DOI: 10.1111/mice.12425

ORIGINAL ARTICLE

WILEY COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING

Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization

Xiao Liang

Department of Civil, Structural and Environmental Engineering, University at Buffalo, The State University of New York, NY, USA

Correspondence
Xiao Liang, Department of Civil, Structural and Environmental Engineering, University at Buffalo, The State University of New York, Buffalo, NY 14260, USA.
Email: liangx@buffalo.edu

Abstract

Many bridge structures, one of the most critical components in transportation infrastructure systems, exhibit signs of deteriorations and are approaching or beyond the initial design service life. Therefore, structural health inspections of these bridges are becoming critically important, especially after extreme events. To enhance the efficiency of such an inspection, in recent years, autonomous damage detection based on computer vision has become a research hotspot. This article proposes a three-level image-based approach for post-disaster inspection of the reinforced concrete bridge using deep learning with novel training strategies. The convolutional neural network for image classification, object detection, and semantic segmentation are, respectively, proposed to conduct system-level failure classification, component-level bridge column detection, and local damage-level damage localization. To enable efficient training and prediction using a small data set, the model robustness is a crucial aspect to be taken into account, generally through its hyperparameters' selection. This article, based on Bayesian optimization, proposes a principled manner of such selection, with which very promising results (well over 90% accuracies) and robustness are observed on all three-level deep learning models.

1 | INTRODUCTION

Many bridge structures, according to ASCE Infrastructure Report Card (2017), are rated between mediocre and poor and are approaching or beyond the initial design service life with exhibited signs of deteriorations. Therefore, structural health condition inspections and rapid damage assessment (Farrar & Worden, 2012) of these bridges, as one of the most critical components in transportation infrastructure systems, are becoming critically important, especially after extreme events (e.g., earthquake). At present, post-disaster condition screening is done manually by sending a dedicated team. Such practice is time consuming and biased in nature, highly relying on the qualitative judgment of an inspector. These inefficiencies can be overcome if the current manual evaluation practices are fully automated based on the visible load-bearing structural components (Zhu & Brilakis, 2010) and the visible surface damage, which lays on them. Remote sensing using light detection and ranging (LiDAR) has been found to be very useful in rapid damage assessment (e.g., Fernandez Galarreta, Kerle, & Gerke, 2015; Khoshelham, Elberink, & Xu, 2013) but largely limits the structural information to the vertical data. In recent years, considerable efforts have been dedicated to automating the visual inspection (Jahanshahi, Kelly, Masri, & Sukhatme, 2009; Koch, Georgieva, Kasireddy, Akinci, & Fieguth, 2015; Rose et al., 2014) using computer vision-based approaches. Basically, these methods are built on conventional image processing techniques, such as histogram transforms, filtering, and texture recognition (Koch et al., 2015). Much of the research in defect detection and assessment largely focused on cracks (e.g., Abdel-Qader, Abudayyeh, & Kelly, 2003; Feng & Feng, 2017; Torok, Golparvar-Fard, & Kochersberger, 2013; Yamaguchi & Hashimoto 2010; Yeum & Dyke, 2015; Yoon,

Elanwar, Choi, Golparvar-Fard, & Spencer, 2016; Zhu, German, & Brilakis, 2011), and to some extent on delamination/spalling (e.g., German, Brilakis, & DesRoches, 2012) and rusting (e.g., Koch et al., 2014). Most of these works generally rely on handcrafted extracting low-dimensional features and classical machine learning techniques. Therefore, these methods are generally still time consuming and may not be effective considering that defects in real-world conditions are complex with inevitable background noise.

Great progress has been made in recent years on computer vision through deep learning in which convolutional neural network (CNN) has been the primary momentum. CNNs saw heavy use in the 1990s (e.g., LeCun, Bottou, Bengio, & Haffner, 1998) but then fell out of fashion with the rise of support vector machines (SVMs). Krizhevsky, Sutskever, and Hinton (2012) rekindled interest in CNNs by training a large CNN on the ImageNet (Deng et al., 2009), showing substantially higher accuracy on image classification. Thereafter, many other CNN networks with deeper architectures, such as VGG Net (Simonyan & Zisserman, 2014), Google Net (Szegedy et al., 2015), Inception Net (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016), and Deep Residual Net (He, Zhang, Ren, & Sun, 2016), have been proposed. Several CNN applications for classification in civil engineering include railway defects detection (Soukup & Huber-Mörk, 2014), collapse classification (Yeum, Dyke, Ramirez, & Benes, 2016), concrete crack detection (Cha, Choi, & Büyüköztürk, 2017), structural damage detection through a one-dimensional CNN (Abdeljaber, Avci, Kiranyaz, Gabbouj, & Inman, 2017), pavement crack detection (Vetrivel, Gerke, Kerle, Nex, & Vosselman, 2018; Zhang et al., 2017), structural damage recognition through transfer learning (Gao & Mosalam, 2018), and structural damage detection with feature extracted from deep learning (Lin, Nie, & Ma, 2017). Other notable civil engineering applications of deep learning involve the estimation of concrete strength (Rafiei, Khushefati, Demirboga, & Adeli, 2017), the damage detection in high-rise buildings (Rafiei & Adeli, 2017), the global and local assessment of structural health condition (Rafiei & Adeli, 2018), the accelerated reliability analysis of transportation networks (Nabian & Meidani, 2018), and structural reliability assessment (Dai, 2017).

Object detection is another domain in computer vision that draws a lot of research interest as ones are not only interested in object classification, but detection and localization as well. Its success is driven by the advances in regional proposal methods (Uijlings, Van de Sande, Gevers, & Smeulders, 2013) and region-based CNNs (R-CNNs) (Girshick, Donahue, Darrell, & Malik, 2014). Comprehensive surveys and comparisons of object proposal methods can be found in Hosang, Benenson, and Schiele (2014), Hosang, Benenson, Dollár, and Schiele (2015), and Chavali, Agrawal, Mahendru, and Batra (2015). Compared to computationally expensive R-CNN as originally developed in Girshick et al. (2014), its cost has been reduced dramatically as in Fast R-CNN (Girshick, 2015) through sharing convolutions across proposals. The state-of-the-art incarnation as in Faster R-CNN (Ren, He, Girshick, & Sun, 2017) can achieve near-real-time detection by sharing computation between a region proposal network (RPN) and the Fast R-CNN. Limited application in structural engineering includes the detection and localization of multiple types of damage (Cha, Choi, Suh, Mahmoudkhani, & Büyüköztürk, 2018; Suh & Cha, 2018), shield tunnel lining defects detection (Xue & Li, 2018), and concrete defect detection and geolocalization with a unified methodology (Li, Yuan, Zhang, & Yuan, 2018).

When it comes to the scene understanding and inferring support-relationship among objects (e.g., as is the common case in autonomous driving), the object classification and detection are not sufficient where the semantic segmentation steps in. Recent architectures essentially conduct the pixel-wise classification generally through an encoder–decoder architecture (e.g., Badrinarayanan, Kendall, & Cipolla, 2017; Eigen & Fergus, 2015; Hong, Noh, & Han, 2015; Long, Shelhamer, & Darrell, 2015; Noh, Hong, & Han, 2015; Zheng et al., 2015). Very limited structural engineering application includes defect detection and classification on civil infrastructure (Feng, Liu, Kao, & Lee, 2017), structural inspection of different types of damage (Hoskere, Narazaki, Hoang, & Spencer, 2018), and bridge component extraction (Narazaki, Hoskere, Hoang, & Spencer, 2018).

In this article, an image-based approach for post-disaster inspection of the reinforced concrete (RC) bridge using deep learning is proposed. The proposed approach consists of three sequential levels: system-level failure analysis, component-level detection, and local-level damage localization. These three levels are, respectively, realized by the three developed deep learning models that are corresponding to classification, detection, and semantic segmentation. Many existing approaches utilize images with relatively large defects and nearly uniform background for model training and testing. Instead, in this study, real-world images (Koziarski & Cyganek, 2017) with the cluttered background (e.g., shadow, reflection) are collected. It is noted that the training of the three models is sequential with the strategy that the trained parameter values of the previous model are used to initialize the following one. For example, the deep learning model for bridge column detection is initialized using the trained values of parameters of the model for system-level failure analysis. Another important aspect of the model development is the model robustness, in which the selection of hyperparameters is critical, especially for dealing with the small data set as in this study. A good robustness herein refers to whether the training and testing accuracies are high and comparable. In other words, neither overfitting (high training accuracy while testing one is low) nor underfitting (both accuracies are low) is acceptable. In this study, to enhance model robustness, a

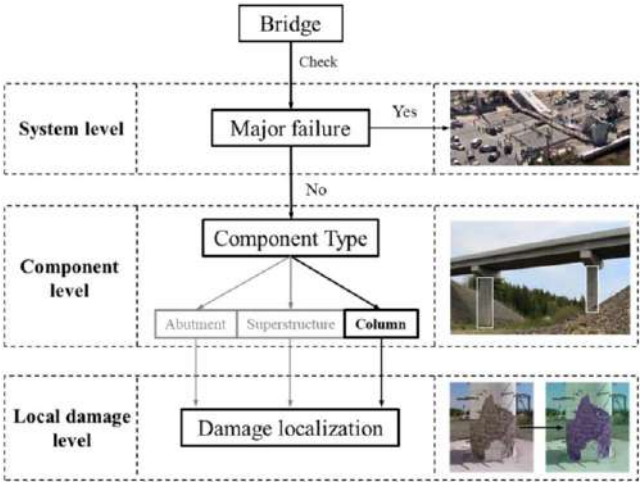


FIGURE 1 The depiction of the proposed image-based inspection approach

FIGURE 3 Sample images used in classification for the major failure of the bridge system

principled manner for hyperparameters' selection utilizing Bayesian optimization (the generalization performance of the deep learning model is treated as a sample from a Gaussian process) is proposed to enhance model robustness with relatively few evaluations.

2 | IMAGE-BASED INSPECTION APPROACH

In this section, an image-based approach for post-disaster bridge inspection using deep learning is introduced. It is constituted by three sequential components: the system-level failure analysis, the structural component-level detection, and the local-level damage localization. The first one determines the existence of a system-level major failure, which refers to a local failure or collapse with visible permanent deformation (Veletzos, Panagiutou, Restrepo, & Sahs, 2008). If no such failure is identified, the structural components of the bridge are detected and localized. In this study, the RC bridge column, as the critical supporting structure, is selected to demonstrate for the component-level detection. Subsequently, the local-level damage of the RC bridge column can be detected and localized. The local damage herein refers to visual changes in the bridge column, such as delamination/spall area, exposed rebar, and cracking (Koch et al., 2015). The entire approach is depicted in Figure 1.

Therefore, in this approach, three models, i.e., the existence of major failure, the component detection, and the damage localization, are developed and correspondingly three sets of real-world images are collected. The main two sources of the collected images are related research reports on RC bridges (e.g., Veletzos et al., 2008; Sideris, 2012) and search engines (e.g., Google Image). The bridges in the collected images are mainly the highway overpasses over the freeways, railways, and rivers (e.g., the recently collapsed bridges in Florida and Italy). The data set also contains the post-earthquake images of damaged RC bridges around the world (e.g., Northridge earthquake in United States, Kobe earthquake in Japan, Izmit earthquake in Turkey, Chi-Chi earthquake in Taiwan) as well as the images of RC columns in different experimental studies (e.g., Calderone, Lehman, & Moehle, 2001; Esmaeily & Xiao, 2005). Figure 2 gives the resolution of the 1,154 images for all the three models. All images are manually labeled by the author. The first set of totally 492 images (about 80% of the images in this set contain the whole bridge) contains two scenarios, of which 291 (Figure 3a) are for bridge systems

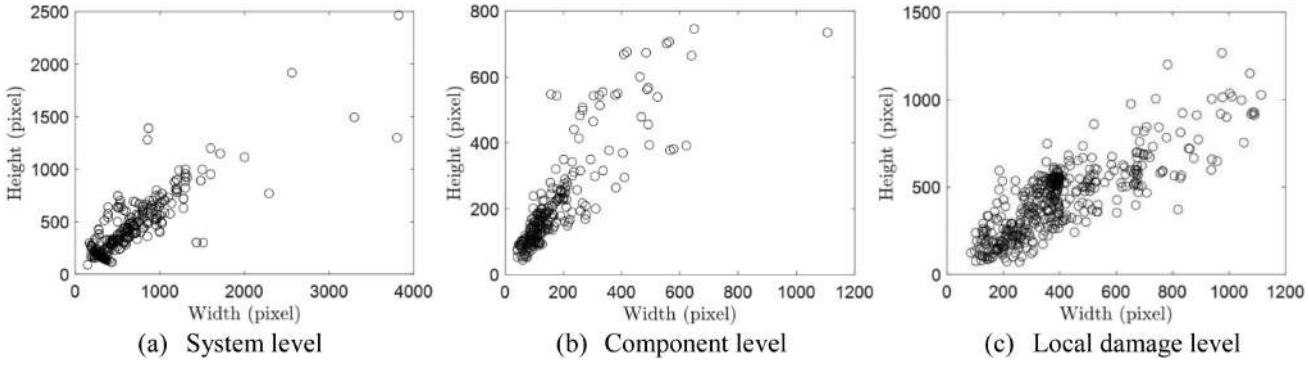


FIGURE 2 Resolution of the collected images for the three deep learning models

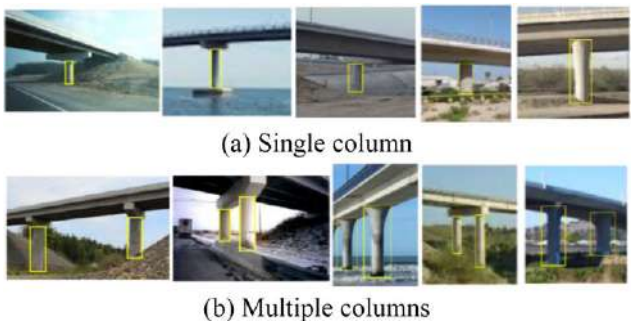


FIGURE 4 Sample images and their ground-truth labels used in bridge column detection

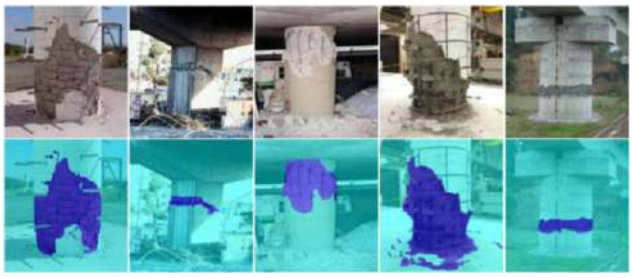


FIGURE 5 Sample images and their ground-truth labels used in damage localization

with a major failure, and the other 201 (Figure 3b) images represent no major failure detected. The second set, which is corresponding to RC bridge column detection, has 236 images with 344 bridge columns labeled with rectangular bounding boxes, indicating that one (Figure 4a) or more columns (Figure 4b) are labeled in each image. A rectangular bounding box can be determined by four numbers: the x-y coordinate corresponding to one of the four angle points, the width and the height of the rectangle (the x-y coordinates of the other three angle points can be determined as well). The last set contains 436 images and the local damages are pixel-wise labeled (Figure 5). It is worth noting that such pixel-wise labeling is much more expensive (Badrinarayanan et al., 2017) than that for the previous two sets of images, i.e., two scenarios for the system failure classification and the rectangular bounding boxes for the bridge column detection. It is noted that the rectangular bounding boxes are not used to label the damage. This is because the damage, as shown in Figure 4, generally does not present as a regular shape, and thus many feasible but likely conservative combinations of rectangular bounding boxes exist that can cover all the damaged areas. For classification problems, the labeled data are usually split into two sets, one for model training and the other for generalization testing. For all three tasks, following the general practice, 80% of the images are used for training and the remaining 20% serve for testing purposes.

3 | MODEL DEVELOPMENT AND TRAINING

The developments and training of the previously mentioned three models using deep learning are described in this section.

3.1 | System-level failure classification

The system-level failure identification is a binary classification task to determine if the bridge has any system-level major failure. This task can be tackled utilizing CNN, which is usually trained using millions of images (e.g., ImageNet). However, this is not the case for the civil engineering application because, as mentioned previously, only 492 images are collected for the task of this level. Even considering that the problem at hand is a binary classification task, the number of images is still very limited to perform an end-to-end training using a deep CNN architecture. This is where the transfer learning can step in and help with the situation of data scarcity, taking advantage of certain pretrained models. In this study, the Simonyan and Zisserman (2014) model (VGG-16) is applied, whose performance and generalization have been verified, as the pretrained model for transfer learning to conduct the system-level failure analysis. Such selection is reached through the comparisons with two other pretrained models: AlexNet (Krizhevsky et al., 2012) and Google Net (Szegedy et al., 2015; the comparisons are reported in Table 4 and will be discussed in the next section). The schematic architecture of the CNN used for this task, which contains different layers, is given in Figure 6. The convolutional (Conv) block, a Conv layer followed by a layer of rectified-linear nonlinearity (ReLU) $\max(0, x)$ (identified as "Conv + ReLU" in Figure 6; Nair & Hinton, 2010), plays the core computational role in CNNs. The key idea is to adopt a set of trainable filters (e.g., a 3×3 filter is used in this study) to slide over the image and calculate their corresponding dot product (convolution) such that its spatial information is utilized, i.e., a pixel is generally not correlated with others that are far from it. This leads to dramatic reduction of the network trainable parameters (in terms of several magnitudes) compared to that of the traditional neural network, which usually consists of only fully connected (FC) layers. Following several rounds of Conv blocks (two or three as in Figure 6), a pooling layer is performed to reduce the dimensionality of the image and maintain invariances to small shifts and distortions. For example, a 2×2 max-pooling (Nagi et al., 2011) with stride 2 (nonoverlapping window) is used in this study. After a series of Conv blocks and pooling layers, the CNNs usually end up with several FC layers (e.g., 3 FC layers for VGG-16 with the sizes of 4,096, 4,096, and 1,000, respectively). FC layers are prone to overfitting because they occupy most of the parameters in the network. To reduce overfitting, 50% dropout is applied to the FC layers (Srivastava, Hinton, Krizhevsky,

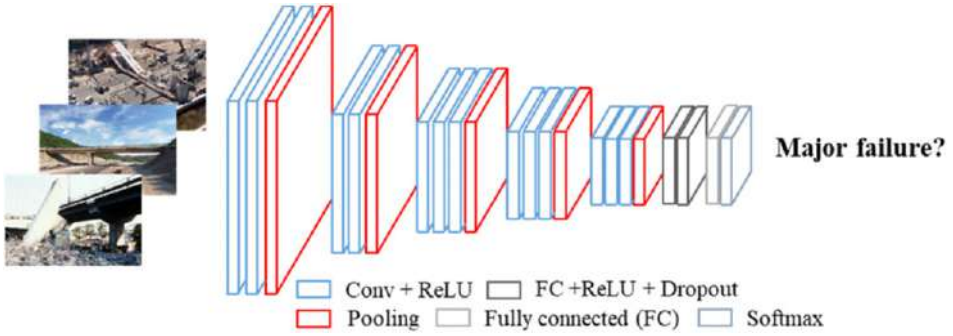


FIGURE 6 The schematic architecture of the CNN for the system-level failure analysis

Sutskever, & Salakhutdinov, 2014). The main idea of dropout is to, during training, randomly drop units along with connections from the neural network, such that only the reduced network is trained on the data, i.e., the dropped ones are not activated in both forward- and backpropagations. Then, a Softmax layer takes the input from the last FC layer and serves as a multiclass classifier to calculate the probabilities of each individual class and output the class with the highest probability as the classification result. It is noted that the VGG-16 (with 16 trainable layers, i.e., 13 Conv and 3 FC ones) was trained on ImageNet for classification of 1,000 classes. However, it cannot be directly applied to this task as the two classes (the major failure and no major failure) are not among the 1,000 classes.

The idea of transfer learning in CNN is to help improve the current classification task using the acquired knowledge (e.g., architectures and weights) from previous ones with training on different data sources. Two general strategies of transfer learning are available in CNNs. The first one is to utilize the pretrained model as a feature extractor and then apply classical machine learning algorithms to perform the classification. For example, the features can be extracted from the last max-pooling layer of VGG-16 and SVM can be subsequently applied to these extracted features for classification. The second strategy, which is adopted in this study, is to reset the last FC layer of the VGG-16 model to have the same size as the number of classes in this task (i.e., two classes) and make the new layer learn much faster than the transferred ones (Goodfellow, Bengio, & Courville, 2016). In this case, data augmentation can be applied to help prevent the network from overfitting and memorizing the exact details of the training images. To retain the architecture of the VGG-16, all images for this task (e.g., those in Figure 3) are resized into $224 \times 224 \times 3$, where 3 denotes the three RGB channels of the image. The numbers of trainable filters for the 13 Conv layers are reported in Table 1.

TABLE 1 Numbers of trainable filters for the 13 Conv layers

Conv #	# of filters	Conv #	# of filters
1–2	64	3—4	128
5–7	256	8—10	512
11–13	512		

3.2 | Bayesian optimization

The training is usually carried out by optimizing the loss function based on backpropagation (LeCun et al., 1989; Ortega-Zamorano, Jerez, Gómez, & Franco, 2017) using stochastic gradient descent (SGD)

$$\theta_{i+1} = \theta_i - \alpha \nabla L\left(\theta_i\right) \tag{1}$$

where i stands for the iteration number, $\alpha > 0$ is the learning rate, θ is the trainable parameter vector, and $L(\theta)$ is the cross-entropy loss function. The SGD algorithm might oscillate along the path of steepest descent toward the optimum. Adding a momentum term to the parameter update is one way to reduce such oscillation (Murphy, 2012).

$$\theta_{i+1} = \theta_i - \alpha \nabla L\left(\theta_i\right) + \gamma\left(\theta_i - \theta_{i-1}\right) \tag{2}$$

where the momentum factor γ determines the contribution of the previous gradient step to the current iteration. Besides previously mentioned dropout technique and data augmentation tricks, adding a regularization term for the weights (weight decay) to the loss function $L(\theta)$ is another way to reduce overfitting (Bishop, 2006; Murphy, 2012). Thus, the loss function with the regularization term takes the following form:

$$\begin{aligned} L_R(\theta) &= L(\theta) + \lambda \Omega(\mathbf{w}) \\ \Omega(\mathbf{w}) &= \tfrac{1}{2}\mathbf{w}^T\mathbf{w} \end{aligned} \tag{3}$$

where $\mathbf{w}$ and λ are, respectively, the weight vector and the regularization coefficient. Since most architectures of the VGG-16 are inherited in the current CNN model, it has many less number of hyperparameters than one that is built from scratch. The hyperparameters herein are the parameters that are not trainable, i.e., the ones whose values are set before the training process begins (e.g., filter sizes and numbers of Conv layers). Still, a few hyperparameters (e.g., the learning rate α, the momentum factor γ, and the regularization coefficient λ) should be considered in the developments of the three

models to achieve a greater robustness for the target small data set. The general practice is to set 20% of the training data as the validation set and to train the model using the remaining images in the training set. In this task, $492 \times 0.8 \times 0.8 \approx 315$, $492 \times 0.8 \times 0.2 \approx 79$, and $492 \times 0.2 \approx 98$ images are, respectively, used for training, validation, and testing purposes.

The objective function E_V, the validation error that is to be minimized, is clearly nonconvex and has no closed-form expression. Therefore, gradient-based optimization approaches are not applicable. In addition, the evaluation of this function is very expensive that prohibits the use of certain brute force methods (e.g., grid search and genetic algorithm).

Bayesian optimization is a powerful strategy for finding the extrema of such objective function and it is particularly useful when the objective function is expensive to evaluate. The idea is to construct a probabilistic model for the objective function and, instead of solely relying on the local gradient and Hessian approximations, to take advantage of all the available information from the previous evaluations. As such, at the expense of performing more computations to determine the next point to evaluate, Bayesian optimization generally can find the extrema of difficult nonconvex functions with relatively few evaluations (Brochu, Cora, & De Freitas, 2010).

The fundamental assumption adopted in Bayesian optimization is that the objective function $E_V(\bar{\theta})$ is drawn from a Gaussian process (GP) prior, i.e., $E_V(\bar{\theta}) \sim N(\mathbf{0}, \mathbf{K})$, where $\bar{\theta}$, in this study, is a vector of the considered hyperparameters. Considering that $E_V(\bar{\theta})$ is corrupted with Gaussian noise with zero mean and standard deviation of σ_{noise}, the kernel matrix is given by

$$\mathbf{K} = \begin{bmatrix} k\left(\bar{\theta}_1, \bar{\theta}_1\right) & \dots & k\left(\bar{\theta}_1, \bar{\theta}_t\right) \\ \vdots & \ddots & \vdots \\ k\left(\bar{\theta}_t, \bar{\theta}_1\right) & \dots & k\left(\bar{\theta}_t, \bar{\theta}_t\right) \end{bmatrix} + \sigma_{\text{noise}}^2 \mathbf{I} \tag{4}$$

where $k(\bar{\theta}, \bar{\theta}')$ is the covariance function. Denote the observations from the previous iterations as $\mathbf{D}_{1:t} = \{\bar{\theta}_{1:t}, \mathbf{E}_{1:t}^V\}$, where $\mathbf{E}_{1:t}^V = E_V(\bar{\theta}_{1:t})$. Denote $\bar{\theta}_{t+1}$ as the next point to evaluate and the value of the function at $\bar{\theta}_{t+1}$ as $E_{t+1}^V = E_V(\bar{\theta}_{t+1})$. Under the GP prior, $\mathbf{E}_{1:t}^V$ and E_{t+1}^V are jointly Gaussian and the following expression for the predictive distribution can be obtained (Rasmussen & Williams 2006):

$$E_{t+1}^V | \mathbf{D}_{1:t} \sim N\left(\mu\left(\bar{\theta}_{t+1}\right), \sigma^2\left(\bar{\theta}_{t+1}\right) + \sigma_{\text{noise}}^2\right) \tag{5}$$

where

$$\mu\left(\bar{\theta}_{t+1}\right) = \mathbf{k}^T \left(\mathbf{K} + \sigma_{\text{noise}}^2 \mathbf{I}\right)^{-1} \mathbf{E}_{1:t}^V \tag{6}$$

$$\sigma^2\left(\bar{\theta}_{t+1}\right) = k\left(\bar{\theta}_{t+1}, \bar{\theta}_{t+1}\right) - \mathbf{k}^T \left(\mathbf{K} + \sigma_{\text{noise}}^2 \mathbf{I}\right)^{-1} \mathbf{k} \tag{7}$$

$$\mathbf{k} = \left[k\left(\bar{\theta}_{t+1}, \bar{\theta}_1\right)\ k\left(\bar{\theta}_{t+1}, \bar{\theta}_2\right) \cdots k\left(\bar{\theta}_{t+1}, \bar{\theta}_t\right)\right]^T \tag{8}$$

Therefore, the predictive posterior distribution $E_{t+1}^V | \mathbf{D}_{1:t}$ is sufficiently characterized by its predictive mean function $\mu(\bar{\theta}_{t+1})$ and predictive variance function $\sigma^2(\bar{\theta}_{t+1})$, which solely depend on the selection of the covariance function $k(\bar{\theta}, \bar{\theta}')$. In this study, the automatic relevance determination (ARD) Matérn 5/2 kernel (Snoek, Larochelle, & Adams, 2012) is used. Such selection accommodates the problem of the ARD squared exponential kernel (usually the default choice), in that, the sample functions with this covariance function are unrealistically smooth for practical optimization problems (Snoek et al., 2012).

As mentioned previously, Bayesian optimization tends to distribute more computations on determining the next point $\bar{\theta}_{t+1}$ to evaluate, and it does so by using an acquisition or utility function constructed from the above-discussed predictive posterior distribution. The acquisition function used in this study is the expected improvement (EI) over the best expected value $\mu_{\text{best}} = \arg\min_{\bar{\theta}_j \in \bar{\theta}_{1:t}} \mu(\bar{\theta}_j)$, which has a closed-form solution under the GP (Snoek et al., 2012) assumption as follows:

$$a_{\text{EI}}\left(\bar{\theta}_{t+1}\right) = \sigma\left(\bar{\theta}_{t+1}\right)\left[Z\,\Phi(Z) + \phi(Z)\right] \tag{9}$$

where $\Phi(\cdot)$, $\phi(\cdot)$, and Z are respectively cumulative distribution function, PDF of the standard normal, and

$$Z = \frac{\mu_{\text{best}} - \mu\left(\bar{\theta}_{t+1}\right)}{\sigma\left(\bar{\theta}_{t+1}\right)} \tag{10}$$

Therefore, what point should be evaluated next can be determined via a proxy optimization: the maximization of the acquisition function, i.e., to maximize the EI over the current best expected value. It is noted that, unlike the original unknown objective function, $a_{\text{EI}}(\cdot)$ in Equation (9) can be cheaply sampled to be maximized. The algorithm of deep learning with Bayesian optimization is summarized in Table 2.

Bayesian optimization is applied to determine the three hyperparameters in developing the three deep learning models each with 60 (a minimum of 30 is recommended in Brochu et al., 2010) iterations or evaluations on the validation error, i.e., 60 deep learning models (with different values of the hyperparameters) are trained for each of the three models. Then the deep learning models can be obtained by training all images in the entire training set (including those used in validation) using the values of the hyperparameters that correspond to the smallest validation error.

3.3 | Component-level bridge column detection

Per the proposed inspection approach illustrated in Figure 1, if the bridge is identified as free of any system failure through

TABLE 2 The algorithm of deep learning with Bayesian optimization

for $t = 1, 2, \ldots$

1. Calculate the predictive mean function $\mu(\bar{\theta}_{t+1})$ in Eq. (6) and predictive variance function $\sigma^2(\bar{\theta}_{t+1})$ in Eq. (7) using the chosen ARD Matérn 5/2 kernel function
2. Find $\bar{\theta}_{t+1} = (\alpha_{t+1},\ \gamma_{t+1},\ \lambda_{t+1})$ by optimizing the acquisition function over the GP: $\bar{\theta}_{t+1} = \arg\max_{\bar{\theta}} a_{\mathrm{EI}}(\bar{\theta}|\mathbf{D}_{1:t})$
3. Evaluate the validation error $E_V(\bar{\theta}_{t+1})$ through the deep learning model with $\bar{\theta}_{t+1} = (\alpha_{t+1},\ \gamma_{t+1},\ \lambda_{t+1})$ determined in step 2
4. Augment the data $\mathbf{D}_{1:t+1} = \{\mathbf{D}_{1:t}, (\bar{\theta}_{t+1}, \mathbf{E}^V_{t+1})\}$ and update the GP

end

the first model, the next step is to detect and localize different structural components. In this study, the RC bridge column, the critical supporting components of bridge systems, is selected to demonstrate for the component-level detection.

Different from image classification, detection requires localizing (likely multiple) objects within an image, and thus contains two stages: a method for region proposal generation in the image and a detector using the proposals. These two stages were generally separated, such as in R-CNN (Girshick et al., 2014) and Fast R-CNN (Girshick, 2015) methods. In this study, a method named Faster R-CNN is used that unifies the two stages into a single network with shareable Conv layers between two tasks. It is composed of two computational sharing modules: an RPN that proposes a set of rectangular regions and the Fast R-CNN detector that uses the proposed regions. Its architecture for this task is illustrated in Figure 7. The single unified network starts with 13 shareable Conv layers as in Figure 6. To generate region proposals, a small network is slid over the feature map output by the last shared Conv layer. This small network takes as input a 3×3 spatial window of the feature map. An anchor, a rectangular box associated with a specified scale and aspect ratio, is centered at the sliding window. Nine anchors are yielded by considering three scales and three aspect ratios for each sliding position. A positive anchor is identified if either its Intersection-over-Union (IoU) overlap with one ground-truth box is highest or it has an IoU overlap greater than 0.6 (0.7 is used in Ren et al., 2017) with any ground-truth box. An anchor is labeled negative if its IoU overlap is less than 0.3 for any ground-truth box. The IoU is defined as follows:

$$\mathrm{IoU} = \frac{\mathrm{Area\,(Predition)} \cap \mathrm{Area\,(Groundtruth)}}{\mathrm{Area\,(Predition)} \cup \mathrm{Area\,(Groundtruth)}} \tag{11}$$

Therefore, the architecture is implemented with a 3×3 Conv layer followed by two sibling 1×1 Conv layers, respectively, for the classification (i.e., being a bridge column or not) and the regression (i.e., the rectangular bounding box). The numbers of trainable filters for these three layers are, respectively, set to be 512, 48, and 96. With all these, the training of RPN is carried out to minimize the multitask loss function as follows:

$$L\left(\{p_l\}, \{t_l\}\right) = \frac{1}{N_{cls}} \sum_l L_{cls}\left(p_l, p_l^*\right) + \eta \frac{1}{N_{reg}} \sum_l p_l^* L_{reg}\left(t_l, t_l^*\right) \tag{12}$$

with

$$L_{reg}\left(t_l, t_l^*\right) = \begin{cases} 0.5\left(t_l - t_l^*\right)^2 & \text{if } \left|t_l - t_l^*\right| < 1 \\ \left|t_l - t_l^*\right| - 0.5 & \text{otherwise} \end{cases} \tag{13}$$

where l denotes the index of the anchor, p_l is the predicted probability of the anchor k being a bridge column, L_{cls} is the log loss over two classes (being a bridge column or not),

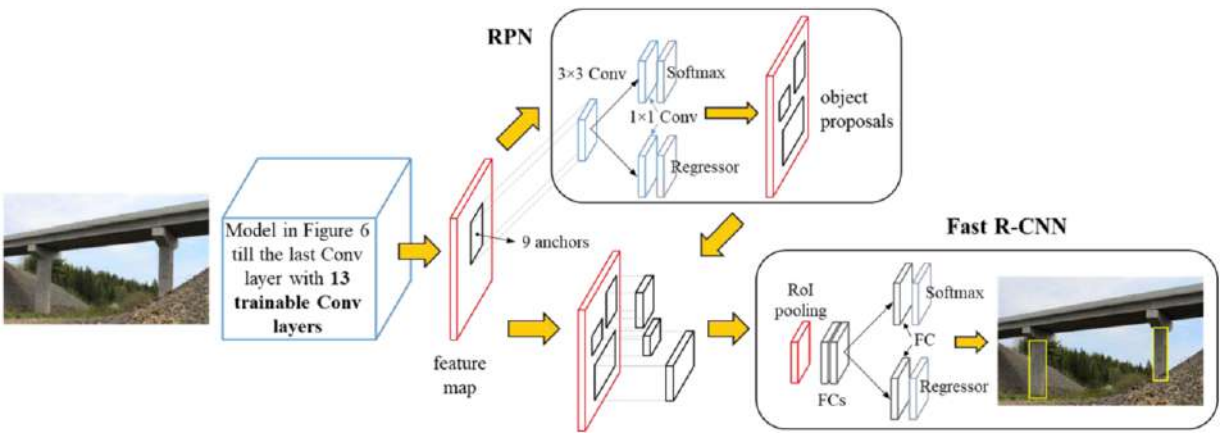


FIGURE 7 The schematic architecture of the Faster R-CNN for bridge column detection

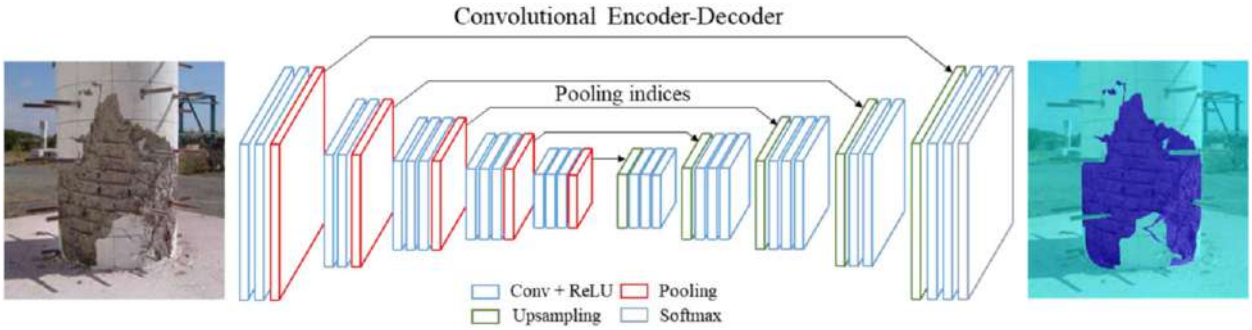


FIGURE 8 The schematic architecture of the semantic segmentation for local damage

TABLE 3 Numbers of trainable filters for the 13 Conv layers in the decoder network

Conv #	# of filters	Conv #	# of filters
1–3	512	4–6	512
7–9	256	10–11	128
12–13	64		

and p_l^* is the label of ground-truth that $p_l^* = 1$ for positive anchors and $p_l^* = 0$ for negative ones. t_l and t_l^* are, for a positive anchor, respectively, vectors of four parameterized coordinates of the predicted and ground-truth bounding boxes. The values of N_{cls} and N_{reg} are, respectively, selected as the mini-batch size and the number of anchors such that the two terms in the loss function are roughly balanced with $\eta = 10$.

After the training of RPN, Fast R-CNN detector is adopted utilizing these proposals. For each objective proposal, a region of interest (RoI) pooling layer (with a grid size of 7×7) takes as input a fixed-size feature map and then maps it to a feature vector by FC layers. Similar to the RPN, following the same multitask loss function in Equation (12), the Fast R-CNN, for each proposal, also has two outputs: softmax probability estimates of being a bridge column and a rectangular bounding box determined by four real-valued numbers.

In this study, a four-step alternating training algorithm recommended in Ren et al. (2017) is adopted with all images for this task (e.g., those in Figure 4) resized such that their width pixels are 400. It starts with training the RPN that is initialized with the model in Section 3.1 (instead of the pretrained VGG-16 model used in Ren et al., 2017) and an end-to-end fine-tuned for the region proposal task. In the second step, using the proposals generated by the first step, Fast R-CNN, initialized also using the model in Section 3.1, is trained. It is noted that, at this point, Conv layers have not been shared by the two networks. Next, the RPN is trained again but it is initialized with the detection network in which 13 shared Conv layers (blue block in Figure 7) are fixed and fine-tunes only the remaining layers (RPN in Figure 7) that are unique to RPN. The final step also only fine-tunes the unique layers of the detection network (Fast R-CNN in Figure 7) while fixing the 13 shared Conv layers. As such, in the prediction stage, the Conv computations are shared between the two networks, leading to an efficient near-real-time object detection algorithm. The last two steps fine-tune the network, so their learning rates are set to be one-tenth of those for the first two steps. In this study, the training epochs of RPN (the first and the third steps) and Fast R-CNN (the second and the final steps) are set to be 50 and 100, respectively. The entire network is still trained under the framework of Bayesian optimization considering the three previously mentioned hyperparameters.

3.4 | Local-level damage localization

The natural next step is to locate all damages, and thus the model must have the ability to delineate damage, irrespective of its shape. For this purpose, a fully deep CNN architecture (Badrinarayanan et al., 2017) is proposed to use for damage semantic pixel-wise segmentation. It contains an encoder network and a corresponding decoder network followed by a final pixel-wise classification layer. The architecture is illustrated in Figure 8. The encoder network consists of 13 Conv layers, which are topologically identical to the convolutional layers in Figure 6 (and also the blue block in Figure 7). At the expense of certain memory, the network also involves storing, for all max-pooling windows of all max-pooling layers in the encoder network, max-pooling indices that record the locations of the maximum feature value. However, it is still much easier to train the network as the FC layers are discarded and thus the number of parameters reduce significantly. The decoder network, the key component of the architecture, is to map the low-dimensional encoder feature maps back to full input resolution ones for pixel-wise classification. It contains a hierarchy of decoders—one corresponding to each encoder. The upsampling layer decodes its input feature map using the previously mentioned memorized max-pooling indices and outputs sparse feature maps. These feature maps are then convolved with a set of trainable decoder filters (the numbers of filters in the decoder network are equal to those in the corresponding encoder network, documented in Table 3)

TABLE 4 Comparisons of the three different pretrained models

Pretrained model	AlexNet	Google Net	VGG-16
Input size	$227 \times 227 \times 3$	$224 \times 224 \times 3$	$224 \times 224 \times 3$
Initial learning rate	9.8572×10^{-4}	7.1446×10^{-4}	5.6645×10^{-4}
Momentum factor	0.8006	0.8594	0.8223
Regularization coefficient	8.4000×10^{-3}	2.8532×10^{-7}	3.2644×10^{-4}
Testing accuracy	93.88%	95.92%	98.98%

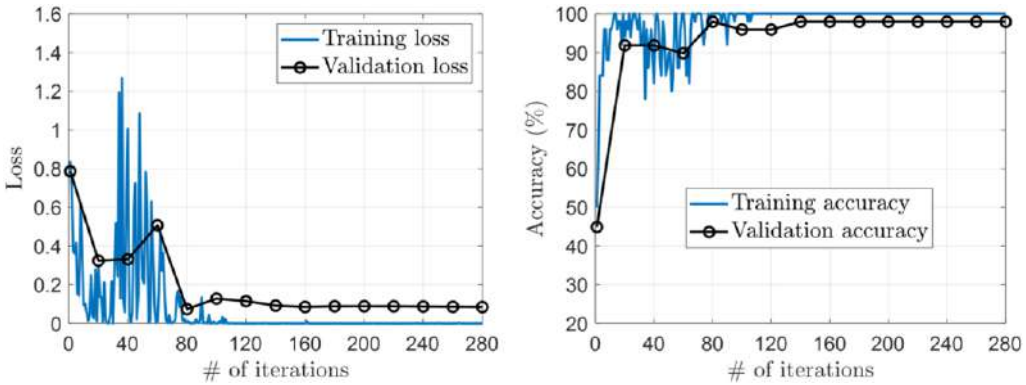


FIGURE 9 System-level failure analysis: loss (left) and accuracy (right) curves for training and validation sets of images using the values of hyperparameters determined by Bayesian optimization

TABLE 5 Comparisons of testing accuracy between with and without Bayesian optimization for the three models

Model	System level		Component level		Local damage level	
Bayesian optimization	Yes	No	Yes	No	Yes	No
Initial learning rate	5.6645×10^{-4}	1.0000×10^{-2}	9.2993×10^{-5}	1.0000×10^{-3}	9.9994×10^{-4}	1.0000×10^{-5}
Momentum factor	0.8223	0.9000	0.9107	0.9000	0.9356	0.8000
Regularization coefficient	3.2644×10^{-4}	1.0000×10^{-2}	1.0000×10^{-4}	1.0000×10^{-2}	6.5009×10^{-6}	1.0000×10^{-2}
Testing accuracy	98.98%	59.49%	96.49%	85.50%	93.14%	80.21%

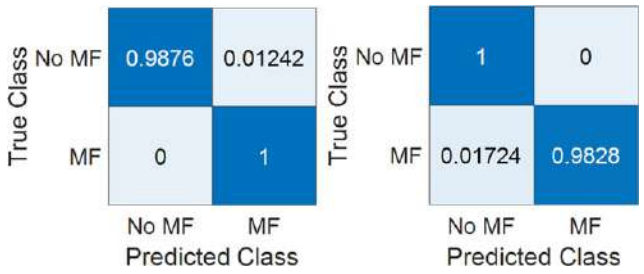


FIGURE 10 System-level major failure (MF) versus no MF: confusion matrices of training (left) and testing (right) sets

to produce dense feature maps. The high-resolution feature representation at the output of the final decoder is then fed to a Softmax layer that classifies each pixel independently.

The cross-entropy loss, in which loss is summed up over all the pixels in a mini-batch, is used as the objective function for training. Ideally, all classes would have an equal number of observations. However, the two classes (i.e., damage and background) in the training images of this task are imbalanced because damage portion generally covers much less area in the image. If not handled correctly, such considerable variation in the number of pixels in each class can be detrimental to the learning process because the learning is biased in favor of the dominant classes (i.e., background in this case). Therefore, a median frequency balancing technique (Eigen & Fergus, 2015) is applied to assign different weights to each class in the loss function. The weight is equal to the median frequency (of all classes) divided by the class frequency. This implies that the larger class (i.e., background) would have a smaller weight. Using the pretrained weights of the model in Section 3.3, the training is then initialized using images resized at 430×400 resolution by considering the three previously mentioned hyperparameters inside the framework of Bayesian optimization.

4 | EXPERIMENTS AND RESULTS

In this section, the results of the three models are presented and discussed. All experiments are conducted using MATLAB 2018a on three computers: an XPS 9550 (a Core

FIGURE 11 Sample testing images together with their predicted classes and the probabilities of those classes

TABLE 6 Computational time (s) for the three models

Model	System level	Component level	Local damage level
Training time (per epoch)	8.6000	26.7567	87.9706
Execution time (per image)	0.0117	0.1687	0.1493

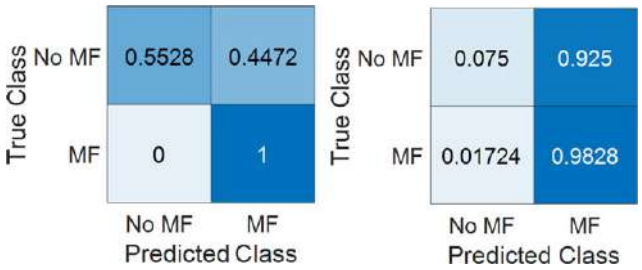


FIGURE 12 System-level major failure (MF) versus no MF: confusion matrices of training (left) and testing (right) sets using HOG features with SVM

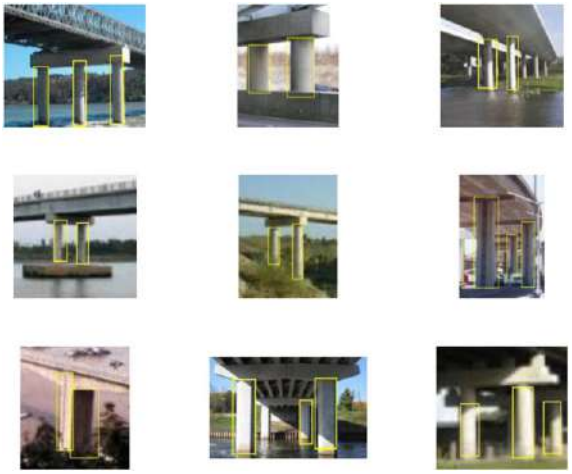


FIGURE 14 Sample testing images with bridge columns detected and localized by yellow rectangular bounding boxes produced by the proposed detector

i7-6700HQ @2.60 GHz, 16 GB DDR4 memory and 8 GB memory GeForce GTX 1080 external GPU), an Alienware Aurora R7 (a Core i7-8700 @3.20 GHz, 16 GB DDR4 memory and 8 GB memory GeForce GTX 1080 GPU), and an XPS 8930 (a Core i7-8700k @3.70 GHz, 32 GB DDR4 memory and 8 GB memory GeForce GTX 1080 GPU).

4.1 | System-level failure classification

As mentioned previously, the selection of VGG-16 as the pretrained model is made through the comparisons with two other pretrained models. Table 4 gives the testing accuracies of the three pretrained models where the values of the hyperparameters (also recorded in Table 4) are determined using Bayesian optimization. The VGG-16, the deepest architecture of the three, achieves the highest testing accuracy where its proneness to overfitting, under the small data set, is well addressed through the hyperparameters' selection. Figure 9 shows the loss and accuracy curves (vs. the numbers of iterations) for the training and the validation sets of images, from which

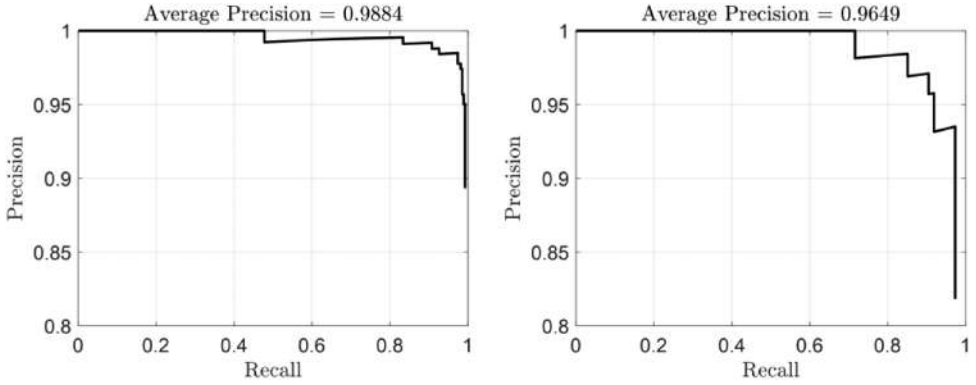


FIGURE 13 Component-level bridge column detection: recall-precision curve of training (left) and testing (right) sets of images using the values of hyperparameters determined by Bayesian optimization

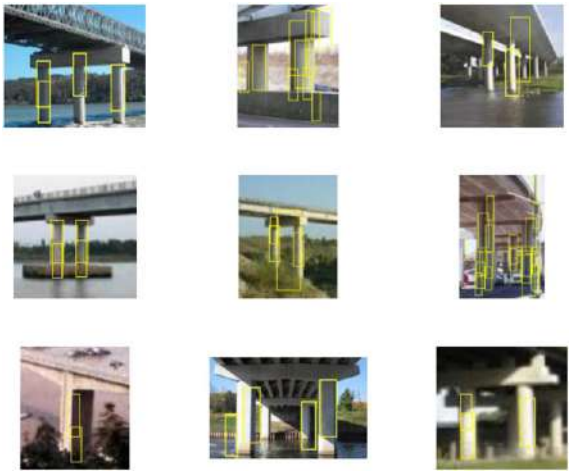


FIGURE 15 Sample testing images with bridge columns detected and localized by yellow rectangular bounding boxes produced by the Viola–Jones algorithm with HOG features

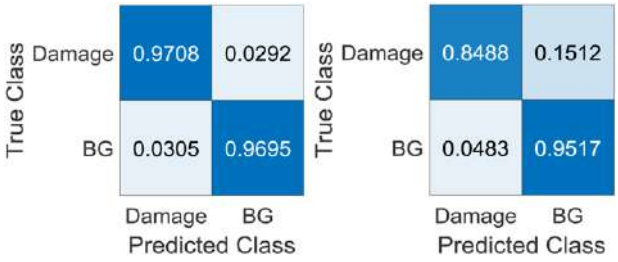


FIGURE 16 Local-level damage versus background (BG): confusion matrices of training (left) and testing (right) sets

the model with Bayesian optimization is observed to possess good robustness. Table 5 records the values of the three hyperparameters (with and without Bayesian optimization) for all the three models. The benefits of Bayesian optimization are evident through the comparisons of the achieved testing accuracies in the last row of Table 5. Figure 10 gives the confusion matrices (Kohavi & Provost, 1998) for both training and testing sets of images to demonstrate the accuracies and misclassification errors for each scenario. For example, for the testing set of images, given that the true classes are a major failure, the model prediction accuracy is 98.28%.

Figure 11 demonstrates sample testing images together with their predicted classes and the probabilities of those classes. Considering that the model is trained using a small data set (~350 images), such accuracies indicate the efficacy of transfer learning that imparts previous learning experience on object classification to the problem of scenario classification in civil engineering. Table 6 documents the computational time of training and execution time for the three models. For comparison, this classification is also conducted using histogram of oriented gradient (HOG) features (Dalal & Triggs, 2005) and an SVM classifier. The training and testing accuracies are, respectively, 81.73% and 61.22% (the confusion matrices are given in Figure 12), which are much lower than the achieved ones of the CNN model with Bayesian optimization.

4.2 | Component-level bridge column detection

In this subsection, the results regarding the bridge column detection are presented. To evaluate the performance of an object detector, the precision-recall curve is usually used. It is computed based on the ranked output (Everingham, Van Gool, Williams, Winn, & Zisserman, 2010) and is used to show the trade-off between the precision and the recall considering different threshold. A high area under the curve represents both high recall and high precision, where a high precision relates to a low false positive rate and a high recall relates to a low false negative rate. High scores for both certify that the detector is returning accurate results (a high precision) as well as returning a majority of all positive results (a high recall). A system with a high recall but a low precision returns many results, but most of its predicted labels are incorrect when compared to ground-truth labels. A system with a high precision but a low recall is just the opposite, most of its predicted labels are correct when compared to

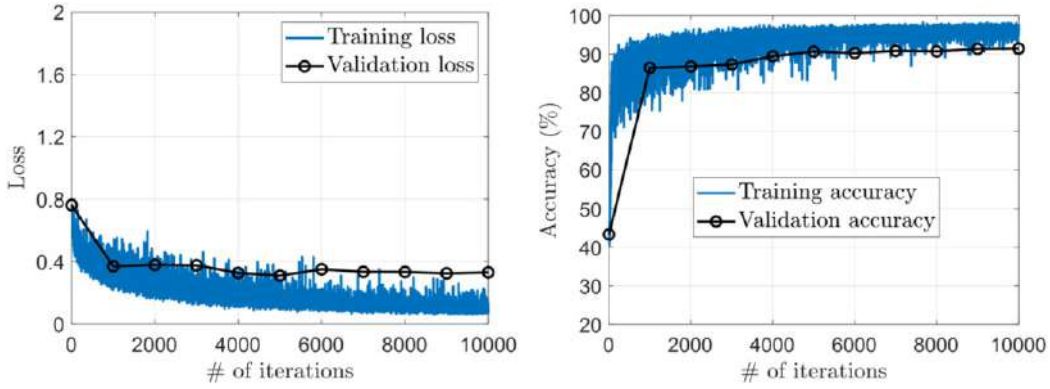
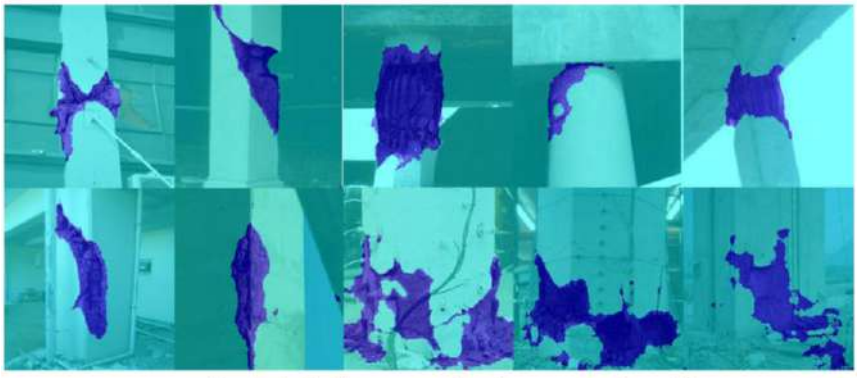


FIGURE 17 Local damage localization: loss (left) and accuracy (right) curves for training and validation sets of images using the values of hyperparameters determined by Bayesian optimization

a) Original images

b) Ground-truth labels

c) Labeled images using Otsu's method



d) Labeled images using the proposed model

FIGURE 18 Sample testing images used in local-level damage localization

ground-truth labels but returns very few results. Therefore, an ideal system with a high precision and a high recall will return many results, with all results labeled correctly. In short, the precision is "how useful the search results are" and the recall is "how complete the results are." Average precision (AP) summarizes the precision-recall plot as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight. Figure 13 shows such plots for both training and testing sets of images with their corresponding APs (well above 95%) in which the three hyperparameters are determined (reported in Table 5) through Bayesian optimization. Figure 14 gives several sample testing images with bridge columns detected and localized by the detector using the rectangular bounding boxes. These two plots well demonstrate the accuracy (a high precision) and the completeness (a high recall) of the detector. For example, in the plot for the testing set, the precision is still well above 80% when the recall is nearly perfect (∼98%).

For comparison, the task of bridge column detection is also conducted using HOG features with the Viola–Jones algorithm (Viola & Jones, 2001). Compared to the proposed column detecter based on faster R-CNN, this method does not work effectively as in Figure 15 where the detection results are presented for the same testing images as in Figure 14.

4.3 | Local-level damage localization

In this subsection, several metrics of resulting local damage localization are reported. The global accuracies regarding all pixels for the training and testing sets of images are, respectively, 96.97% and 93.14% (the three hyperparameters are determined through Bayesian optimization and reported in Table 5) where the confusion matrices are given in Figure 16. Figure 17 presents the loss and accuracy curves for training and validation sets of images, indicating a good model robustness. The IoU, also known as the Jaccard similarity coefficient, is the most commonly used in benchmarking (Csurka, Larlus, & Perronnin, 2013). For each class, the IoU is the ratio of correctly classified pixels to the total number of ground-truth and predicted pixels in that class. In the case that images have disproportionally sized classes, the weighted IoU (average IoU of each class, weighted by the number of pixels in that class) is used such that the impact of errors in the small classes on the aggregate quality score is reduced. The weighted IoUs for the training and testing sets are 94.32% and 87.65%, respectively. Figure 18 gives the qualitative comparisons between the achieved labels and ground-truth ones for several testing images (the original images are also presented), in which a good consistency, in general, can be observed. Also presented in Figure 18 are the results using the classical Otsu's method (Otsu, 1979). Apparently, it does not work effectively for this task with the real-world images, probably due to the distraction (e.g., shadows) in the background of the image. It is noted that the accuracies for the training and testing sets of images for all the three developed models are comparable. Such good robustness of the models is attributed to the use of Bayesian optimization for the hyperparameters' tuning regarding the model robustness.

5 | CONCLUSIONS

The inevitable aging of bridge structures, as one of the most critical components in transportation infrastructure systems, makes them vulnerable to future extreme events. Therefore, after extreme events, rapid and efficient condition screening of these lifeline structures is critical for rehabilitation of infrastructures and thus the resiliency of their local communities. This article, taking advantage of deep learning with novel training strategies, proposes a three-level image-based approach for post-disaster bridge inspection. The three levels are, respectively, for the system-level failure analysis, the structural component-level detection, and local-level damage localization. Three corresponding deep learning models are developed sequentially with the training strategy that the trained parameter values of the previous model are fed to the following one for training initialization. The hyperparameters' selection is vital to the training and model robustness, especially for the small data set considered in this study. A principled manner of such selection is proposed based on Bayesian optimization. It embraces the premise that the generalization performance of the model is to act as a sample from a Gaussian process. The benefits of the proposed approach are evident through comparisons between the testing accuracy of the models with Bayesian optimization adopted and the ones with the conventional training strategy. Comparisons are also conducted between the performance of the developed models and that using the classical computer vision techniques. The obvious performance enhancement offers the possibility to enable post-disaster autonomous inspection for near-real-time damage detection and assessment. The proposed approach is applicable to identifying damages in other structural components of the bridge when their corresponding labeled training data are available. For such future extensions, just as the role of VGG-16 in this study, the developed models in this article can serve as the pretrained models for transfer learning.

REFERENCES

Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., & Inman, D. J. (2017). Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, *388*, 154–170.

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

ASCE, Infrastructure Report Card (2017). Retrieved from https://www.infrastructurereportcard.org/

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(10), 2481–2495.

Bishop, C.M. (2006). *Pattern recognition and machine learning*. New York, NY: Springer.

Brochu, E., Cora, V. M., & De Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, *arXiv preprint arXiv:1012.2599*.

Calderone, A., Lehman, D. E., & Moehle, J. P. (2001). *Behavior of reinforced concrete bridge columns having varying aspect ratios and varying lengths of confinement*. Berkeley, CA: Pacific Earthquake Engineering Research Center.

Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, *32*(5), 361–378.

Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018). Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering*, *33*, 9–11.

Chavali, N., Agrawal, H., Mahendru, A., & Batra, D. (2015). Object-proposal evaluation protocol is "gameable." *arXiv*: *1505.05836*.

Csurka, G., Larlus, D., & Perronnin, F. (2013). What is a good evaluation measure for semantic segmentation? In *Proceedings of the British Machine Vision Conference*, 32.1–32.11. http://doi.org/10.5244/C.27.32

Dai, H. (2017). A wavelet support vector machine-based neural network meta model for structural reliability assessment. *Computer-Aided Civil and Infrastructure Engineering*, *32*, 4, 344–357.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR'05*. Vol. 1, San Diego, CA: IEEE, 886–893. https://doi.org/10.1109/CVPR.2005.177

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Li, F. (2009). ImageNet: A large-scale hierarchical image database. In *IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, San Diego, CA: IEEE, 248–255.

Eigen, D., & Fergus, R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of IEEE International Conference on Computer Vision*, San Diego, CA: IEEE, 2650–2658.

Esmaeily, A., & Xiao, Y. (2005). Behavior of reinforced concrete columns under variable axial loads: Analysis. *ACI Structural Journal*, *102*(5), 736–744.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The Pascal visual object classes (VOC) challenge. *International Journal on Computer Vision*, *88*(2), 303–338.

Farrar, C. R., & Worden, K. (2012). *Structural health monitoring: A machine learning perspective*. New York, NY: John Wiley & Sons.

Feng, C., Liu, M. Y., Kao, C. C., & Lee, T. Y. (2017). Deep active learning for civil infrastructure defect detection and classification. In *Computing in Civil Engineering 2017*, 298–306.

Feng, D., & Feng, M. Q. (2017). Experimental validation of cost-effective vision-based structural health monitoring. *Mechanical Systems & Signal Processing*, *88*, 199–211.

Fernandez Galarreta, J., Kerle, N., & Gerke, M. (2015). UAV-based urban structural damage assessment using object-based image analysis and semantic reasoning. *Natural Hazards and Earth System Sciences*, *15*(6), 1087–1101.

Gao, Y, & Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, *33*, 372–388.

German, S., Brilakis, I., & DesRoches, R. (2012). Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety assessments. *Advanced Engineering Informatics*, *26*(4), 846–858.

Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, 1440–1448.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 580–587.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge, MA: MIT Press.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hong, S., Noh, H., & Han, B. (2015). Decoupled deep neural network for semi-supervised semantic segmentation. In *Proceedings 28th International Conference on Neural Information Processing Systems*, 1495–1503.

Hosang, J., Benenson, R., Dollár, P., & Schiele, B. (2015). What makes for effective detection proposals? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(4), 814–830.

Hosang, J., Benenson, R., & Schiele, B. (2014). How good are detection proposals, really? Presented at *Proceedings of British Machine Vision Conference*, Nottingham, England.

Hoskere, V., Narazaki, Y., Hoang, T., & Spencer Jr., B. (2018). Vision-based structural inspection using multiscale deep convolutional neural networks. *arXiv preprint arXiv:1805.01055*.

Jahanshahi, M. R., Kelly, J. S., Masri, S. F., & Sukhatme, G. S. (2009). A survey and evaluation of promising approaches for automatic image-based defect detection of bridge structures. *Structure and Infrastructure Engineering*, *5*(6), 455–486.

Khoshelham, K., Elberink, S. O., & Xu, S. (2013). Segment-based classification of damaged building roofs in aerial laser scanning data. *IEEE Geoscience and Remote Sensing Letters*, *10*(5), 1258–1262.

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., & 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.

Koch, C., Paal, S. G., Rashidi, A., Zhu, Z., König, M., & Brilakis, I. (2014). Achievements and challenges in machine vision-based inspection of large concrete structures. *Advances in Structural Engineering*, *17*(3), 303–318.

Kohavi, R., & Provost, F. (1998). Confusion matrix. *Machine Learning*, *30*(2-3), 271–274.

Koziarski, M., & Cyganek, B. (2017). Image recognition with deep neural networks in presence of noise: Dealing with and taking advantage of distortions. *Integrated Computer-Aided Engineering*, *24*(4), 337–349.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 1097–1105.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Li, R., Yuan, Y., Zhang, W., & Yuan, Y. (2018). Unified vision-based methodology for simultaneous concrete defect detection and geolocalization. *Computer-Aided Civil and Infrastructure Engineering*, *33*(7), 527–544. https://doi.org/10.1111/mice.12351

Lin, Y. Z., Nie, Z. H., & Ma, H. W. (2017). Structural damage detection with automatic feature-extraction through deep learning, *Computer-Aided Civil and Infrastructure Engineering*, *32*, 12, 1025–1046.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. Cambridge, MA: The MIT Press.

Nabian, M. A., & Meidani, H. (2018). Deep learning for accelerated reliability analysis of transportation networks. *Computer-Aided Civil and Infrastructure Engineering*, *33*(6), 443–458.

Nagi, J., Ducatelle, F., Di Caro, G. A., Cireşan, D., Meier, U., Giusti, A., … Gambardella, L. M. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *IEEE International Conference on Signal & Image Processing Applications (ICSIPA)*, 342–347.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 807–814.

Narazaki, Y., Hoskere, V., Hoang, T. A., & Spencer Jr., B. F. (2018). Automated vision-based bridge component extraction using multiscale convolutional neural networks. *arXiv preprint arXiv:1805.06042*.

Noh, H., Hong, S., & Han, B. (2015). Learning deconvolution network for semantic segmentation. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 1520–1528.

Ortega-Zamorano, F., Jerez, J. M., Gómez, I., & Franco, L. (2017). Layer multiplexing FPGA implementation for deep back-propagation learning. *Integrated Computer-Aided Engineering*, *24*(2), 171–185.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 62–66.

Rafiei, M. H., & Adeli, H. (2017). A novel machine learning based algorithm to detect damage in highrise building structures. *The Structural Design of Tall and Special Buildings*, *26*, 18. https://doi.org/10.1002/tal.1400

Rafiei, M. H., & Adeli, H. (2018). A novel unsupervised deep learning model for global and local health condition assessment of structures. *Engineering Structures*, *156*, 598–607.

Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2017). Supervised deep restricted Boltzmann machine for estimation of concrete compressive strength. *ACI Materials*, *114*(2), 237–244.

Rasmussen, C. E., & Williams, C. (2006). *Gaussian processes for machine learning*. Cambridge, MA: MIT Press.

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149.

Rose, P., Aaron, B., Tamir, D. E., Lu, L., Hu, J., & Shi, H. (2014). *Supervised computer-vision-based sensing of concrete bridges for crack-detection and assessment* (No. 14–3857).

Sideris, P. (2012). *Seismic analysis and design of precast concrete segmental bridges*. Thesis, State University of New York at Buffalo.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv*:1409.1556.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, Lake Tahoe, NV.

Soukup, D., & Huber-Mörk, R. (2014). Convolutional neural networks for steel surface defect detection from photometric stereo images. *International Symposium Visual Computing*, New York, NY: Springer International Publishing, 668–677.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Suh, G., & Cha, Y. J. (2018, March). Deep faster R-CNN-based automated detection and localization of multiple types of damage. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018* (Vol. 10598, pp. 105980T). International Society for Optics and Photonics.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2818–2826.

Torok, M. M., Golparvar-Fard, M., & Kochersberger, K. (2013). Image-based automated 3D crack detection for post-disaster building assessment. *Journal of Computing in Civil Engineering*, *28*(5), A4014004.

Uijlings, J. R., Van de Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. *International Journal of Computer Vision*, *104*(2), 154–171.

Veletzos, M., Panagiutou, M., Restrepo, J., & Sahs, S. (2008). *Visual inspection & capacity assessment of earthquake damaged reinforced concrete bridge elements* (No. CA08-0284). California Department of Transportation, Division of Research and Innovation.

Vetrivel, A., Gerke, M., Kerle, N., Nex, F. C., & Vosselman, G. (2018). Disaster damage detection through synergistic use of deep learning and 3D point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning, *ISPRS Journal of Photogrammetry and Remote Sensing*, *140*, 45–59.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. *CVPR 2001*, Kauai, HI, December 8-13, 2001.

Xue, Y. D., & Li, Y. C. (2018). A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects. *Computer-Aided Civil and Infrastructure Engineering*, *33*, 8.

Yamaguchi, T., & Hashimoto, S. (2010). Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Machine Vision and Applications*, *21*(5), 797–809.

Yeum, C. M., & Dyke, S. J. (2015). Vision-based automated crack detection for bridge inspection. *Computer-Aided Civil and Infrastructure Engineering*, *30*(10), 759–770.

Yeum, C. M., Dyke, S. J., Ramirez, L., & Benes, B. (2016). Big visual data analysis for damage evaluation in civil engineering. In *International Conference on Smart Infrastructure and Construction*, Cambridge, U.K., June 27–29.

Yoon, H., Elanwar, H., Choi, H., Golparvar-Fard, M., & Spencer, B. F. (2016). Target-free approach for vision-based structural system identification using consumer-grade cameras. *Structural Control & Health Monitoring*, *23*(12), 1405–1416.

Zhang, A., Wang, K. C. P., Li, B., Yang, E., Dai, X., Peng, Y., … Chen, C. (2017). Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. *Computer-Aided Civil and Infrastructure Engineering*, *32*(10), 805–819.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., … Torr, P. H. S. (2015). Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 1529–1537.

Zhu, Z., & Brilakis, I. (2010). Concrete column recognition in images and videos. *Journal of Computing in Civil Engineering*, *24*(6), 478–487.

Zhu, Z., German, S., & Brilakis, I. (2011). Visual retrieval of concrete crack properties for automated post-earthquake structural safety evaluation. *Automation in Construction*, *20*(7), 874–883.

How to cite this article: Liang X. Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization. *Comput Aided Civ Inf*. 2018;1–16. https://doi.org/10.1111/mice.12425