using vision-based methods has attracted a lot of research attention in the past decades. Most computer vision approaches focused on crack detection (Koch et al. 2015). Traditional image processing techniques tended to use hand-engineered visual features, such as edge detectors (Abdel-Qader et al. 2003). However, the performances of hand-engineered features were not reliable when dealing with noise in the images, such as brightness. More recently, deep machine learning-based approaches for extracting visual features showed significant performance improvement (Zhang et al. 2016). Crack detection was also extended to other types of defects, such as pop-outs, spalling, and exposed rebars (Koch et al. 2015). On the other hand, bridge component detection has been

studied (Narazaki et al. 2018) but with less attention. In most of these efforts, deep supervised machine learning-based approaches were used to reach stable performance levels.

Several large-size annotated datasets for civil infrastructure defect detection tasks exist. For example, the German Asphalt Pavement Distress Dataset (Eisenbach et al. 2017) includes 1,969 pavement images with full labeling with six pavement distress types. Also, the Road Damage Dataset (Maeda et al. 2018) contains 9,053 road damage images with 15,435 instances of road surface damage annotations. However, the majority of existing datasets have insufficient variability in terms of including various backgrounds and different camera poses, because the images were taken in a fixed-setting or from a close look at the surface. In addition, overall, there is a lack of bridge-specific component and defect datasets.

METHODOLOGY

A semantic image retrieval and clustering method for supporting domain-specific bridge component and defect detection is proposed. The proposed method aims to collect and classify a large size of relevant images with various scenes, angles, and backgrounds from the Web. The retrieved and clustered images aim to serve as pseudo training data for supporting machine learning-based bridge component and defect detection. The proposed method includes four primary steps (as per Figure 1): (1) query formation and image search and retrieval: a set of domain-specific words was extracted from bridge inspection documents, the LTBP Program Protocols, and used as queries for retrieving a large number of images from the Web; (2) image representation: using a transfer learning approach, a deep CNN model pre-trained on a large-scale annotated dataset, the ImageNet (Russakovsky et al. 2015), was adapted and used to extract the features of the retrieved images, generating feature vectors for image representation; (3) image clustering: the images were clustered, based on their feature vectors, using clustering algorithms; and (4) evaluation.

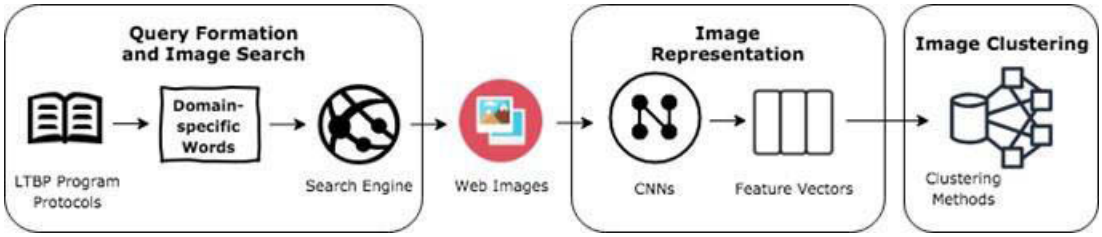

Figure 1. Proposed image retrieval and clustering method

Query Formation and Image Search and Retrieval

The most frequent bridge component and defect-related words were extracted from a set of domain-specific documents, the LTBP Program Protocols, to formulate queries for retrieving bridge component and defect-related images from the Web. A corpus of 3,706 unique words with their frequency counts was collected from the protocols after stop-word removal. By analyzing word frequency, the top 20 bridge component and top 20 bridge defect words were extracted (examples are shown in Table 1). In order to retrieve more relevant results, the domain-specific words were concatenated to the word, "bridge", forming the search queries. The queries were then used to retrieve the images from the Web, using the Google Image search. A set of 50 images per query were retrieved, resulting in a total of 1,848 images.

Table 1. Top 5 Bridge Component and Top 5 Bridge Defect Words.

Bridge Component Word	Frequency Count	Bridge Defect Word	Frequency Count
Bridge	777	Defect	281
Deck	341	Crack	164
Girder	157	Corrosion	102
Bearing	132	Delamination	54
Surface	131	Deterioration	28

Feature Extraction and Image Representation Using Transfer Learning

Using a transfer learning approach, a pre-trained deep CNN model was adapted and used to extract the visual features of the retrieved images for image representation. The model was pre-trained on an existing large-scale annotated dataset, the ImageNet (Russakovsky et al. 2015). The model is composed of three parts: convolutional blocks, fully-connected layers, and a softmax layer. To transfer the knowledge (i.e., features, weights) from the pre-trained model to the new domain-specific model, the pre-trained weights were used to initialize the convolutional blocks and fully-connected layers. Since the softmax layer was designed for classification not clustering, it was discarded in this study. The adapted model generates two layers of features/vectors from the retrieved (domain-specific) images for image representation. Four types of deep CNN models were tested and evaluated: the VGG16 and VGG19, Inception v3, and ResNet 50. These four models were selected because they show state-of-the-art performance in several computer-vision tasks. The sizes of the layers are shown in Table 2. For implementation, Keras (Chollet 2015) 2.2.0 with Tensorflow 1.9.0 (Abadi et al. 2015) libraries were used.

Table 2. Length for Each CNN Layer.

	VGG16	VGG19	ResNet50	Inception v3
Last Layer of Convolutional Blocks	25,088	25,088	51,200	100,352
Last Fully-connected Layer	1,000	1,000	1,000	1,000

Image Clustering and Outlier Removal

Image clustering aimed to identify and classify images that have the same bridge components and parts, and remove non-related images. Two clustering algorithms, K-means and agglomerative hierarchical clustering, were tested and evaluated. These two algorithms were selected because they have been commonly used in the existing literature for image clustering (Radford et al. 2015). For selecting the number of clusters, a set of cluster numbers were chosen to initialize the algorithm, as further discussed in the Experimental Results and Discussion Section. The cluster numbers, ranging from 2 to 50, were designed to cover bridge component and defect classes and non-related image classes. The cluster centers, *c*, were calculated by averaging the image features, f_i, within the clusters, where *i* is the order of features within a cluster. The Euclidean distance, $d_i = \|c - f_i\|$, from each feature point to its cluster center was then calculated and ranked within each cluster. The outliers were determined as the feature points that their distance ranking is larger than the ranking order, *r*. A set of experiments were conducted to identify the relationship between the ranking order, *r*, and the clustering

performance. To examine the quality of the dataset, the silhouette coefficients were calculated after removing outliers on different ranking orders. The total number of images were then counted to examine the quantity of the dataset after removing outliers on different ranking orders.

Table 3. Silhouette Coefficient Results on Two Clustering Algorithms, (1) K-Means And (2) Agglomerative Hierarchical Clustering Using Different CNN Models as Image Representation.

Cluster	Feature Layer	VGG16		VGG19		ResNet50		Inception v3	
		(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
10	conv.	0.044	0.040	0.037	0.035	0.010	0.014	0.003	0.004
	fully	**0.134**	**0.118**	**0.141**	**0.142**	**0.184**	**0.158**	**0.207**	**0.291**
15	conv.	0.028	0.060	0.034	0.044	0.017	0.014	0.020	0.004
	fully	**0.164**	**0.144**	**0.164**	**0.165**	**0.182**	**0.185**	**0.294**	**0.286**
20	conv.	0.040	0.046	0.039	0.034	0.014	0.011	0.019	0.012
	fully	**0.123**	**0.150**	**0.166**	**0.182**	**0.185**	**0.186**	**0.266**	**0.259**
25	conv.	0.046	0.037	0.032	0.066	0.018	0.017	0.008	0.013
	fully	**0.162**	**0.164**	**0.166**	**0.161**	**0.198**	**0.214**	**0.277**	**0.287**
30	conv.	0.042	0.042	0.058	0.035	0.025	0.009	0.020	0.019
	fully	**0.173**	**0.202**	**0.181**	**0.191**	**0.219**	**0.205**	**0.298**	**0.296**
35	conv.	0.039	0.044	0.035	0.039	0.031	0.017	0.017	0.013
	fully	**0.157**	**0.175**	**0.177**	**0.183**	**0.206**	**0.194**	**0.318**	**0.317**
40	conv.	0.102	0.041	0.073	0.029	0.019	0.022	0.018	0.015
	fully	**0.172**	**0.173**	**0.163**	**0.199**	**0.215**	**0.214**	**0.303**	**0.320**
45	conv.	0.084	0.087	0.048	0.063	0.025	0.017	0.020	0.019
	fully	**0.202**	**0.172**	**0.184**	**0.185**	**0.214**	**0.213**	**0.337**	**0.287**
50	conv.	0.059	0.039	0.071	0.037	0.024	0.017	0.023	0.020
	fully	**0.179**	**0.171**	**0.190**	**0.183**	**0.215**	**0.189**	**0.321**	**0.330**

Evaluation

The clustering performance was evaluated based on the average silhouette coefficient (Rousseeuw 1987). The average silhouette coefficient is the average of the silhouette coefficients of all the data in a dataset. The silhouette coefficient for a sample is defined as per Eq. 1, where *a* is the mean intra-cluster distance, which refers to the average distance of the sample with all other data in the same cluster, and *b* is the mean nearest-cluster distance, which refers to the average distance of the sample with all other data in the closest cluster. The silhouette coefficient ranges from -1 to 1. A value near 1 indicates a good clustering performance, i.e., that the samples are far from their closest neighboring clusters and close to each other within their clusters.

$$s = \frac{b - a}{\max(a, b)} \tag{1}$$

EXPERIMENTAL RESULTS AND DISCUSSION

The silhouette coefficient of the retrieved dataset is 0.63, which indicates that the pre-trained CNN models, as feature extractors, are suitable for clustering unlabeled bridge component and

defect-related Web images. A series of experiments were conducted to test the performances of different feature extraction models, different image representation layers, and different clustering techniques. The experimental results also indicate that the proposed method could increase the clustering quality after removing outliers while maintaining the quantity of the dataset.

Feature Extraction, Image Representation, and Clustering Results

The clustering results, using the four adapted CNN models, are summarized in Table 3. The K-means clustering algorithm showed the highest silhouette coefficient using the last fully-connected layer of Inception v3 as features. Overall, using the last fully-connected layer as features outperformed using the last layer of convolutional blocks. The reason might be that the last fully-connected layer captures higher-level features, such as objects; whereas, the last layer of convolutional blocks contains lower-level features. These results indicate that complex architectures, Inception v3 and ResNet50, extracted better features for clustering bridge component and defect-related images than simple sequential CNNs, the VGGs. The results also indicate that the proposed image representation is suitable for clustering multiple classes with more clusters.

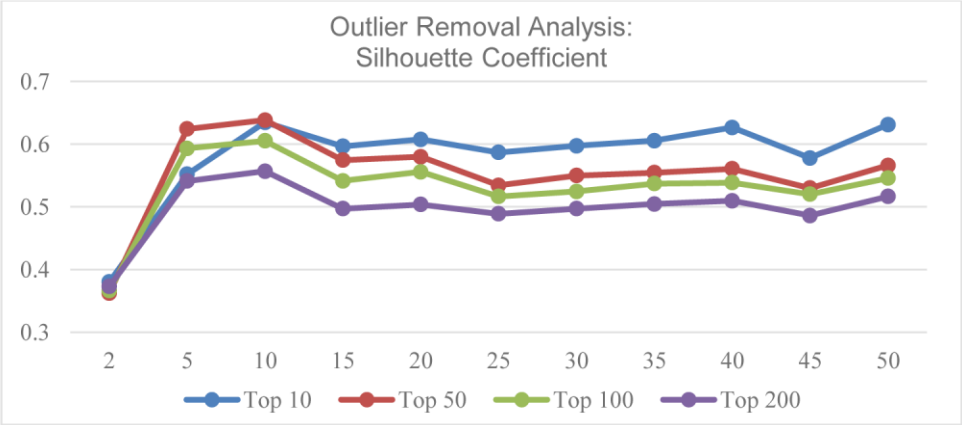

Figure 2. Clustering results after removing outliers on different ranking orders

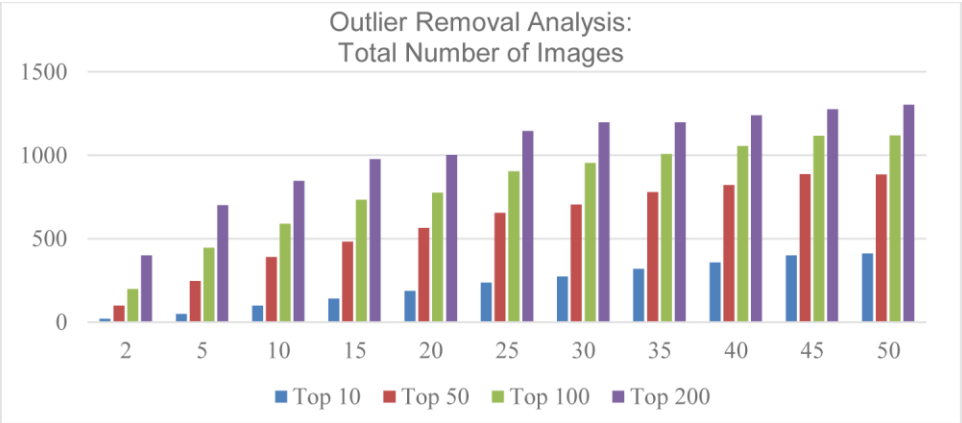

Figure 3: Number of images after removing outliers on different ranking orders

Outlier Removal Analysis

The results of outlier removal analysis are illustrated in Figures 2 and 3, using a ranking

order of 10, 50, 100, and 200. The results show that removing outliers out of the top 10 image features could achieve 0.63 silhouette coefficient. The performances were boosted with silhouette coefficients achieving 0.5 in most of the outlier removal conditions. After removing the outliers, the proposed method was able to collect a large number of images while increasing the quality of clustering. More than 800 images were collected using a ranking order of 50, and more than 1,200 images were collected using a ranking order of 200.

CONCLUSION AND FUTURE WORK

In this paper, the authors proposed a domain-specific image retrieval and clustering method to retrieve and cluster a large set of bridge component and defect-related images from the Web. The dataset aims to serve as pseudo training data for supporting machine learning-based bridge component and defect detection tasks. A set of bridge component and defect-related words, 40 words, were extracted and used as queries for retrieving a large number of images from the Web. A transfer learning technique was used to transfer visual knowledge from an existing large-scale annotated dataset, ImageNet, to the domain-specific image clustering problem. A CNN model was pre-trained with the ImageNet, and the pre-trained models then extracted feature vectors for image representation. Clustering techniques were tested and evaluated for clustering images based on the extracted feature vectors. Outliers were removed after analysis. Several experiments were conducted, which indicated that the proposed image representation and clustering method is able to cluster similar domain-specific images and remove non-related images. The results show a good clustering performance, with a silhouette coefficient of 0.63.

Two main limitations of this work are acknowledged. First, the scope of the proposed image retrieval and clustering method is limited to collecting sufficient images and creating training datasets for supporting domain-specific bridge component and defect detection. A component and defect detector that would utilize such datasets could be developed in future work. Second, the evaluation effort in this paper was limited to the verification of the proposed method, by evaluating the clustering performance based on the average silhouette coefficient. Further validation of the proposed method will be conducted in future work to evaluate the performance of the created dataset in its intended application, namely component and defect detection.

Additional directions will be pursued in future work to further improve the proposed image retrieval and clustering method. First, further efforts on data cleaning and image annotation will be conducted to develop a ready-to-use training dataset for supporting semi-supervised or unsupervised machine learning-based defect detectors. Second, this work only focused on the visual features for image clustering. In future work, the authors plan to extract features from heterogeneous sources, such as text related to the images, to better support fully automatic annotation of images.

ACKNOWLEDGMENT

The authors would like to thank the National Science Foundation (NSF). This material is based on work supported by the NSF under Grant No. 1937115.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., and Ghemawat, S. (2015). "TensorFlow: Large-scale machine learning on heterogeneous systems, 2015." Software available from tensorflow.

Abdel-Qader, I., Abudayyeh, O., and Kelly, M. E. (2003). "Analysis of edge-detection techniques for crack identification in bridges." *J. Computing in Civil Engineering, 17(4), 255-263*.

ASCE. (2017). "2017 infrastructure report card." Reston, VA: ASCE 2017.

Bureau of Labor Statistics. (2019). "Census of fatal occupational injuries (CFOI) - Current and revised data." (https://www.bls.gov/iif/oshcfoi1.htm)

Chollet, F. https://github.com/fchollet/keras, (2015) "Keras."

Eisenbach, M., Stricker, R., Seichter, D., Amende, K., Debes, K., Sesselmann, M., and Gross, H. M. (2017). "How to get pavement distress detection ready for deep learning? A systematic approach." *Proc., 2017 Intl. Joint Conf. Neural Networks (IJCNN)*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation." *Proc., IEEE Conf. Computer Vision and Pattern Recognition*, 580-587.

Guérin, J., Gibaru, O., Thiery, S., and Nyiri, E. (2017). "Cnn features are also great at unsupervised classification." 1707.01700.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition." *Proc., IEEE Conf. Computer Vision and Pattern Recognition*.

Hong, Q., Wallace, R., Ahlborn, T. M., Brooks, C. N., Dennis, E. P., and Forster, M. (2012). "Economic evaluation of commercial remote sensors for bridge health monitoring." *TRB 92nd Annual Meeting Compendium of Papers.*

Hooks, J. M., and Weidner, J. (2016). "Long -Term Bridge Performance (LTBP) Program Protocols, Version 1." Turner-Fairbank Highway Research Center.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks." *Proc., Advances in Neural Information Processing Systems (NIPS) Conf*.

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., and Fieguth, P. (2015). "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure." *Advanced Engineering Informatics*, 29(2), 196-210.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., and Zitnick, C. L. (2014). "Microsoft coco: Common objects in context." *Proc., European Conf. Computer Vision*.

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., and Omata, H. (2018). "Road damage detection and classification using deep neural networks with smartphone images." *Computer-Aided Civil and Infrastructure Engineering*, 33(12), 1127-1141.

Narazaki, Y., Hoskere, V., Hoang, T. A., and Spencer, B. F. (2018). "Vision-based automated bridge component recognition integrated with high-level scene understanding." 1805.06041.

Radford, A., Metz, L., and Chintala, S. (2015). "Unsupervised representation learning with deep convolutional generative adversarial networks."

Rousseeuw, P. J. (1987). "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis." *J. Computational and Applied Mathematics, 20, 53-65*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., and Berg, A. C. (2015). "Imagenet large scale visual recognition challenge." *Intl. J. Computer Vision,* 115(3), 211-252.

Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). "CNN features off-the-shelf: an astounding baseline for recognition." *Proc., IEEE Conf. Computer Vision and Pattern Recognition Workshops*.

Simonyan, K., and Zisserman, A. (2014). "Very deep convolutional networks for large-scale

image recognition." 1409.1556.
Szegedy, C., Vanhoucke V., Ioffe, S., Shlens, J., and Wojna, Z. (2016) "Rethinking the inception architecture for computer vision." *Proc., IEEE Conf. Computer Vision and Pattern Recognition*.
Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). "How transferable are features in deep neural networks?" *Proc., Advances in Neural Information Processing Systems (NIPS) Conf*.
Zhang, L., Yang, F., Zhang, Y. D., and Zhu, Y. J. (2016). "Road crack detection using deep convolutional neural network." *Proc., 2016 IEEE Intl. Conf. Image Processing (ICIP)*.