DOI: 10.1111/mice.12698

INDUSTRIAL APPLICATION

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Similarity learning to enable building searches in post-event image data

Jongseong Choi[1,2] | **Ju An Park**[3] | **Shirley J. Dyke**[4,5] | **Chul Min Yeum**[3] | **Xiaoyu Liu**[4] | **Ali Lenjani**[4] | **Ilias Bilionis**[4]

[1] Department of Mechanical Engineering, The State University of New York (SUNY Korea), Incheon, South Korea

[2] Department of Mechanical Engineering, The State University of New York, Stony Brook University, Stony Brook, New York, USA

[3] Department of Civil and Environmental Engineering, University of Waterloo, Waterloo, Ontario, Canada

[4] School of Mechanical Engineering, Purdue University, West Lafayette, Indiana, USA

[5] Lyle School of Civil Engineering, Purdue University, West Lafayette, Indiana, USA

Correspondence
Jongseong Choi, Department of Mechanical Engineering, The State University of New York, SUNY Korea, Incheon, South Korea.
Email: jongseong.choi@stonybrook.edu

Funding information
National Science Foundation OAC Program, Grant/Award Number: OAC-1835473; National Research Foundation of Korea; Korea government(MSIT), Grant/Award Number: 2021R1G1A1012298

Abstract

Reconnaissance teams collect perishable data after each disaster to learn about building performance. However, often these large image sets are not adequately curated, nor do they have sufficient metadata (e.g., GPS), hindering any chance to identify images from the same building when collected by different reconnaissance teams. In this study, Siamese convolutional neural networks (S-CNN) are implemented and repurposed to establish a building search capability suitable for post-disaster imagery. This method can automatically rank and retrieve corresponding building images in response to a single query using an image. In the demonstration, we utilize real-world images collected from 174 reinforced-concrete buildings affected by the 2016 Southern Taiwan and the 2017 Pohang (South Korea) earthquake events. A quantitative performance evaluation is conducted by examining two metrics introduced for this application: Similarity Score (SS) and Similarity Rank (SR).

1 | INTRODUCTION

Structural reconnaissance teams are deployed after each natural disaster to gather a vast quantity of scientific data in the form of images. These perishable data are meant to support vital researches focused on identifying gaps in construction and design practices and advancing improvements in building codes (American Concrete Institute, 2017; Monical, 2020). The investment in gathering reconnaissance data after natural disasters is growing exponentially. In 2016, the National Science Foundation (NSF) established a shared-use large facility known as the Natural Hazards Engineering Research Infrastructure (hereafter, NHERI) that is dedicated to research in hazards and resilience. The NHERI network has two facilities explicitly designed to support such data collection: (i) the RAPID facility at the University of Washington (Designsafe-CI, 2016), and (ii) a data repository, DesignSafe-CI, at the University of Texas, Austin (DesignSafe, 2016). Several similar data repositories do exist, for instance, through the

Earthquake Engineering Research Institute (EERI), Canterbury Earthquake Digital Archive, QuakeCore (in New Zealand), DataCenterHub (at Purdue University), and the Pacific Earthquake Engineering Research Center (PEER, at the University of California, Berkeley) (Datacenterhub, 2014; EERI, 2009; PEER, 2013; QuakeCoRE, 2016; UC CEISMIC, 2012).

Despite the massive investments made to acquire and host these data, current repositories and manual search procedures are inadequate for effectively organizing these data for domain use and efficiently conducting scientific research. They are voluminous with a wide variety of unstructured and complex images. Since they are frequently reformatted or resized before publication or distribution, geotagging or time-mapping the images with metadata (EXIF) information is often not available or is incorrect (e.g., GPS data). Consequently, the usability of these data is significantly diminished since a large portion of the images remains uncurated.

Several vision-based post-disaster evaluation methods have been developed and published over the last few years. Some of these capabilities are embedded within our automated reconnaissance image organizer (ARIO), an online tool for automatically generating reconnaissance reports with pre-trained classifiers. The capability of ARIO has been demonstrated on a post-disaster dataset in the last decade (Yeum et al., 2018, 2019). A rapid post-event assessment technique was also developed, with which building façades may be rapidly inspected using a large volume of aerial images collected from UAVs (Choi et al., 2018). A streamlined lifecycle visual inspection of landmark buildings was achieved by automating data collection procedures through the open-source visual data crowdsourced from citizens (Choi & Dyke, 2020). The capability of comparing pre- and post-disaster building damage has been demonstrated by exploiting Google Streetview imagery (Lenjani, et al., 2020a, 2020b). Recently, a complementary technique that automatically generates an inspector's indoor path was developed using a video footage captured during a reconnaissance mission (X. Liu et al., 2020).

The majority of recent vision-based analytics and algorithms dealing with post-disaster imagery are based on image recognition. Post-disaster inspection of a reinforced concrete (RC) bridge has been proposed, which performs image classification, object detection, and semantic segmentation, with hyperparameter selection based on Bayesian optimization (Liang, 2019). An efficient edge computing method was achieved by pruning convolutional neural networks (CNNs) and Taylor expansion for use in a robotic inspection of an infrastructure (Wu et al., 2019). Deep learning-based damage detection approaches using Faster R-CNN were introduced for post-disaster reconnaissance (Mondal et al., 2020; Ren et al., 2015). Several building recognition systems have been proposed in the last decade (Kokare et al., 2003; Li et al., 2014). A deep learning-based image retrieval method was proposed to search various building scenes and its performance was evaluated with mean Average Precision (mAP) (Gordo et al., 2016). However, such approaches may be further strengthened if there is a capability to identify each of the individual buildings (Iqbal & Aggarwall, 1999). None of the past studies has developed a method for rapidly retrieving and connecting different images of the same building from a large pool of unorganized post-disaster images. This gap exists because it is infeasible to use a pre-designed category structure for identifying images of the same building. For example, a classifier may be trained to identify several buildings-of-interest by using a predefined set of classes in a multiclass structure (e.g., building A, building B, and building C). However, if there arises a new building(s)-of-interest, this classifier would not be able to identify the new buildings immediately, and to do so the predefined set of classes would necessitate a redefinition of the classes to include the new building and retraining. The classifier infers a function from labeled training data consisting of a set of example of categories (Mohri et al., 2018). Thus, existing image classification approaches do not perform this function of organizing and gathering the images of buildings.

To address this challenge, we have developed an approach that, when given a query image of the building-of-interest, will rapidly search throughout the entire contents of a given database for images of the same building. The approach is to filter the database to extract a ranked set of images that are potential matches for the query image, called building overview images (BOVs). Each of these BOVs is compared with the query image to compute a similarity metric based purely on visual contents. BOVs are the images that capture an overall view of an entire building from different viewpoints and locations (Yeum et al., 2019). In typical building reconnaissance missions, BOVs are often the first set of images captured of a building during an initial post-event survey and are immediately followed by additional images captured at a closer distance to the building for a detailed study (e.g., building components or indoor damage). In most cases, the images from a reconnaissance mission are organized by building, thus storing BOVs and the detailed images of a building in the same location. In other words, BOVs can be used as a visual tag, allowing users to search for the same building images from an uncurated image database. Identifying the stored locations of BOVs in a database also increases the likelihood of finding additional images associated with the query building. Furthermore, BOVs provide unique visual information about individual buildings that can be used to differentiate them from others. Thus, BOVs are a key enabler to achieve an automated building search capability by extracting and

incorporating visual features present on the BOVs. The one-shot learning capability of S-CNNs is ideally suited to retrieve similar images using just a small quantity of BOVs to train classifiers (Qi et al., 2016; Vinyals et al., 2016).

In this paper, we develop the capability to search the entire contents of a given database for images of the same building as a user's query BOV image. The input to such a system is the query image, and the output is a ranked set of retrieved images containing images of the buildings that are most similar to the query image. This approach is capable of generating a sorted set of images based on a similarity metric, which allows a user to view and choose among them. This capability will enable the rapid integration of multiple datasets of the same building even when collected at different times by different reconnaissance teams. This capability directly supports scientific research by expanding the image data available to study the performance of a given building under one of more events. We implement this task-oriented image retrieval system using S-CNN that combines feature extraction and metric learning into a single framework (Bromley et al., 1994; Gordo et al., 2016; Hadsell et al., 2006; Koch et al., 2015). By training on pairs of BOVs, which include both pairs of the same and pairs of different buildings, the S-CNN can automatically rank building images corresponding to the query image. By establishing this powerful search capability, researchers, engineers, and scientists can efficiently utilize the vast amounts of data being collected at great expense, and rapidly transform visual data into knowledge about our built environment. This novel approach would be best suited for data searches in large databases. Quantitative performance evaluation of the machine is originally designed for this domain application and is conducted by examining the values for: (1) similarity score (hereafter, SS) among true-matches and false-matches, (2) similarity rank (hereafter, SR) distribution of true-matches with 251 different queries, (3) SR performance per 30 different test buildings, and (4) the probability that at least one true-match exists within the SR of the top 10 using our method. Additionally, the performance of the model is evaluated by considering a binary classification metric by considering both SS and SR as thresholds.

2 | TECHNICAL APPROACH

The method developed herein is intended to provide the power to search across an image database to gather images from the query building. We assume databases contain a wide variety of scenes collected from many different buildings in the form of images (e.g., building overview, indoors, outdoor scenes, columns, walls, etc.), and that a large number of these images remain unlabeled and unclassified. The technique automatically ranks the BOVs for a corresponding building query using a similar searching capability to identify images from the same building.

An overview of the approach is illustrated in Figure 1. Implementation of the approach has four steps: Step 1, the user query as the input; Step 2, BOV classification to extract all BOV images from the database; Step 3, similarity score computation (SS); and Step 4, BOV retrieval based on similarity rank (hereafter, SR). In the user query step, we characterize the contents of the query image provided by the user with a single vector denoted μ. This step exploits convolutional layers trained with a contrastive loss function based on S-CNNs (Bromley et al., 1994; Hadsell et al., 2006; Koch et al., 2015). To perform image retrieval, BOVs are first extracted from the reconnaissance database using our robust BOV classifier. A second vector ν_i, which characterizes each BOV using the same trained convolutional layers, is then compared to the vector μ that is obtained from the user query image. These two vectors are then used to compute their visual similarity, denoted by $f(\mu, \nu_i)$, yielding SS_i iteratively for each BOV found within the database. Note that Steps 2 and 3 are not necessarily conducted every time. When the user plans to test multiple query buildings, the BOVs and corresponding ν vector are stored in the database and reused without repeating these steps. Finally, BOVs from the database are ranked by SS, allowing users to access images of the query building from the database, assuming that such images do exist.

Before the details of each step are discussed in the following subsection, we emphasize that the proposed image retrieval model does not have to be trained using images from the reconnaissance database being searched. The image retrieval model can be applied to any existing reconnaissance image database. This strategy is possible since the model extracts general building features to assess the visual similarity between a query BOV and all other BOVs, rather than searching for the query building that the model has already seen. Thus, the model is not trained to extract specific features that can characterize individual buildings in the training dataset, which is the approach commonly used in developing image classification models. This is the key difference between conventional CNNs and S-CNNs which allows this technique to achieve a high level of generalization.

2.1 | User query

In developing the approach, we use BOVs as the query image because it contains visual features that can uniquely determine the individual building. The method is intended for the query image to be the one captured by another reconnaissance team and it can be taken either pre-event

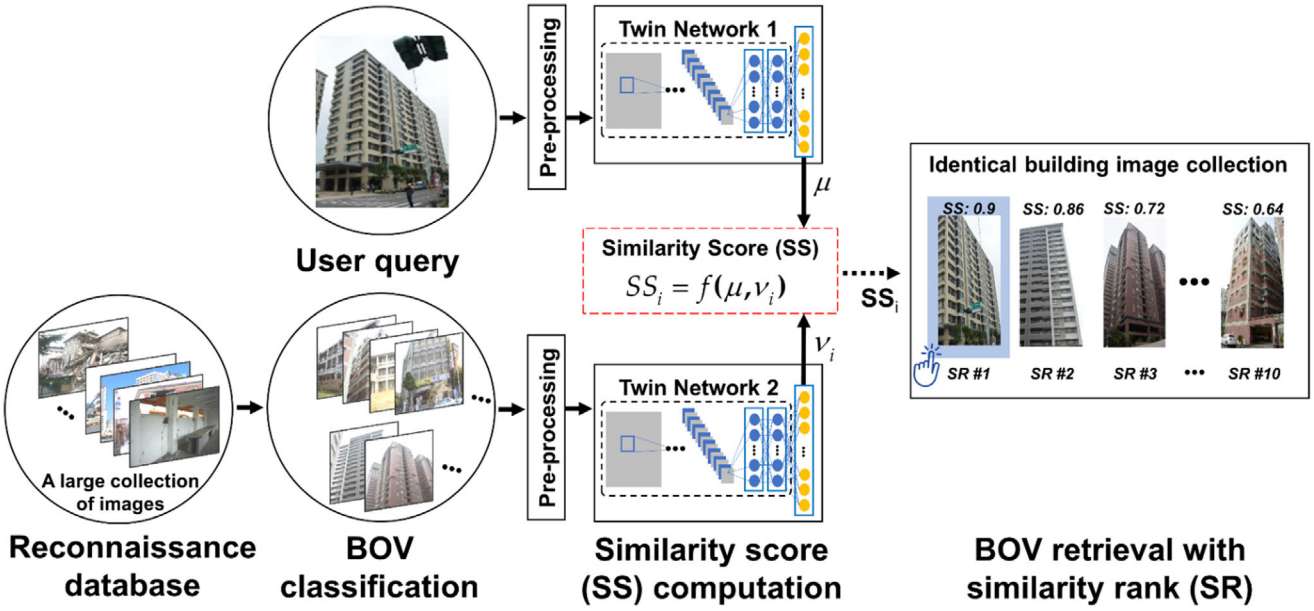

FIGURE 1 Overview of the building search method: Step 1: User query; Step 2: BOV classification; Step 3: Similarity score (SS) computation; and Step 4: BOV retrieval with similarity rank (SR)

FIGURE 2 Samples of building overview images (BOVs)

or post-disaster. BOVs of severely damaged buildings that are unrecognizable would not be suitable in this work. Sample BOV images are shown in Figure 2. Herein, BOVs are defined as images that contain the complete external appearance of the building, usually from a distance, and include one entire side view or a front view of the building and have at least 70% of the building being visible in the image even though partial obstructions may exist. For example, images that contain the entire building façade, a canonical view of the building or one complete side of the building are possible examples (Yeum et al., 2019). BOVs are an essential and common type of scene used in reconnaissance data collection because they are typically used to identify the building that is under investigation before any other images containing specific damage information or structural details are captured. Thus, such an image can act as a visual tag to identify and link the same building across the database.

3 | BOV CLASSIFICATION

Reconnaissance data collected for evaluating the performance of our infrastructure generally includes complex, unordered, and unstructured scenes. These teams need to gather a large volume of perishable data in a short period of time in a way that does not interrupt recovery efforts or harm the residents of that region. The visual data collected focuses on buildings and building components and include specific damage to columns and beams such as shear cracks, buckling, and spalling. Additionally, important metadata such as drawings, GPS devices, or

measurements (e.g., a column with a measuring tape) may be captured in the form of an image. Furthermore, the data may also contain a significant amount of irrelevant (e.g., people, random objects, vehicles) or even corrupted (e.g., blurred, noisy, dirt on the camera lens) images. In our previous work, a multiclass classifier was trained and successfully demonstrated our automated approach for classifying and documenting such post-earthquake reconnaissance images (Yeum et al., 2019). Field engineers were consulted in the development of appropriate visual data categories, including BOVs, along with an associated hierarchical structure to support the domain research needs (Yeum et al., 2018). Deep CNN algorithms were implemented to extract robust features of key visual contents in the images. The tool has been quite successful in classifying the images into pre-designed categories, of which BOV classification is one of its functions.

In this step, we extract BOVs from the database using the pre-trained image classifier developed in the previous study. Unlike the other classes in the schema, BOVs contain the visual characteristics of each individual building as a single image. Thus, with this classifier, all BOVs may be automatically extracted from the database for ready comparison with a query BOV, to be described in the following steps.

4 | SIMILARITY SCORE COMPUTATION

In the last decade, CNNs have enabled remarkable progress in the computer vision field and the associated use of large-scale databases for supporting vision-based applications (Adeli, 2020; Krizhevsky et al., 2012; LeCun et al., 1990). A CNN typically has one or more convolutional layers with tunable weights and pooling layers to extract features that are invariant to small scale, rotation and translation transformations, and fully-connected layers that interpret these features to classify image or object categories. Generally, a training phase is used where the goal is to learn a large number of CNN parameters in the convolutional and fully connected layer to minimize the loss to estimate true labels of a large number of training images. One solves this minimization problem through a variant of stochastic gradient descent. In data augmentation, raw images are resized and square-cropped to be transformed into the input size of the CNN, typically a square with low resolution. Furthermore, additional image augmentations are added to reduce model overfitting. The augmentations used herein include random rotations of ±10 degrees, zoom of ±20% (zoom out/in), brightness changes of ±20%, contrast changes of ±20%, and saturation changes of ±10%. After passing through the entire dataset, a batch is assigned using randomly ordered images (i.e., jittering) at each epoch. Several deep learning algorithms proposed recently have demonstrated excellent performance for particular applications (Alam et al., 2020; Pereira et al., 2020; Rafiei & Adeli, 2017). Their accuracy varies depending on how one configures the network architecture with input data. Researchers have been incrementally introducing new layer structures and network configurations to improve performance. The performance of state-of-the-art CNNs is often evaluated using one or more of the existing benchmark datasets (e.g., ImageNet, PASCAL). Thus, it has become common for these state-of-the-art CNNs to be applied in a wide variety of disciplines, where the parameters of the network are fine-tuned prior to usage. This approach is adopted because these networks are often quite effective for practical use even when applied to new datasets. Optimal network architectures for a certain image type remain a topic of research and are under development in many domains of application. However, CNNs are generally able to provide excellent performance for an object category classification and detection when dealing with natural images (Russakovsky et al., 2015).

Among the several possible implementations of CNNs, we focus on the class of CNNs known as the S-CNN to build an image search tool. Because the BOV retrieval task is not a classification problem, each building cannot be trained as a separate class. As a result, the S-CNN is used, which maps images to a learnable feature space (in this study, images mapped to this space are represented as a vector). For this study, the network learns to map same-building image pairs close to each other in the feature space while different building image pairs are mapped far from each other. After the query image and the rest of the BOV images in the database are mapped to the feature space, the image pairs are ranked according to the SS, which is computed as follows:

$$SS_i = f(u, v_i) = \frac{1}{1 + ||u - v_i||} \tag{1}$$

where u is the query image vector, v_i is the ith BOV image vector in a database computed using a trained S-CNN, and $\|u - v_i\|$ is the $L2$-norm of the difference between the two vectors (otherwise known as the Euclidian distance between two vectors). As a result, each SS has a range between 1 and 0. Image pairs with SS values close to 1 indicate a highly similar pair, while image pairs with SS values close to 0 indicate a highly dissimilar pair. The metric SS is used as the metric to simplify model performance analysis as it can be linearly derived from the $L2$-norm while conveniently having finite bounds from 0 to 1, whereas the $L2$-norm can have bounds from 0 to infinity. The key underlying assumption that enables the training of S-CNNs for this study is that a pair of BOVs captured from the same building but from different

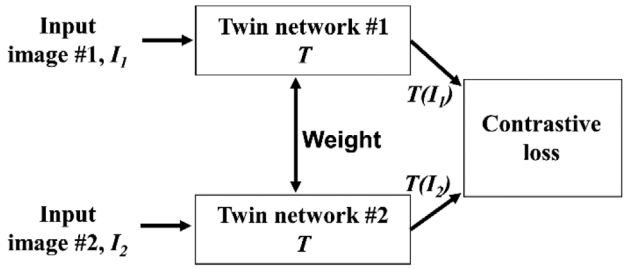

FIGURE 3 The architecture of a Siamese neural network (S-CNN) used in this research; contrastive loss function is used in S-CNN

viewpoints sharing common visual features that make them distinguishable from other buildings. The ground truth training set, known here as the "similarity ground-truth dataset," contains labeled images of the same buildings. The weights in the network are trained on randomly generated pairs from this dataset. The relevant capabilities of S-CNNs are that they: (1) learn to score visual relevance by mapping same building image pairs to close to each other, and different building image pairs far apart from each other in the feature space; and (2) learn from an extremely limited quantity of data, which is called one-shot learning. The S-CNN is not a brand-new concept, as it is widely used in various applications (e.g., facial re-identification (Chopra et al., 2005; Wang et al., 2016; Zhang et al., 2016), medical imagery (Y. Liu et al., 2017)). Here, for building identification, the S-CNN is assessed in terms of its ability to quantify the visual similarity between the input, the user query image, and each of the images in a database. Those pairings found to have the highest similarity to the query image that can be retrieved using this similarity score.

The details of the similarity learning network and our training schema are provided here. S-CNN consists of twin networks with shared weights (weight values and structure of each network are identical to each other) that take distinct inputs and output a vector in the learnable feature space. The vectors are then fed into a contrastive loss function rather than a classification loss function, as shown in Figure 3. The contrastive loss function contains dual terms, where the loss criteria for similar-pairs and dissimilar-pairs are different when optimizing the weights of the twin networks (Koch et al., 2015).

The S-CNN architecture is presented in detail in Hadsell et al. (2006). The parameters between the twin networks are identical and share the same weights, but each network accepts different image inputs, which are denoted I_1 and I_2. The twin networks consist of a sequence of identical convolutional layers, where the nodes in the final convolutional layer are flattened into a single vector and followed by two fully-connected layers. T represents the tunable convolutional and fully-connected weights in the twin networks used to map input images I_1 and I_2 to a feature space, shown as $T(I_1)$ and $T(I_2)$ in Figure 3. Afterwards, the resulting two vectors are fed into the contrastive loss function in Equation (2), which then optimizes the hidden vector T. The first fully-connected layer has 4096 nodes and is followed by the last fully-connected layer, which has 10 nodes and its output is used to compute the SS. Note the output feature space dimensions match with the number of nodes of the last dense layer, that is, it is 10. We empirically selected 10 output nodes after several trials using various output node sizes for optimal performance. The contrastive loss function takes the following form:

$$(1-Y)\frac{1}{2}D^2 + Y\frac{1}{2}\{\max(0, m-D)\}^2 \tag{2}$$

where D is defined as the $L2$-norm (e.g., Euclidean distance) between the two output vectors computed for the corresponding image pair using the twin networks. The $L2$-norm between the feature vectors of the two images is computed as follows:

$$D = ||T(I_1) - T(I_2)|| = \sqrt{\sum_{k=1}^{n}\left(T(I_1)_k - T(I_2)_k\right)^2} \tag{3}$$

where the subscript k indicates the kth element in vectors $T(I_1)$ and $T(I_2)$, and n is the number of output nodes (10 in this study). Thus, a large D indicates that the image pairs are very dissimilar, and when D is small, the image pairs are quite similar. In Equation (2), Y is 0 when both images in the pair are sampled from the same building, and 1 when each image in the pair has been sampled from different buildings. Thus, when Y is 0 (images are from the same building), the 2nd term in Equation (2) disappears and the loss function penalizes dissimilarity D between the image pair, and optimization brings the image pair closer to each other in the feature space. On the other hand, when Y is 1 (images are from different buildings), the 1st term in Equation (2) disappears and the loss function penalizes the negative dissimilarity (which can be understood as similarity) capped to a maximum value of the margin, m for training stability. Thus, optimization for this case pushes the image pair further away from each other from the feature space. The margin is in place to optimize the network by eliminating the case of abnormally high value of D among those that are labeled as "dissimilar pair." Dissimilar pairs with a value of visual similarity that is beyond this margin will not contribute to the loss. In essence, the network deems that dissimilar pairs with D greater than the margin are sufficiently far apart and do not need to be optimized. The optimal value for parameter m is usually dependent on the dimensionality of the network's output

vector which can be empirically derived through manual performance validation or automatically determined during training (Sun et al., 2014). Once the distance between dissimilar pairs is sufficiently broad (i.e., greater than the margin), that pair is ignored so that other dissimilar pairs with distances lower than the margin can contribute to the model training. The networks are optimized with the contrastive loss function using a similarity ground-truth dataset that can contain many similar and dissimilar pairs, which indicate two images from a single building and two different buildings, respectively.

The key to the successful training of the networks lies in setting the similarity ground-truth dataset. Here, many pairs from past reconnaissance field missions are prepared and used as ground-truth data, as discussed in Section 3. We assume that the visual similarity (score) between a pair of BOVs captured from a single building is a function of the relevant features. However, an inherent challenge to define the similarity metric is that the BOVs may contain entirely different sides of a single building some of which may not be visually similar although they are from the same building. To overcome this challenge, we further filter the training dataset in which similarity is defined by examples provided by simply matching handcrafted features between images instead of learning-based features (Chherawala et al., 2013). Specifically, we first group all BOVs for a given building. Then we extract the scale-invariant feature transform (SIFT) features with cross-matching to choose the two most similar images. We annotate those two images as a similar pair, generating a feature map to identify patterns by groupings of pixels of the image (Peker, 2011). In other words, not all BOVs captured from the same building are defined as similar pairs, rather only image pairs of the same building that share a certain number of these SIFT features are assigned to be similar pairs. For image pairs taken from the same building that do not share enough SIFT features, they are not assigned as dissimilar pairs but are simply removed from training. Dissimilar pairs are only annotated when BOVs are taken from different buildings. Thus, only annotated similar and dissimilar pairs are involved in the training process, and pairs of BOVs captured from the same building but without insufficient visual overlap are ignored. While SIFT is used in this paper, other feature detection algorithms (Bay et al., 2008; Leutenegger et al., 2011; Matas et al., 2004; Rosten & Drummond, 2005) could alternatively be used for filtering similar image pairs. The details of such cases are demonstrated with BOV examples and discussed in the following section.

As pre-processing for training, we apply several augmentation methods to the BOVs, including: (1) setting a possible minimum size of the square window to crop only the portion of the BOV image containing the building (this step directly supports minimizing errors caused by image ratio and resizing) and (2) using RGB color channels rather than greyscale images to allow for color to be considered in the matching process.

5 | BOV RETRIEVAL WITH SIMILARITY RANK

Using the trained twin networks, the SS is computed for each BOV in the database and the query image. The SS is computed as the logistic of the Euclidean distance, D, and falls within the range between 0 (least similar) and 1 (most similar). Then, BOVs are ranked based on the SS and the corresponding ranking values are used to develop SR.

The ideal scenario is that the BOVs captured from the query building have higher SS and lower SR (higher rank) values than their values from the other buildings. However, the actual implementation and its evaluation should be different; not all BOVs from a query building share common visual features with a single query image, as mentioned in the previous subsection. Furthermore, producing good SR values for all those images might not be a necessary condition to judge the performance of the application that needs this search capability. The application requires at least one of the BOV images be ranked high so that a researcher can immediately recognize the identical query building from the ranked BOVs. This process is quite subjective in generalizing what values are considered high and low as it may vary depending on the user's preferences as well as the size of the database used. Certainly, the performance of image retrieval will vary depending on the size of the database, quality of the images, and consistency in data collection procedures.

To generalize the method, we thus evaluate both the SS and SR values designated as *true-match* so that one can estimate its performance. Here, when the BOVs in the database are retrieved after a user query, we denote the BOVs corresponding to the user query as *true-matches* and all the other BOVs as *false-matches*. Thus, our final goal is to provide at least one *true-match* image to the user, that is, one image which was captured from the same building as the query image. Ideally, the method should assign BOVs that are a *true-match* high SS and low SR values (high ranking). In the following section, we demonstrate that our method can provide search and retrieval capabilities to link different data sets from the same building, which will support research in infrastructure performance.

6 | PERFORMANCE EVALUATION

We demonstrate and evaluate the similarity-based building search method using real-world examples considering

RC buildings captured in past earthquake reconnaissance missions. We select images of RC buildings in this performance evaluation study because: (1) RC buildings are the most prevalent type of buildings around the world; (2) a large number of images is available because a significant portion of reconnaissance databases focuses on RC buildings; and (3) many types of structural damage can be observed in RC buildings that practitioners and researchers may be interested in for further investigation (Datacenterhub, 2014; DesignSafe, 2016; EERI, 2009; PEER, 2013; QuakeCoRE, 2016; UC CEISMIC, 2012).

For this evaluation, we train S-CNN model to measure the visual similarity of BOVs using two reconnaissance image datasets, which were collected by several reconnaissance teams with different cameras and different camera settings: the 2016 Southern Taiwan earthquake and the 2017 Pohang earthquake, South Korea. Both datasets focus on the performance of RC buildings and include a variety of images and several BOVs for some of the buildings. We then conduct a quantitative evaluation of the performance of the similarity-based building search method by examining the values for: SS among *true-matches* and *false-matches*, SR distribution of *true-matches*, SR of *true-matches* per building, and the probability that at least one *true-match* exists within the SR of the top 10 using our method. Additionally, the performance of the model is evaluated considering a binary classification metric by changing the thresholds for either SS or SR.

7 | BOV DATASET

An extensive post-event reconnaissance image collection was developed by Yeum et al. (2018, 2019) for use in researches related to reconnaissance image classification and organization (Yeum et al., 2018, 2019). The database includes over 100,000 color images collected by structural engineering teams after natural disasters such as earthquakes, hurricanes, and tornados, then archived in various databases (e.g., Purdue University's datacenterhub.org, Canterbury Earthquake Digital Archive, and Earthquake Engineering Research Institute's reconnaissance archive) (Datacenterhub, 2014; EERI, 2009; UC CEISMIC, 2012). To investigate similarity classification, we focus on recent events and data from RC buildings including, the 2016 Southern Taiwan earthquake, with a total of 14,102 images, and the 2017 Pohang earthquake, South Korea, with a total of 4101 images (NCREE, 2016; Sim et al., 2017). Both datasets contain a variety of typical structural scenes such as building components (e.g., wall, columns, or beams), damage type (e.g., cracking, spalling, or collapse), and image location (e.g., building interior or exterior). Useful metadata are also commonly collected in the form of images during a reconnaissance mission (e.g., images of GPS devices, drawings, watches, or measurements).

The BOV dataset used in this evaluation is a subset of these two reconnaissance image collections. The step of BOV classification is well demonstrated and proved with the larger dataset in the authors' previous work, thus we do not demonstrate the performance of BOV classification in detail (Yeum et al., 2019). The resulting BOV dataset consists of 1332 BOVs of 151 RC buildings. Of these, 953 BOVs are from 97 buildings visited by field teams after the 2016 Southern Taiwan earthquake, and 379 BOVs are from 54 buildings visited by field teams after the 2017 Pohang earthquake, South Korea. The number of BOVs for each building varies considerably, ranging from 2 to upwards of 24 images. All of these BOVs were collected with commercial DSRL cameras of which resolutions range from 10 to 24 megapixels. Each building in the database was reviewed and found that the buildings have a distinct characteristic such as color, shape, size, the number of stories or window arrangements. These distinct characteristics allow S-CNN to extract building features sets to evaluate the similarity of the buildings. BOVs contain a partial or entire building exterior view as well as different viewpoints. BOVs from the same building can contain many perspectives and different sides of the building.

8 | MODEL TRAINING

We combine two datasets, the 2016 Southern Taiwan earthquake and the 2017 Pohang earthquake, South Korea, and split it into training and testing sets where buildings from both datasets are intermixed into either set instead of training on one dataset and then testing on the other dataset. While cross-validation is known to provide an unbiased estimate of prediction error, it is also known that its variance may be very large (Breiman, 1996). When a model is trained using images from just one isolated region, the network is likely to extract and learn features about the regional building styles. Intermixing datasets helps generalizing the model and enhancing its robustness by avoiding the network trained with the regional building styles from one isolated dataset.

For training the model that determines the SS, we set aside around 80% of the total data by buildings: 1081 BOVs of 121 buildings. We initially create BOV pairs using cross-matching among all of the 1081 BOVs, yielding a total of 583,740 pairs after excluding self- and commutative-pairs (e.g., $(1081 \times 1081 - 1081)/2$). Here, we highlight that the images used for training and testing were captured from entirely different buildings, thus these sets do not share any images. Specifically, images from one building are either only in the training dataset or only in the testing

FIGURE 4 Samples of (a) similar pair, (b) different pair, and (c) dissimilar pair

TABLE 1 Composition of train and test dataset

	Train			Test		Total
No. of building	121			30		151
No. of BOV	1081			251		1332
No. of pairs	583740			31375		–
	Similar pairs	*Different pairs*	*Dissimilar pairs*	*True-matches*	*False-matches*	
	2544	574887	6304	1477	29898	

dataset and are never split between the training and testing datasets. To designate *similar* and *dissimilar pairs*, we first exclude any BOV pairs that contain two images captured from a single building but contain two entirely different façades, named here as *different pairs*. The number of SIFT feature matching is used to distinguish between *similar* and *different* pairs. The definitions of each category criteria are as follows: (1) *similar pair*: BOV pairs that are captured from the same building with similar viewpoints and sharing common visual features; (2) *different pair*: BOV pairs that are captured from the same building but contain different regions (or sides) of the building. The visual appearance of the buildings in these pairs does not overlap much; and (3) *dissimilar pair*: pair that is captured from two distinct buildings. Sample images for these pairs are present in Figure 4. We used "vl_ubcmatch" in the VLFEAT open-source computer vision library to match descriptors from SIFT features from a pair of images (Vedaldi & Fulkerson, 2010). If the number of matched features is greater than 10, we assign the pairs as similar pairs, otherwise they are assigned as different pairs. Finally, while any *different pairs* are disregarded, *similar* and *dissimilar pairs* are used to train the model. This is because pairs defined as different pairs are overall weak in visual similarity, but this does not mean that the buildings in those images are dissimilar. The relevant number of pairs available and used for this training are shown in Table 1.

For training, we implement S-CNN as introduced in the previous section, using the MobileNetV2 (Sandler et al., 2018) architecture with image input size $224 \times 244 \times 3$. Each BOV is cropped with a possible minimum square from image center point to have a uniform aspect ratio and to ignore irrelevant background scenes. Because we are looking for the same building, the RGB color channels are used rather than using a greyscale version of the original image. To monitor the performance during the training, we track both training and validation loss for each epoch. We determine an epoch of 200 empirically to have training termination criterion using Adam optimization algorithm (Kingma & Ba, 2015) with a batch size of 32 and learning rate of 5×10^{-4}. A margin of value 2 was used. A Linux workstation with i9-7920X CPU clocked at 4.3 GH, 64 GB memory, GTX 1080Ti GPU with 11 GB of video memory and 64-bit operating system is used to run NVIDIA CUDA-enabled PyTorch (a Python package) to train the model. The total training time with the given specifications of the workstation takes 40 to 45 minutes on average. Model training results are shown in Figure 5.

9 | PERFORMANCE EVALUATION USING A SIMILARITY SCORE AND SIMILARITY RANK

The performance of the method is quantitatively measured using a task-oriented evaluation metric specifically designed in this study. While we aim to achieve *true-matches* that are highly ranked among the large volume of BOVs, the decision as to whether or not the retrieval is effective does also depend on assuming a reasonable amount of effort on the part of the user. We focus this evaluation on the two most direct and simple criteria available:

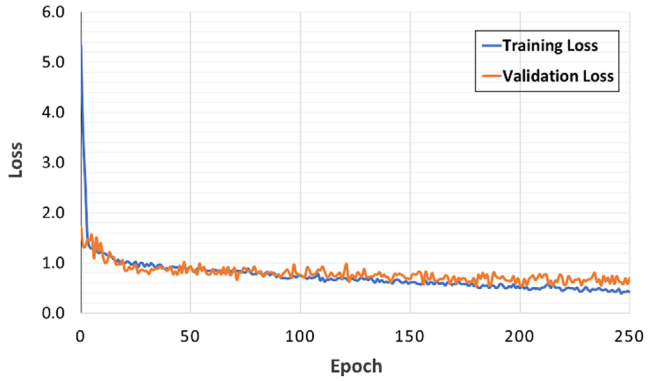

FIGURE 5 The model training result of both training and validation losses

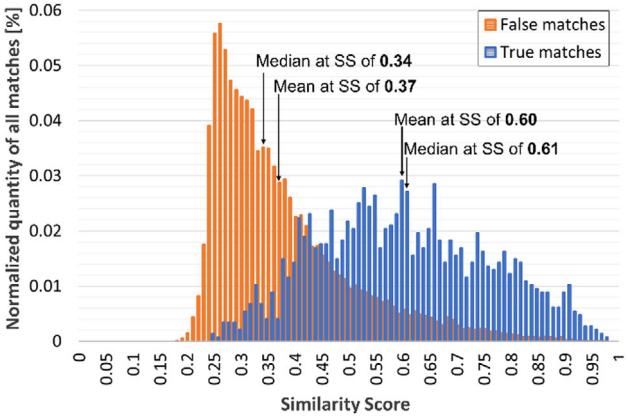

FIGURE 6 Normalized distribution of SS values for all 31,375 possible matches generated from the test dataset

similarity score (SS) and similarity rank (SR) of the *true-matches* among the all BOVs in a retrieval. The detailed results for the performance evaluation are given in this section considering SS, SR, and a binary classification metric (e.g., precision-recall curves) over the continuum of varying both SS and SR.

For this quantitative analysis, we use a test dataset of 251 BOVs from 30 buildings (see Table 1). For each test, each BOV is used as the "query BOV" and the remaining 250 BOVs are regarded as "retrieval BOVs." Then the next BOV is used as the "query BOV" and so on. In the test dataset, the total number of BOVs per building varies considerably, ranging from 2 to upwards of 24 images. Each "retrieval BOV" is individually compared with the query BOV in the machine, and then a SS is assigned. The SS falls in the range from 0 (least similar) to 1 (most similar). Our goal is to predict higher SS values for *true-matches*. This process is repeated for each BOV. Cross-matches between 251 BOVs yield total of 31,375 matches to compare after eliminating self- and commutative-matches (e.g., (251 × 251 − 251)/2). Here this includes 1,477 *true-matches* and 29,898 *false-matches* across the 30 test buildings, as shown in Table 1.

One direct way to evaluate the performance is to look for a different result for the SS value based on our observations of the *true-* and *false-matches*; here *different pairs* are included in *true-matches* as the data for the test is regarded as a raw data. The distribution of SS values for all of the 31,375 possible matches is shown in Figure 6. The number of *true-* and *false-matches* is normalized by dividing with the total number of each case. Note that the values do not fall into distinct clusters, and a few of the *true matches* actually have quite low SS values. The performance of our machine is found to be reasonable in distinguishing *true-matches* from *false-matches*. In the test, the average SS from *true-matches* (mean: 0.60, median: 0.61, and standard deviation: 0.28), blue-colored bars, are much higher

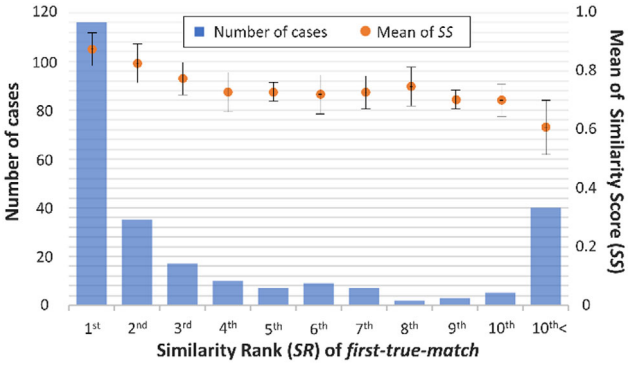

FIGURE 7 SR distribution of *first-true-matches* from all 251 BOV test dataset

than the ones from *false-matches* (mean: 0.37, median 0.34, and standard deviation: 0.127), orange-colored. The values of mean, median, and standard deviation change slightly with multiple trials although it is observed that those values of *true-matches* are always higher than *false-matches*.

Next, we analyze SR (rankings). With 250 retrieval BOVs per query in our test, the SR is assigned to each "retrieval BOV" and has a range from 1 (highest rank) to 250 (lowest rank). We set criteria for success as yielding at least one *true-match* that is ranked within the top 10. For evaluating the method, we focus on the sole *true-match* that is highest ranked among all *true-matches* in the retrieval, named *first-true-match* here. We are not concerned yet with all of the *true-matches*. This is because each building has different number of BOVs and, on the application side, ranking one BOV from *true-matches* (*first-true-match*) in a higher rank is more important than the inclusion of all true-matches within the higher rank. The SR distribution of *first-true-matches* is shown for all 251 tests in Figure 7. The average SS score for the *first-true-match* is also shown in Figure 7. Among the 251 tests, 215 *first-true-matches* have an SR within the top 10. Their average SS value is 0.83.

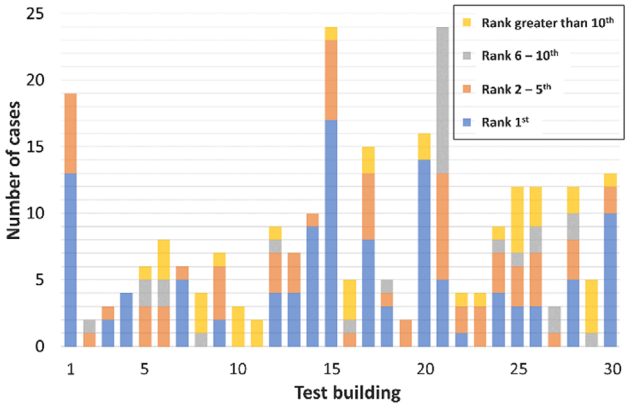

FIGURE 8 SR of *first-true-matches* per building where we identify the corresponding building across the 30 test buildings when each BOV is used as a query; the SR of *first-true-match* within top 10 for the query, bars with blue-, orange-, and gray-colored is dominant for most of the buildings

Surprisingly, 116 *first-true-matches* have an SR value of 1 (with an average SS value of 0.90). These results indicate that a BOV of the same building as the "query BOV" within the top 10 retrieved BOVs with a probability of 85.6% (sum of the number of cases from 1st to 10th divided by 251). This result tells us that users need only look at the top 10 BOVs automatically ranked from our system to successfully.

The performance may also vary for each building. Figure 8 provides the SR values of the *first-true-matches* per building, where we identify the corresponding building across all 30 test buildings. As mentioned earlier, each building has a different number of possible *true-matches*, that is, a different number of matches exists for each building in the entire data set, and thus the bars are different sizes. To generate a single bar in this plot, each BOV is used as a query, and the SR value for all of the *true-matches* is examined. For example, suppose that one BOV from building #1 is used as a "query BOV." If the *first-true-match* retrieved has an SR of 1, then this case is the blue-colored portion of the bar in Figure 8. The same procedure is followed for all 251 tests. The SR values of *first-true-matches* within the top 1, 2–5, and 6–10 for the query are present at the bars with blue-, orange-, and gray-coloring, respectively, and they are dominant for most of the buildings (see the portions of blue, orange and gray compared with the one with yellow). There are two unfavorable cases, noticeable for buildings #10 and #11 for which all results are low ranked, shown as yellow-colored bars. Upon investigation, we observed that these cases have very few BOVs (less than 3) showing different façades of the building that are not visually similar. This makes sense that they are "different pairs" defined in the previous section, which are not included in the training. Figures 9a, b, and c show sample BOVs match results with SS and SR values for similar, different, and dissimilar pairs. The performance on these three different pairs is demonstrated with an identical query BOV in the first row in Figure 9. Note that from our observations of the available datasets, collecting fewer than three BOVs per building is quite an unusual scenario in a real-world reconnaissance mission. In general, reconnaissance teams capture multiple BOVs for each façade of a building. This study concludes that for 28 out of 30 test buildings our trained network produces satisfactory results with at least one *true-match* existing with an SR value in the top 10, with the exception of the two cases mentioned previously.

We found that the number of BOVs for a given building is one of the major parameters affecting the performance of this method. Obviously, the odds for obtaining a *true-match* increases when more BOVs corresponding to the "query BOV" exist among the "retrieval BOVs." Thus, to estimate the minimum number of BOVs that should be collected by a building reconnaissance team, we track the performance while varying the number of corresponding BOVs from 1 to 23. Here, 2 and 24 are the minimum and maximum number of BOVs among test dataset. Figure 10 shows the probability that at least one *true-match* exists. The blue dotted line represents the top 10 SR values using our method and the orange line indicates its theoretical probability when any 10 BOVs are randomly selected BOVs among the total 250 "retrieval BOVs." To calculate this theoretical probability of detecting at least one *true-match* (POD) with a random selection of 10 BOVs, we adopt the hypergeometric distribution, with probability mass function represented in Equation (3) (Rice, 2006):

$$P(X = k) = \binom{K}{k}\binom{N-K}{n-k} \Big/ \binom{N}{n} \tag{4}$$

where N and n stand for the number of retrievals of BOVs available in the entire data set and the number of randomly selected BOVs, respectively, K represents the number of *true-matches* desired in the top n selected BOVs and k is the number of BOVs corresponding to the query building (BOV). While the value of k may take on values between from 1 to 23 for different buildings in our dataset, other parameters are fixed for our testing dataset with $N = 250$ and $n = 10$. In combinatorics, based on Vandermonde's identity, the following relationship in Equation (5) holds:

$$\sum_{k=0}^{n} P(X = k) = 1 \tag{5}$$

FIGURE 9 Sample BOVs match results with SS and SR values from (a) similar, (b) different, and (c) dissimilar pairs

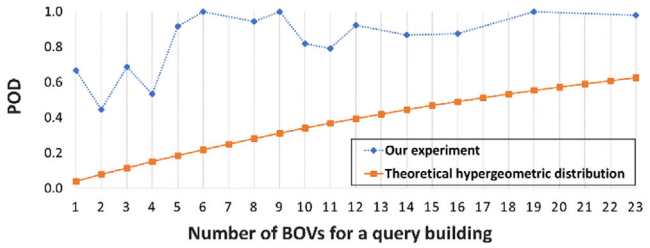

FIGURE 10 The probability that at least one *true-match* exists within either: (1) the top 10 SR values from our experiment; or (2) any 10 BOVs randomly selected using hypergeometric distribution

Thus, POD can be found by subtracting no detection probability ($k = 0$) from 1 in Equation (6):

$$\text{POD} = 1 - P(X = 0) \tag{6}$$

In this comparison, our method is shown to consistently and considerably outperform random selection even in the case where only one corresponding BOV is available. In our test with 30 buildings, the probability of finding a BOV ranked in the first ten outcomes is at least 80% when a total of 251 BOVs are used to retrieve and the corresponding datasets include over five BOVs per building. Thus, a guideline suggested to potential users for the proposed system is that the buildings having more than five BOV images can be likely detected by a single query BOV image when they observe the top 10 ranked images. When the user thinks that the number of corresponding BOVs is less than five in the dataset, the POD value can be increased by increasing the number of n, which is the number of ranked BOVs observed by users.

10 | PERFORMANCE EVALUATION BASED ON PRECISION AND RECALL

Evaluating the model with widely used metrics such as *precision* and *recall* in binary classification will support its performance and capability. However, the model does not directly provide binary classification results; our similarity-based model is to measure visual similarity via SS and SR. For example, when each query is regarded as a single trial of a test, the performance would highly depend on the user's preference and the quantity and quality of the database in which the user can choose values of SS and SR appropriate for defining true or false classification. Here, we define solid criteria in setting ground-truth labeling (true or false) and decision threshold (positive or negative). To develop intuition about these values, we investigate precision and recall of the similarity-based image retrieval depending on a threshold of SS (SS_T) and the top n BOVs considerations (n_T). The metrics for computing precision and recall are defined as follows:

- True Positive (TP): A case that any *true-match* has an SS equal to or higher than SS_T (positive decision) and at least one *true-match* exists within top n_T in the retrieval (true label);
- False Positive (FP): A case that any *true-match* has an SS of equal to or higher than (SS_T) (positive decision) and at any *true-match* does not exist within n_T in the retrieval (false label);
- True Negative (TN): A case that any *true-match* has an SS less than (SS_T) (negative decision) and any *true-match* does not exist within the top n_T in the retrieval (false label);

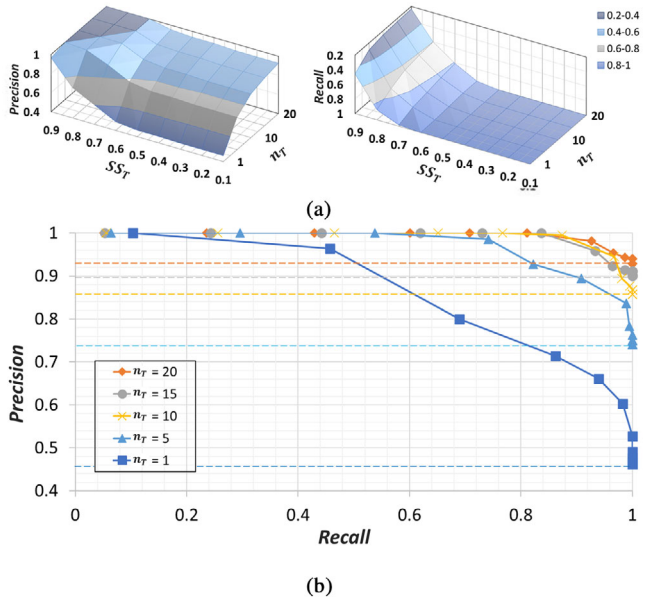

FIGURE 11 Evaluation with a binary classification metric: (a) *precision* and *recall* obtained from setting ground-truth labeling varying SR, and decision threshold varying SS; and (b) *precision–recall* curves obtained by varying nT of 1, 5, 10, 15, and 20

- False Negative (FN): A case that any *true-match* has an SS less than (SS_T) (negative decision) and at least one *true-match* exists within the top n_T in the retrieval (true label).

These values are computed based on 251 trials in the test dataset. Figure 11a demonstrates the precision and recall depending on the SS_T and n_T. For instance, when $SS = 0.8$ and $SR = 10$, the *precision* would be described with the phrase: among a total of 251 trials, the proportion of the cases containing at least one *true-match* within the top 10 among the cases containing any *true-match* having an SS of above 0.8. The corresponding phrase for *recall* is: among total 251 trials, the proportion of the cases containing at least one *true-match* having an SS of above 0.8 among the cases at least one *true-match* within the top 10.

We summarize the results in Figure 11b using the precision–recall curves when n_T is 1, 5, 10, 15 and 20. The result shows that, for higher n_T, more retrieved images are considered as a true classification. It is noticeable that both *precision* and *recall* can be achieved nearly "1" when SR is ramping up. This result makes sense in that as more data are reviewed by users, the chance to detect the BOV corresponding to query building increases. This also demonstrates that observing more BOVs is the way of increasing POD mentioned in Section 3.3. By increasing the numbers of BOVs in the database, the precision and recall curves could be going down in the anti-diagonal direction because the number of false-positive detection is increased in proportion to the size of the database. Thus, this graph shows that users could use this graph to balance the trade-off between accuracy (detecting at least one BOV corresponding to the query building) and usability (the number of ranked BOVs observed).

11 | CONCLUSION

In this paper, we have developed a method to search a building reconnaissance dataset by using a building image as the input to the query. The approach aims to reorganize the large volume of images in terms of visual similarity to the query image, achieving this goal by leveraging and adapting recent advances in deep learning research, specifically Siamese CNNs. Building overview images are automatically extracted from a database, and then are compared to a query image using our method. The technique is demonstrated using data collected from two recent reconnaissance missions, the 2016 Southern Taiwan earthquake (1226 BOVs from 117 buildings) and the 2017 Pohang earthquake, South Korea (379 BOVs from 54 buildings). A quantitative evaluation is conducted via various metrics developed for this application. The results demonstrate that our similarity-based, identical building search approach can be used effectively to search for RC buildings by measuring visual similarity between their overview images, thereby serving as the basis for retrieving relevant images from another dataset.

The evaluation also demonstrates that our trained model distinguishes *true-matches* from *false-matches* successfully, with the SS obtained with *true-matches* (mean: 0.60 and median: 0.61) being much higher than those obtained with *false-matches* (mean: 0.37 and median 0.34). Also, this study shows that our trained network produces satisfactory results for 28 out of 30 test buildings with at least one *true-match* identified with an SR value in the top 10. A key contribution of this work is to develop and validate the building search capability using a post-event database of RC buildings containing real-world images collected from recent natural disasters. We expect that this technique provides an engineer with tools that will reduce efforts, improve consistency, and accelerate decisions after a major disaster. In our future work, we will deploy this capability available online for public users; we will also include useful functions such as retrieving nearby images along with the searched BOV to provide relevant data to support further investigation.

ACKNOWLEDGMENTS

We acknowledge that this work supported by the National Science Foundation OAC Program under grant no OAC-1835473 and the National Research Foundation of Korea

(NRF) grant funded by the Korea government (MSIT) (No. 2021R1G1A1012298). We are grateful to the NVDIA Corporation for the donation of a high-end GPU board and thank Dr. Chungwook Sim for providing large-scale post-earthquake reconnaissance images. Also, we acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

ORCID

Jongseong Choi https://orcid.org/0000-0002-6138-8809
Shirley J. Dyke https://orcid.org/0000-0003-3697-992X

REFERENCES

Adeli, H. (2020). Four decades of computing in civil engineering. In *CIGOS 2019, Innovation for sustainable infrastructure* (pp. 3–11). Springer.

Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing and Applications*, *32*(12), 8675–8690.

American Concrete Institute. (2017). Disaster reconnaissance. *ACI 133*. ACI: Farmington Hills, MI.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, *110*(3), 346–359.

Breiman, L. (1996). Bagging predictor. *Machine Learning*, *24*(2), 123–140.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1994). Signature verification using a "Siamese" time delay neural network. *Advances in Neural Information Processing Systems*, *6*, 737–744.

Chherawala, Y., Roy, P. P., & Cheriet, M. (2013). Feature design for offline Arabic handwriting recognition: Handcrafted vs automated? *12th International Conference on Document Analysis and Recognition, USA*, 290–294.

Choi, J., & Dyke, S. J. (2020). CrowdLIM: Crowdsourcing to enable lifecycle infrastructure management. *Computers in Industry*, *115*, 103185.

Choi, J., Yeum, C. M., Dyke, S. J., & Jahanshahi, M. R. (2018). Computer-aided approach for rapid post-event visual evaluation of a building façade. *Sensors*, *18*(9), 3017.

Chopra, S., Hadsell, R., & LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, *USA*, *1*, 539–546.

Datacenterhub. (2014). https://datacenterhub.org/

DesignSafe. (2016). DesignSafe-CI. https://www.designsafe-ci.org/

Designsafe-CI. (2016). Rapid Experimental Facility. https://rapid.designsafe-ci.org/

EERI. (2009). Earthquake Clearinghouse–Earthquake Engineering Research Institute. https://www.eeri.org/

Gordo, A., Almazán, J., Revaud, J., & Larlus, D. (2016). Deep image retrieval: Learning global representations for image search. *Paper presented at the European Conference on Computer Vision*, 241–257.

Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, *2*, 1735–1742.

Iqbal, Q., & Aggarwall, J. K. (1999). Applying perceptual grouping to content-based image retrieval: Building images. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Cat. No PR00149), *USA*, *1*, 42–48.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations*, San Diego.

Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. *ICML Deep Learning Workshop*, 2.

Kokare, M., Chatterji, B. N., & Biswas, P. K. (2003). Comparison of similarity metrics for texture image retrieval. *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, *2*, 571–575.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*, 1097–1105.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 396–404.

Lenjani, A., Dyke, S. J., Bilionis, I., Yeum, C. M., Kamiya, K., Choi, J., Liu, X., & Chowdhury, A. G. (2020a). Towards fully automated post-event data collection and analysis: Pre-event and post-event information fusion. *Engineering Structures*, *208*, 109884.

Lenjani, A., Yeum, C. M., Dyke, S., & Bilionis, I. (2020b). Automated building image extraction from 360° panoramas for postdisaster evaluation. *Computer-Aided Civil and Infrastructure Engineering*, *35*(3), 241–257.

Leutenegger, S., Chli, M., & Siegwart, R. (2011). BRISK: Binary robust invariant scalable keypoints. *International Conference on Computer Vision*, Barcelona, Spain, 2548–2555, https://doi.org/10.1109/ICCV.2011.6126542.

Li, J., Huang, W., Shao, L., & Allinson, N. (2014). Building recognition in urban environments: A survey of state-of-the-art and future challenges. *Information Sciences*, *277*, 406–420.

Liang, X. (2019). Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization. *Computer-Aided Civil and Infrastructure Engineering*, *34*(5), 415–430.

Liu, X., Dyke, S. J., Yeum, C. M., Bilionis, I., Lenjani, A., & Choi, J. (2020). Automated indoor image localization to support a post-event building assessment. *Sensors*, *20*(6), 1610.

Liu, Y., Chen, X., Cheng, J., & Peng, H. (2017). A medical image fusion method based on convolutional neural networks. *20th International Conference on Information Fusion (Fusion)*, 1–7.

Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, *22*(10), 761–767.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning*. MIT Press.

Mondal, T. G., Jahanshahi, M. R., Wu, R.-T., & Wu, Z. Y. (2020). Deep learning-based multi-class damage detection for autonomous post-disaster reconnaissance. *Structural Control and Health Monitoring*, *27*(4), e2507.

Monical, J. (2020). *Building Surveys after Earthquakes* (1.0) [Data set]. DEEDShub. https://datacenterhub.org/deedsdv/publications/view/137

NCREE. (2016). *Datacenterhub—Resources: Performance of reinforced concrete buildings in the 2016 Taiwan (Meinong) earthquake*. https://datacenterhub.org/resources/14098

PEER. (2013). Pacific Earthquake Engineering Research Center. https://peer.berkeley.edu/

Peker, K. A. (2011). Binary sift: Fast image retrieval using binary quantized sift features. *9th International Workshop on Content-Based Multimedia Indexing (CBMI)*, 217–222.

Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J. P., & Adeli, H. (2020). FEMa: A finite element machine for fast learning. *Neural Computing and Applications*, *32*(10), 6393–6404.

Qi, Y., Song, Y.-Z., Zhang, H., & Liu, J. (2016). Sketch-based image retrieval via Siamese convolutional neural network. *IEEE International Conference on Image Processing (ICIP)*, 2460–2464.

QuakeCoRE. (2016). http://www.quakecore.nz/.

Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 3074–3083.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, *39*(6), 91–99.

Rice, J. A. (2006). *Mathematical statistics and data analysis*. Cengage Learning.

Rosten, E., & Drummond, T. (2005). Fusing points and lines for high performance tracking. *Tenth IEEE International Conference on Computer Vision (ICCV'05), 1*, 1508–1515.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., & Bernstein, M. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4510–4520.

Sim, C., Laughery, L., Chiou, T. C., & Weng, P. (2017). Datacenterhub—esources: 2017 Pohang earthquake—Reinforced concrete building damage survey: About. Pohang earthquake – Reinforced boncrete building damage survey. https://datacenterhub.org/resources/14728#2017%20Pohang%20Earthquake

Sun, Y., Wang, X., & Tang, X. (2014). Deep learning face representation by joint identification-verification. *ArXiv:1406.4773 [Cs]*. http://arxiv.org/abs/1406.4773

UC CEISMIC. (2012). Canterbury Earthquake Digital Archive. http://www.ceismic.org.nz/

Vedaldi, A., & Fulkerson, B. (2010). VLFeat: An open and portable library of computer vision algorithms. *Proceedings of the 18th ACM International Conference on Multimedia, Italy*, 1469–1472.

Vinyals, O., Blundell, C., Lillicrap, T., & Wierstra, D. (2016). Matching networks for one shot learning. *NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems*, 3637–3645.

Wang, J., Cheng, Y., & Feris, R. S. (2016). Walk and learn: Facial attribute representation learning from egocentric video and contextual data. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, USA*, 2295–2304.

Wu, R.-T., Singla, A., Jahanshahi, M. R., Bertino, E., Ko, B. J., & Verma, D. (2019). Pruning deep convolutional neural networks for efficient edge computing in condition assessment of infrastructures. *Computer-Aided Civil and Infrastructure Engineering*, *34*(9), 774–789.

Yeum, C. M., Dyke, S. J., Benes, B., Hacker, T., Ramirez, J., Lund, A., & Pujol, S. (2019). Postevent reconnaissance image documentation using automated classification. *Journal of Performance of Constructed Facilities*, *33*(1), 04018103.

Yeum, C. M., Dyke, S. J., & Ramirez, J. (2018). Visual data classification in post-event building reconnaissance. *Engineering Structures*, *155*, 16–24.

Zhang, C., Liu, W., Ma, H., & Fu, H. (2016). Siamese neural network based gait recognition for human identification. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), China*, 2832–2836.

How to cite this article: Choi J, Park Ju An, Dyke SJ, Yeum CM, Liu X, Lenjani A, Bilionis I. Similarity learning to enable building searches in post-event image data. *Comput Aided Civ Inf*, 2021;1–15. https://doi.org/10.1111/mice.12698