Received: 19 January 2019 | Revised: 19 December 2019 | Accepted: 20 December 2019

DOI: 10.1002/stc.2507

RESEARCH ARTICLE

WILEY

Deep learning-based multi-class damage detection for autonomous post-disaster reconnaissance

Tarutal Ghosh Mondal[1] | **Mohammad R. Jahanshahi**[1,2] | **Rih-Teng Wu**[1] | **Zheng Yi Wu**[3]

[1]Lyles School of Civil Engineering, Purdue University, West Lafayette, Indiana

[2]School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana

[3]Bentley Systems, Incorporated, Watertown, Connecticut

Correspondence
Mohammad R. Jahanshahi, Lyles School of Civil Engineering, Purdue University, West Lafayette, IN.
Email: jahansha@purdue.edu

Abstract

Timely assessment of damages induced to buildings due to an earthquake is critical for ensuring life safety, mitigating financial losses, and expediting the rehabilitation process as well as enhancing the structural resilience where resilience is measured by an infrastructure's capacity to restore full functionality post extreme events. Since manual inspection is expensive, time consuming and risky, low-cost unmanned aerial vehicles or robots can be leveraged as a viable alternative for quick reconnaissance. Visual data captured by the sensors mounted on the robots can be analyzed, and the damages can be detected and classified autonomously. The present study proposes the use of deep learning-based approaches to this end. Region-based convolutional neural network (Faster RCNN) is exploited to detect four different damage types, namely, surface crack, spalling (which includes façade spalling and concrete spalling), and severe damage with exposed rebars and severely buckled rebars. The performance of the proposed approach is evaluated on manually annotated image data collected from reinforced concrete buildings damaged under several past earthquakes such as Nepal (2015), Taiwan (2016), Ecuador (2016), Erzincan (1992), Duzce (1999), Bingol (2003), Peru (2007), Wenchuan (2008), and Haiti (2010). Several experiments are presented in the paper to illustrate the capabilities, as well as the limitations, of the proposed approach for earthquake reconnaissance. It was observed that Inception-ResNet-v2 significantly outperforms the other networks considered in this study. The research outcome is a stepping stone forward to facilitate the autonomous assessment of buildings where this can be potentially useful for insurance companies, government agencies, and property owners.

KEYWORDS

damage detection, deep learning, Faster RCNN, earthquake reconnaissance, UAV

1 | INTRODUCTION

1.1 | Background

Buildings form an important part of urban infrastructure systems. Damage in buildings caused by earthquake events not only renders the residents homeless but also brings to a halt various economic activities, which are directly or indirectly

dependent on building infrastructures. It also disrupts essential service utilities, which act as lifelines for the people of the locality. Therefore, it is of vital importance to ensure that the full functionality of buildings is quickly restored in the wake of an earthquake event. The ability of a building to withstand damage and recover in a timely manner following an extreme event is called structural resilience, which is recognized by the scientific community as a promising research area owing to its profound impact on life safety and overall economy. However, an expeditious disaster recovery calls for a rapid and comprehensive evaluation of the nature and extent of damages inflicted by the extreme event on building infrastructure systems. The existing earthquake reconnaissance practices are predominantly manual. A group of certified inspectors visits the affected buildings, taking measurements from the damaged areas, post processing the collected information, and finally arriving at the retrofit decision. Needless to say, this procedure is time consuming and expensive as it requires a lot of manpower. Sometimes, it also involves risk as the human inspectors need to visit or go very close to a damaged structure, which is about to collapse, to record an accurate reading. As a viable alternative, such manual methods can be replaced by inexpensive UAVs or inspection robots, which can cruise autonomously through potentially damaged buildings looking for damages and collecting critical information using on-board vision-based and other types of sensor systems, which will help identify the problem areas requiring immediate attention. An autonomous engineering assessment of this sort will help identify the risks and mitigate life-safety hazards for human inspectors by preventing them from entering a building, which is prone to collapse. It will also enable quick evaluation of recovery and repair cost and financial loss induced by downtime. Additionally, reserve capacity for different structural elements can be assessed from corresponding damage levels, and retrofit operations can be planned accordingly.

Several studies in the past focused on improving the resilience of infrastructure systems by proposing improved techniques for rapid structural inspection and damage assessment capitalizing on the recent advancements made in fields of computer vision and deep learning. Image processing-based techniques are exploited by many researchers to this end. Yamaguchi and Hashimoto[1] proposed a percolation-based image processing technique for fast and efficient concrete crack detection. German et al.[2] exploited entropy-based thresholding in conjunction with template matching and morphological operations for rapid detection and quantification of concrete spalling during post earthquake safety assessments. Similar studies related to damage detection in other forms of structural systems include identification of cracks in bridges,[3] pavement surfaces,[4-10] and underground pipes and subway tunnels.[11,12] Several researchers[13-16] in the past also focused on machine learning-based techniques for automatic vision-based damage detection where the feature vectors are selected manually in these methods.

On the other hand, rapid enhancement in computational capacity in recent times has triggered a resurgence of deep learning-based convolutional neural networks (CNNs) garnering significant attention from the research community cutting across all disciplines. On this account, a number of studies in the past[17-19] explored the possibility of applying CNN for efficient and autonomous detection of various structural damages. However, very few studies indeed looked into damage types, which are relevant to earthquake reconnaissance of reinforced concrete (RC) building systems, which is the focus of the present study. Cha et al.[20] investigated multiple damage categories such as concrete cracks, steel corrosion, bolt corrosion, and steel delamination with the help of region-based CNN (RCNN). However, the damage categories considered by the authors were not closely correlated and are not applicable to RC buildings on the whole. Cha et al.[21] exploited sliding window approach for detecting cracks on concrete surfaces. However, it ignored other types of damages that are commonly observed in RC buildings post earthquake events. Yeum et al.[22] recently proposed an RCNN-based approach for spalling recognition in RC buildings. This study also does not take into account other damage types that may possibly result when such buildings are subjected to seismic vibrations. Kim et al.[23] capitalized on image binarization and CNN to distinguish crack from crack-like noncrack noise patterns (e.g., dark stains, shades, dust, lumps, holes, etc.) on concrete surfaces. Therefore, this study also had limited scope in terms of varieties of damage types considered. Chen and Jahanshahi[24] proposed a CNN-based approach to detect cracks on nuclear reactors. The false detections were discarded using Naïve Bayes data fusion by aggregating information from successive frames in inspection videos. However, this study was also exclusively focused on identification of cracks, and other types of damages were ignored. Hoskere et al.[25] harnessed pixel-wise classification of images using deep CNN to identify multiple damage classes in civil infrastructure systems. However, only two (concrete crack and concrete spalling) out of six damage categories considered in this study were relevant to RC buildings, and the rest corresponded to the deterioration in steel structures and asphalt pavements. For instance, some important damage types such as buckling of column rebars caused by severe earthquake vibrations were not considered in this study. Additionally, it involved expensive training data preparation process like pixel-wise labeling of images. It should be noted that ignoring severe damage categories such as exposed and buckled rebars may have adverse safety ramifications, as it may lead to underestimation of the damage severity and falsely encourage the

human inspectors to enter a building, which is on the verge of collapsing, resulting in fatal injuries. This underlines the necessity of including multiple damage categories representing the entire spectrum of severity in the autonomous damage detection pipeline. Recently, Gao and Mosalam[26] presented a CNN-based approach for structural damage classification. Although, this study considered a range of damage categories and various classification modalities, namely, component type identification, spalling condition check, damage level estimation, and damage type determination, it did not focus on localizing the damage in the images. This is an important limitation, which was recommended as a part of the further works by the authors.

1.2 | Scope and contribution

It is important that the application of deep CNN is extended to multiple damage categories, which will immensely benefit earthquake reconnaissance and safety evaluations. The present study aims at filling this gap by proposing a Faster RCNN-based detection technique taking into account multiple damage types that may be caused in RC buildings when subjected to earthquake ground motion. Four different damage categories are considered in this study, which are surface crack, spalling (which includes façade spalling and concrete spalling), severe damage (i.e., spalling with exposed rebars), and severely buckled exposed rebars. The CNN architectures that were exploited to this end are Inception v2,[27] ResNet-50,[28] ResNet-101,[28] and Inception-ResNet-v2.[29] The efficiency of the proposed algorithms is evaluated with the help of earthquake reconnaissance data collected after several past earthquakes such as Nepal (2015), Taiwan (2016), Ecuador (2016), Erzincan (1992), Duzce (1999), Bingol (2003), Peru (2007), Wenchuan (2008), and Haiti (2010). Inception-ResNet-v2 is observed to significantly outperform other architectures considered in this study. The authors believe that this study will help enhance autonomous post disaster reconnaissance of RC buildings.

The datasets used in this study for training and evaluation of detection algorithms were collected from different countries representing wide variation in local construction practices and design specification. Therefore, the images contained damages in various shapes, sizes, and aspect ratios. This poses a challenge of dealing with this scale variation and devising a detection algorithm which is scale agnostic. This research challenge was addressed in this study by modifying the region proposal network (RPN).[30] Typically, a 3×3 sliding window is applied to the feature map generated by the last convolutional layer in the RPN. At each sliding window location, a number of anchor boxes having different scales and aspect ratios are considered as region proposals to account for scale variability of objects. In the default configuration of Faster RCNN proposed by Ren et al.,[30] a total of nine anchor boxes were proposed with three different scales and three different aspect ratios. However, in this study, seven different scales and eight different aspect ratios were used leading up to 56 anchor boxes, which improved detection accuracy significantly.

Research on robot-based autonomous inspection and condition assessment has many facets. A number of researchers in the past focused on developing advanced robotic systems and path planning algorithms with an eye to futuristic inspection operations. Simultaneously, recent advances in the fields of computer vision and deep learning evoked profound interests in vision-based damage diagnosis, which is investigated in this study. The scope of this work is limited to visual data analysis for autonomous multi-class seismic damage identification. Hands-on experiment with real physical robot is not considered here and is a part of future work. The manuscript has been arranged in the following order. Section 2 discusses the Faster RCNN approach for object detection. Various CNN architectures considered in this study are briefly described in Section 3. Image dataset used for the evaluation of the proposed approach is presented in Section 4. The training scheme and other implementation details are summarized in Section 5. The detection results are presented and discussed in Section 6. Finally, conclusions are summarized in Section 7.

2 | FASTER RCNN

The inception of Faster RCNN can be traced back to introduction of Regions with CNN features (RCNN) by Girshick et al.[31] In RCNN, around 2,000 category-independent region proposals are extracted from the input image using selective search algorithm. Each region proposal is then sent to a CNN to generate a fixed-length feature vector. Finally, category-specific linear Support Vector Machine (SVMs) are used to classify each regFaster RCNN is designed to work with variableion proposal. At the end, greedy non-maximum suppression is employed to get rid of the redundant detections. However, RCNN is slow during training and testing because features are extracted from each region proposal in each image and written to the disk. Girshick[32] addressed this shortcoming by replacing the multi-stage training pipeline of RCNN with a single-stage algorithm. In this refined approach called Fast RCNN, an input image together with a set of object proposals is input to a series of convolutional and max pooling layers to obtain a feature map. Then, a region of

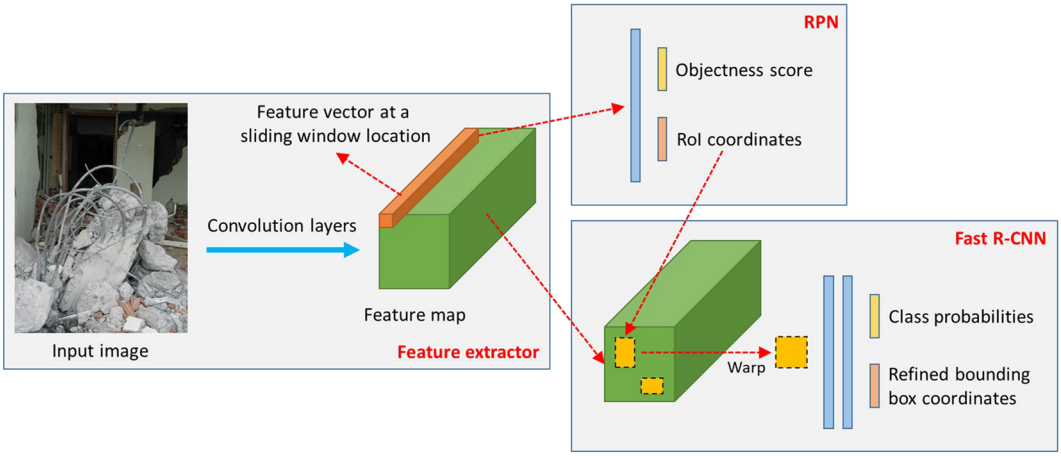

FIGURE 1 Faster RCNN architecture

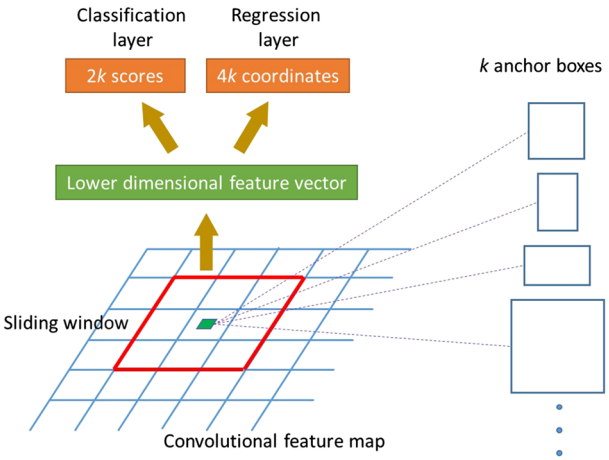

FIGURE 2 Region proposal network

interest (RoI) pooling layer is invoked to extract a fixed-length feature vector from the feature map for each of the object proposals. Each of the feature vectors is then fed into a sequence of fully connected layers, which eventually bifurcate into two collateral output layers constituting a softmax classifier and a bounding box regressor. Fast RCNN is significantly faster than RCNN during training and testing, owing to computation and memory sharing across the RoIs from the same image. It also offered higher accuracy compared with RCNN. However, the region proposal computation was a bottleneck; elimination of which was likely to further speed up the testing process. This was materialized by Ren et al.,[30] who proposed an RPN drastically reducing the cost of region proposal computation at the test time.

2.1 | Region proposal network

RPN is a fully convolutional network trained to predict object bounds along with objectness scores. The region proposals generated by RPN are used by Fast RCNN for accurate detection. The RPN and the Fast RCNN modules are unified into a single network enabling sharing of convolutional layers (Figure 1). An $n \times n$ sliding window is applied to the feature map generated by the last shared convolutional layer mapping it down to a lower dimension (Figure 2). At each sliding window location, a set of k anchor boxes having different scales and aspect ratios are considered as region proposals. The lower dimensional feature is fed into two collateral fully connected layers, namely, a box-regression layer and a box-classification layer. The box-regression layer has $4k$ outputs denoting the coordinates of k bounding boxes. On the other hand, the box-classification layer produces $2k$ outputs representing the objectness score of each bounding box. For more details about the training scheme and other implementation details, the readers may refer to the original paper by Ren et al.[30]

On the whole, a CNN is used at first to generate a feature map from the input image. In this study, four different network architectures are exploited to this end, which are described in the Section 3. Subsequently, RPN is used to generate regions proposals; following which, Fast RCNN module is utilized for classifying the RoIs and refining the bounding box

coordinates. In a way, RPN incorporates "attention" mechanism telling the classifier where to look. More details about the implementation scheme are provided in Section 5.

3 | NETWORK ARCHITECTURES

3.1 | Inception v2

Prior to the introduction of inception network,[33] it was a common trend to stack additional convolutional layers to increase the accuracy of the network, leading to a very deep network. Such a deep network is fraught with numerous limitations such as overfitting, vanishing gradients, and expensive computational cost. On the other hand, wide variation in the object size makes it challenging to estimate the most optimum kernel size for convolution operations. Large kernel size is preferred when salient information are globally distributed in the image. On the other hand, locally distributed information call for smaller kernels. These limitations pertaining to prevailing CNN architectures led to the development of a series of inception networks. Szegedy et al.[33] proposed Inception v1 by stacking filters of various sizes on the same level, making the network wider rather than deeper. The outputs from different filters are concatenated and sent to the next layer. Additionally, 1×1 convolutions are introduced aiming at reducing the dimension of input channels and thereby reducing the computational cost. Szegedy et al.[27] proposed Inception v2 by implementing a set of iterative improvements over Inception v1. The authors factorized a 5×5 convolution layer to two 3×3 convolution layers, which reduced the computational cost significantly. The cost was further reduced by replacing a $n \times n$ convolution with a $1 \times n$ convolution followed by an $n \times 1$ convolution. Additionally, the filter banks were expanded to curtail the representational bottleneck (loss of information due to excessive reduction in dimension). More details about the Inception v2 network can be found in Szegedy et al.[27]

3.2 | ResNet-50 and ResNet-101

Very deep neural networks are cursed with the problem of vanishing gradients. As a result, the performance of the network saturates and eventually starts degrading with an increase in depth. He et al.[28] introduced the idea of "skip connection," which enables the activation of one layer to be fed directly to another layer much deeper in the network bypassing one or many intermediate layers. This eventually led to the development of residual block, which facilitates training of very deep neural networks without any appreciable loss of performance. The objective in residual network is to ensure that a deep network does not produce higher training error than its shallower counterpart. Skip connection introduces an identity mapping, which is easier for the residual block to learn pushing the residual function to zero. This ensures that the addition of extra layers does not adversely impact the accuracy of the network and the deep network performs at least at par with its shallower counterpart. On top of that, if the added layers manage to learn something useful, then the deep residual network can even outperform its nonresidual version. Exploiting this notion of residual block, a series of deep networks are developed. ResNet-50 and ResNet-101, having 50 and 101 convolutional layers, respectively, are investigated in this study. The details of the ResNet architectures can be found in He, Zhang, Ren, and Sun.[28]

3.3 | Inception-ResNet-v2

Inception-ResNet-v2 network[29] incorporates residual connections proposed by He et al.[28] in combination with the latest developments in the inception architecture.[27] Residual connections add the convolution outputs of the inception module to the input. Residual block requires that input to and output from the convolution module have the same dimension; 1×1 convolutions are used to compensate for the dimensionality reduction induced by the inception blocks. The pooling operations inside the original inception modules were replaced in favor of the residual connections, and the same was retained in the reduction blocks. The residual activations were scaled by a factor ranging from 0.1 to 0.3 to get rid of vanishing gradients. A detailed discussion on Inception-ResNet-v2 is beyond the scope of the present study and can be found elsewhere.[29]

4 | DATASETS AND EXPERIMENTAL PROGRAM

Images of buildings damaged by earthquakes experienced in the recent past in different parts of the world (Nepal, Taiwan, Ecuador, Erzincan, Duzce, Bingol, Peru, Wenchuan, and Haiti) are used in this study for the evaluation of the proposed approach. Data from the Taiwan earthquake (2016) were collected by the authors of this study, whereas data from the

FIGURE 3 Illustrative examples of images depicting wide variation in lighting condition and data quality

Nepal (2015), Ecuador (2016), Erzincan (1992), Duzce (1999), Bingol (2003), Peru (2007), Wenchuan (2008), and Haiti (2010) earthquakes were downloaded from Datacenterhub.org of Purdue University, United States.[34-37] Diversity in training data is necessary for reducing model variance, which is a measure of sensitivity of the model to specific observations. A learning algorithm with high variance performs well on training data. However, the performance declines when the model encounters data, which are not used for training. The problem of high variance can be alleviated by introducing variations in the training data. Learning models trained with data collected from various sources are supposed to be more robust when tested on previously unseen data. The images used in this study for training of the detection algorithms represent wide variations in image resolution, lighting condition, blurring, and degree of distortions. Specifically, the database comprises images with 69 different resolutions. The sample images presented in Figure 3 are illustrative of the wide-ranging lighting conditions encountered in the dataset. The database is enriched with diversity, which resulted in better generalization capability of the learning models. All the damages observed under the said earthquakes were subdivided into four categories. The first damage category (Damage 1) denoted surface cracks. Spalling, which includes façade spalling and surface spalling, constituted the second damage category (Damage 2). The third damage category (Damage 3) was composed of spalling with exposed rebars. Severely buckled rebars formed the fourth damage category (Damage 4). Some example images representing the four damage categories are shown in Figure 4. The images were manually annotated and were divided into training set and validation set. Fourfold cross-validation was conducted to examine how well the detection models generalize to independent datasets. The distribution of training and validation data at each round of cross-validation is shown in Table 1. Of all available data, 10% was used for validation at each cross-validation round, and the remaining 90% was used for training. No sample was used twice for validation. The evaluation metrics obtained from all four rounds of cross-validation were averaged to produce a single estimation. It is evident from Table 1 that the training data have uneven representation from different classes, which can potentially make the predictor biased towards the over-represented classes. To mitigate this problem of class imbalance, class specific weights are assigned to the loss function so as to impose additional penalty for misclassifying an under-represented class.[30]

5 | IMPLEMENTATION DETAILS

The Faster RCNN algorithm was implemented using TensorFlow open-source software library[38] and was run on two NVIDIA Titan X (Pascal) Graphics Processing Unit (GPUs). Faster RCNN is designed to work with variable image size and aspect ratio. However, previous studies indicated that resizing the images enhances the performance. Therefore, the input images were resized in this study to a minimum dimension of 600 pixels and maximum dimension of 1,024 pixels keeping the aspect ratio intact. In other words, if the longer dimension of the input image is less than 1,024 pixels, then the shorter dimension is resized to 600 pixels, and the longer dimension is modified proportionally keeping the aspect ratio the same. On the other hand, if the longer dimension of the input image is greater than 1,024 pixels, then the longer dimension is resized to 1,024 pixels, and the shorter dimension is resized appropriately to keep the aspect ratio unchanged. The input images were horizontally flipped randomly with a probability of .5 as part of the data augmentation. Then features are extracted from the input image using a sequence of convolutional layers, which were a part of CNN architectures considered in this study. As for Inception-ResNet-v2, a set of atrous filters are slid over this feature map to carry out atrous convolution.[39] This enables object encoding at multiple scales by extending the receptive field without increasing the

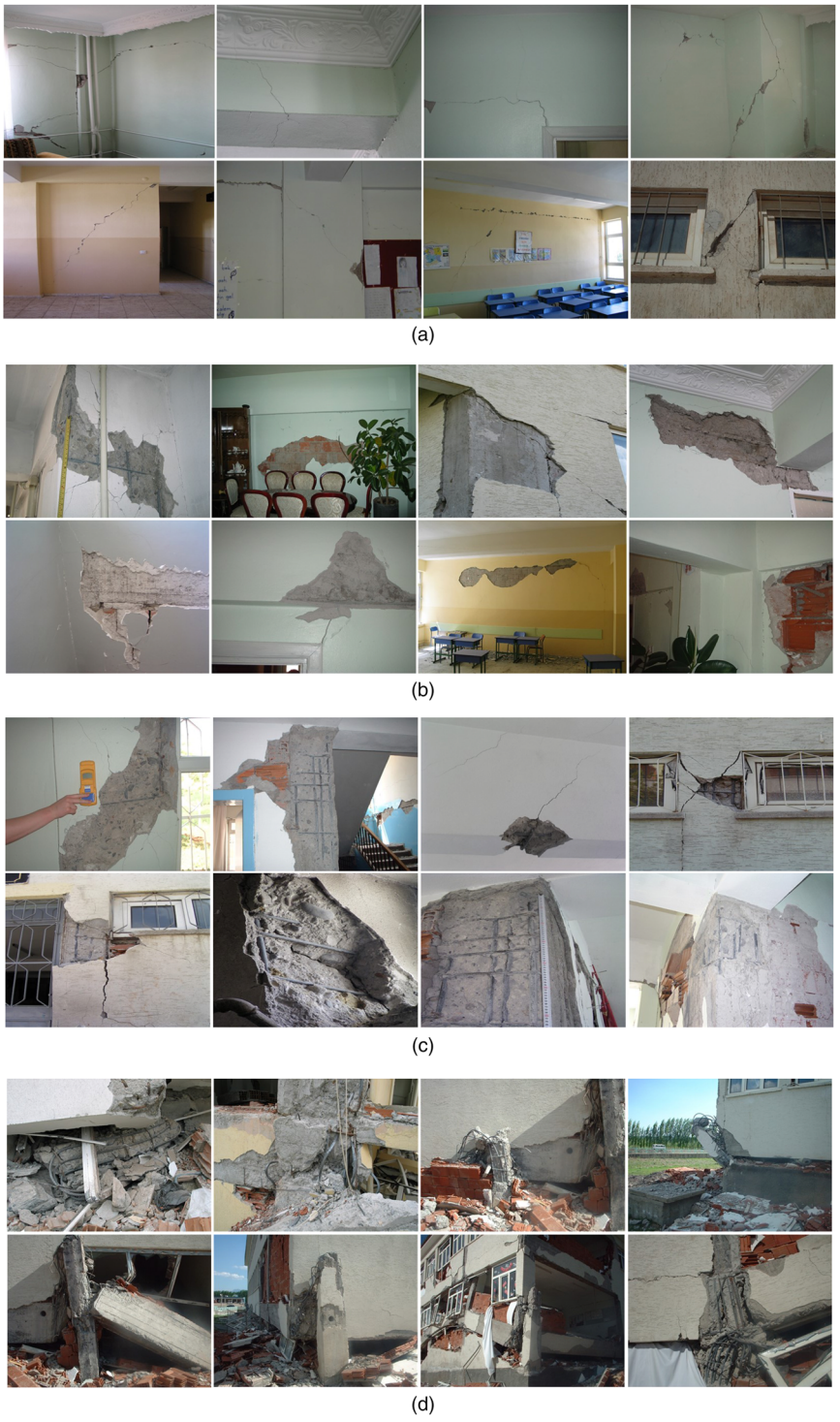

FIGURE 4 Damage categories considered for detection - (a) Damage 1, surface crack; (b) Damage 2, spalling; (c) Damage 3, spalling with exposed rebars; (d) Damage 4, severely buckled rebars

number of parameters and number of operations. In order to generate region proposals using RPN, Ren et al.[30] proposed nine anchor boxes ($k = 9$) with three different scales and three different aspect ratios. However, a wider range of scales and aspect ratios leading to a higher value of k was found to enhance the detection accuracy significantly in this study. Various scales used in this study for anchor box generation include 0.125, 0.25, 0.5, 1.0, 2.0, 4.0, and 8.0. The aspect ratios had the values of 0.125, 0.5, 1.0, 2.0, 4.0, 6.0, 8.0, and 10.0. The minimum of input height and width was considered as base anchor size. The anchor boxes were strided by 8 pixels both along the height and the width. Three hundred region proposals were generated per image calling for elimination of multiple detections. To filter all the duplicate boxes, a greedy procedure called non-maximum suppression[40] was employed, where all candidate boxes are first sorted in the order of their objectness score. The best scoring box was selected, and all other boxes having an intersection-over-union (IoU)

TABLE 1 Category-wise sample size used for training and validation

Cross-validation round	Damage 1		Damage 2		Damage 3		Damage 4	
	Training	Validation	Training	Validation	Training	Validation	Training	Validation
1	865	97	1,751	272	554	148	473	126
2	868	94	1,791	232	547	155	437	162
3	773	189	1,685	338	622	80	563	36
4	863	99	1,811	212	479	223	490	109

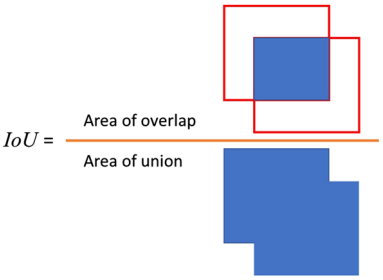

FIGURE 5 Intersection over Union (*IoU*): It is the ratio of the area of intersection to the area of overlap between two boxes

greater than 0.7 with the selected box were discarded. *IoU* is an evaluation metric, which is defined as the ratio of area of overlap to the area of the union between a ground-truth box and a predicted box (Figure 5). The remaining boxes were then classified and refined using a Fast RCNN module. The *IoU* threshold used for non-maximum suppression at this stage was 0.6. The weights of the backbone networks were initialized by a model pretrained on MSCOCO dataset and fine-tuned thereon. MSCOCO[41] is a large repository of images (328*k*) containing 90 different objects that are commonly encountered in everyday life. The said model had 90 neurons in the last layer representing 90 classes. Therefore, this layer was replaced by one with only four neurons in this study. The weights for last layer was initialized from a uniform distribution as suggested by Glorot and Bengio.[42] All the weights are subsequently updated using stochastic gradient descent[43] with a momentum value of 0.9. The problem of exploding gradient is commonly encountered while training a very large neural network. The gradients shoot off exponentially during successive back propagation through the network layers, rendering the learning process highly unstable. This problem can be averted by clipping the gradients by a preset threshold. A threshold of 10 was set for the gradient norm to this end. The initial learning rate was set to 0.003 and was gradually reduced thereafter with training steps.

6 | RESULTS AND DISCUSSIONS

The performance of the proposed approach was evaluated on the validation data described in Section 4. The test images were input to the trained network and bounding boxes were predicted with respective object class as shown in Figure 6. The predicted boxes were compared with the ground truth boxes, and any prediction having an *IoU* greater than a threshold of 0.5 was considered to be *true positive*. If multiple boxes are predicted for a single ground truth box, then only the highest scoring box is considered to be *true positive,* and rest all are dubbed as *false positive*. If a ground truth box does not possess any predicted box associated with it, it is designated as *false negative*. The detection performance of the proposed algorithm with four different CNN architectures are measured in terms of precision and recall values. Precision is defined as the ratio of *true positive* to the sum total of *true positive* and *false positive*. In other words, it tells us what percentage of the overall detections are correct detections. The mean and standard deviation of precision values obtained from four rounds of cross-validation are reported in Table 2. For instance, the precision values obtained from Inception-ResNet-v2 architecture for four classes considered in this study had the mean values of 65.5, 50.0, 52.0, and 53.8%, respectively, whereas the corresponding standard deviation values were estimated as 7.9, 9.5, 5.6, and 5.5%, respectively. It means that 65.5% of all predicted boxes classified as Damage 1 belongs to the correct detections in average sense. Similar interpretations can likewise be extended to other damage classes and CNN architectures. Another evaluation metric, which is often used alongside precision score, is recall. It is the ratio of *true positive* to the sum total of *true positive* and *false negative*. It indicates what percentage of the actual ground truth objects have been successfully identified by the detection network. The mean recall values that the trained network produced for four classes with Inception-ResNet-v2 architecture were 78.8, 65.3, 62.5, and 59.5%, respectively (Table 3). Corresponding standard deviations were evaluated as 8.2, 16.0, 4.2, and 7.8%, respectively. In other words, 78.8% of all damages annotated as Damage 1 was correctly predicted on the average,

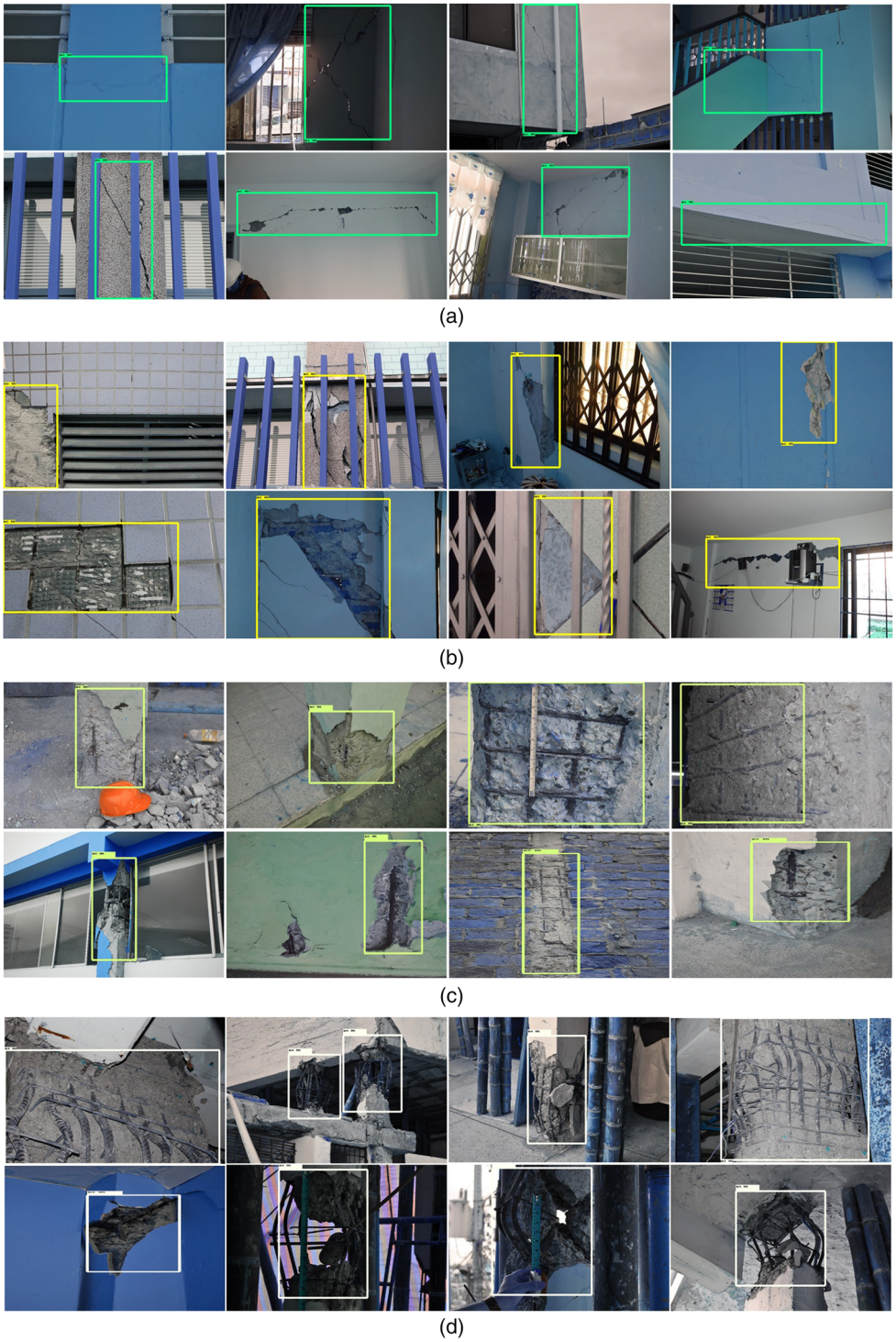

FIGURE 6 Sample detection results—(a) Damage 1, surface crack; (b) Damage 2, spalling; (c) Damage 3, spalling with exposed rebars; (d) Damage 4, severely buckled rebars

and likewise for other classes and CNN architectures. The said precision and recall values are considerably higher in comparison with that reported by Yeum et al.[22] for single class (spalling) detection (Precision: 40.48%, Recall: 62.16 %) on similar dataset. Minor cracks in concrete are typically hard to detect due to potential noise infusion.[24] However, the earthquake-induced cracks used in this study were, by and large, distinct and easily detectable. On the other hand, the other three damage categories were all related to spalling and therefore contained significant visual correlation, which

TABLE 2 Mean (μ) and standard deviation (σ) of precision for different CNN architectures

Architecture	Damage 1		Damage 2		Damage 3		Damage 4	
	μ	σ	μ	σ	μ	σ	μ	σ
Inception v2	0.576	0.110	0.432	0.079	0.410	0.061	0.497	0.092
ResNet-50	0.532	0.121	0.405	0.091	0.448	0.068	0.403	0.111
ResNet-101	0.613	0.089	0.435	0.096	0.423	0.045	0.458	0.103
Inception ResNet v2	0.655	0.079	0.500	0.095	0.520	0.056	0.538	0.055

TABLE 3 Mean (μ) and standard deviation (σ) of recall for different CNN architectures

Architecture	Damage 1		Damage 2		Damage 3		Damage 4	
	μ	σ	μ	σ	μ	σ	μ	σ
Inception v2	0.750	0.112	0.628	0.118	0.553	0.043	0.507	0.116
ResNet-50	0.750	0.153	0.598	0.125	0.572	0.068	0.545	0.087
ResNet-101	0.770	0.108	0.638	0.131	0.565	0.040	0.578	0.098
Inception ResNet v2	0.788	0.082	0.653	0.160	0.625	0.042	0.595	0.078

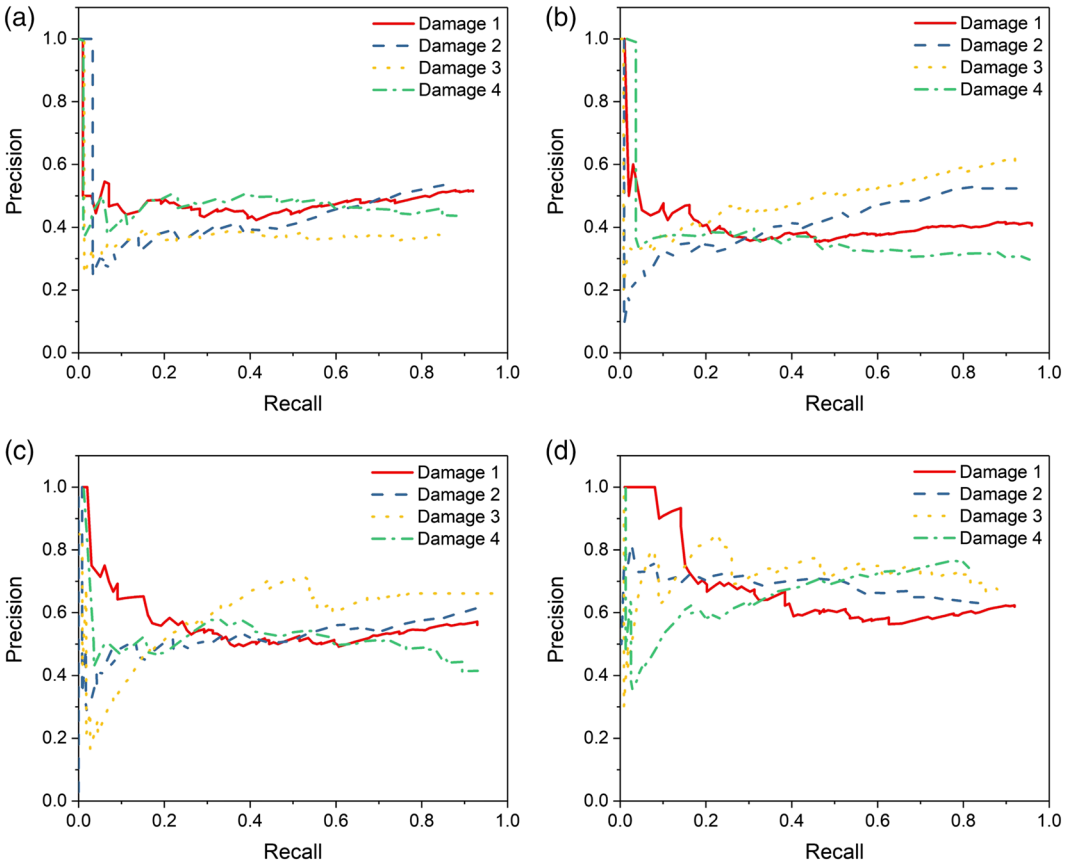

FIGURE 7 Precision - recall curves - (a) Inception v2, (b) ResNet-50, (c) ResNet-101, and (d) Inception ResNet v2

made them less distinguishable from each other, resulting in a lot of classification error. This potentially led to relatively high detection performance vis-à-vis Damage 1 as compared with the rest of the damage categories.

Precision and recall values are inversely related. Setting the detection threshold to a low value will allow the network to predict most of the objects in the image. However, it will generate a large number of false positives at the same time. On the other hand, a high value of the detection threshold will produce very few false positive. However, it will result in numerous missed detections. It is therefore customary not to rely entirely on either of the two decision metrics for the sake of comparison among different detection models. Alternatively, the entire precision–recall curve (Figure 7) is looked into and the area under the curve is used as an evaluation metric. This parameter, also known as the AP, sums up the precision–recall curve to a single number. Higher values of AP indicate better performance of the detector. The AP values are calculated from the precision–recall curves for all four damage types and all four CNN architectures considered in this study, and the mean and standard deviation values (averaged over four rounds of cross-validation) are reported in Table 4.

Figure 8 presents more detailed information with regard to the dispersal of all the evaluation metrics (precision, recall, and AP) for different CNN architectures and damage categories. The variation range formed by one standard deviation on either sides of the mean value is represented by a rectangular box in this figure.

A careful analysis of the information presented in Table 4 indicates that no consistent pattern exists in the performance hierarchy of the four CNN architectures evaluated with respect to the mean AP value. For instance, ResNet-50 produced a higher mean AP for Damages 1, 2, and 3 in comparison with the Inception v2 architecture. However, the trend was reversed for Damage 4 where Inception v2 outperformed the ResNet-50 architecture in terms of the same evaluation metric. Similarly, ResNet-101 performs better than ResNet-50 in terms of mean AP in identifying Damages 1 and 4. However, when it comes to the detection of Damages 2 and 3, ResNet-50 turns out to be more efficient. This anomaly can be resolved by averaging the AP values over all object classes. The evaluation metric thus generated is called mean average precision (MAP), which is typically used to compare the efficiency of different detection algorithms. The mean MAP values (averaged over four rounds of cross-validation) for all four CNN architectures are shown in Figure 9. It is evident from the figure that Inception v2 architecture afforded the lowest accuracy with a mean MAP value of 51.0%. A 3.0% increase in the mean MAP value was observed when ResNet-50 was employed. Invoking ResNet-101 rendered a further increase of 1.5% to the same. However, the best performance was observed with Inception-ResNet-v2 architecture, which produced a mean MAP value of 60.8%.

TABLE 4 Mean (μ) and standard deviation (σ) of AP for different CNN architectures

Architecture	Damage 1		Damage 2		Damage 3		Damage 4	
	μ	σ	μ	σ	μ	σ	μ	σ
Inception v2	0.566	0.096	0.455	0.028	0.490	0.090	0.528	0.023
ResNet-50	0.577	0.102	0.505	0.036	0.568	0.057	0.508	0.113
ResNet-101	0.658	0.107	0.494	0.056	0.529	0.094	0.538	0.131
Inception ResNet v2	0.681	0.080	0.554	0.068	0.627	0.089	0.570	0.088

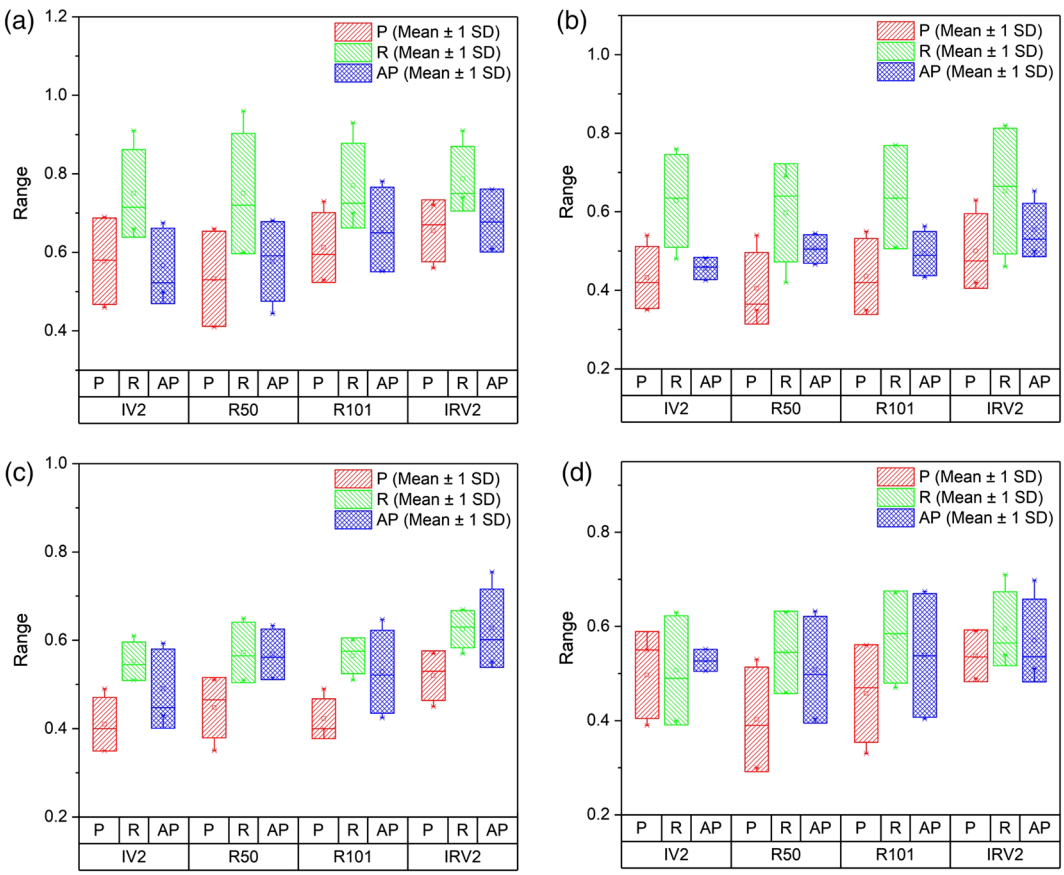

FIGURE 8 Variation of evaluation metrics over all rounds of cross-validation for (a) Damage 1, (b) Damage 2, (c) Damage 3, and (d) Damage 4. IV2 – Inception v2; R50 – ResNet-50; R101 – ResNet-101; IRV2 –, Inception ResNet v2; P – precision; R – recall; AP – average precision; *SD* – standard deviation

Apart from accuracy, another parameter, which is often taken into account while comparing various detection algorithms is the computational cost. The computational cost is measured in this study in terms of average processing time for a single image. It was observed that the architectures that exhibited higher MAP values actually had slower processing speed (Figure 10). This led to the conclusion that detection accuracy and processing speed are inversely related and the selection of a suitable detector is a trade-off between the two. However, the values presented in Figure 10 are highly subjective and are dependent on the image resolution and specific GPU architecture used and, therefore, should not be taken in the absolute sense. However, in relative terms, it can be inferred that Inception v2 is the fastest of all architectures considered in this study. ResNet-50 and ResNet-101 take about 1.2 and 1.7 times the time taken by an Inception v2 architecture to process an image of the same resolution. However, Inception ResNet v2 was identified as the slowest of all considered architectures taking about 5.6 times the time taken by Inception v2 to accomplish the same task.

The ultimate objective of developing damage detection algorithm is to integrate it with robotic systems for autonomous inspection. A major challenge that is encountered to this end is wide-ranging camera specifications leading to huge variations in image resolution and quality, which may potentially affect the performance of the proposed neural network-based approach. However, it should be noted here that the images used in this study for training and validation of the neural networks were collected from nine different past earthquakes, and the resulting datasets contained huge variations in image resolution, lighting condition, blurring, and other distortions. This enriched the database with diversity and made the neural network robust against previously unseen data and also added to its generalization ability. Relatively large dispersal in the evaluation metrics as observed in Figure 8 and Tables 2–4 are a direct consequence of this diversity. Damage detection on a number of images captured by an UAV-mounted camera in the aftermath of Taiwan earthquake (2016) is presented in Figure 11. The damages were detected by the trained Faster RCNN algorithm with Inception-ResNet-v2 as backbone architecture. Limited computation capability of on-board processing units is another bottleneck in robot-based

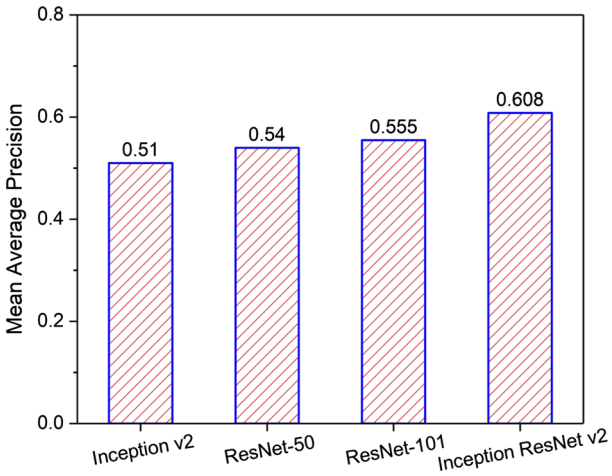

FIGURE 9 Comparison of mean average precision (MAP) for different CNN architectures

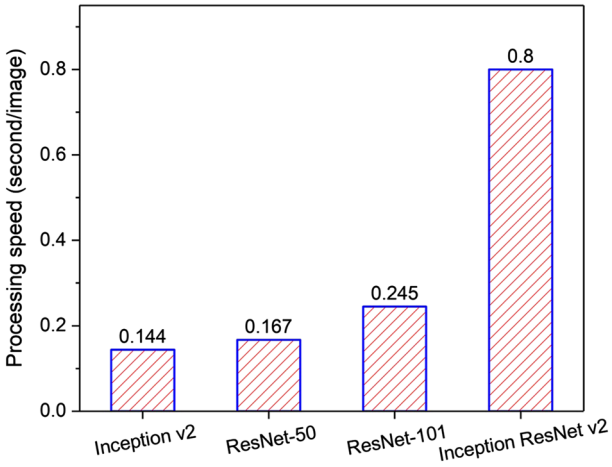

FIGURE 10 Comparison of processing speed for different CNN architectures

FIGURE 11 Damage detection results for images captured by a UAV post Taiwan earthquake (2016). Predicted boxes for Damage 1 (surface crack), Damage 2 (spalling), Damage 3 (spalling with exposed rebars), and Damage 4 (severely buckled rebars) are shown in different colors

real-time damage diagnosis. Commercially available portable power-efficient embedded Artificial Intelligence computing devices such as NVIDIA Jetson TX2 can provide a viable solution to this problem and is a scope for future research. Future studies should also focus on making the network smaller, faster, and consequently more suitable for on-board real-time computation by pruning of redundant neurons, which do not contribute significantly to the network outputs, as demonstrated by Wu, et al.[44]

7 | CONCLUSIONS

Faster RCNN algorithm is used in this study to detect multiple damage categories in RC buildings. Four different CNN architectures, namely, Inception v2, ResNet-50, ResNet-101, and Inception-ResNet-v2 are exploited to this end. A pre-trained model was used for the initialization of the network weights, which were subsequently fine-tuned by stochastic gradient descent optimization approach with momentum. The networks were trained using image data collected from several past earthquakes, namely, Nepal (2015), Taiwan (2016), Ecuador (2016), Erzincan (1992), Duzce (1999), Bingol (2003), Peru (2007), Wenchuan (2008), and Haiti (2010). Four different damage categories were considered in this study, namely, surface crack, spalling, spalling with exposed rebars, and severely buckled rebars. The performance of the trained networks was evaluated on the validation dataset. It was observed that Inception-ResNet-v2 significantly outperforms the other networks considered in this study producing a MAP value of 60.8%. It was also noted that the processing speed of the detection algorithms reduces with increase in accuracy. The authors believe that this study will broaden the scope for vision-based autonomous inspection of civil infrastructures.

Semantic segmentation of earthquake-induced building damages is a scope for future work. In segmentation-based algorithms, each pixel in an image is classified and labeled according to the class it represents. Therefore, it has the ability to predict the shape of a damaged area more accurately than bounding box-based approaches such as Faster RCNN. It may immensely benefit vision-based structural inspection, given that the shape of an damaged region is a powerful discriminator among different damage categories relevant to earthquake reconnaissance of RC buildings. For instance, shear cracks are preeminently diagonal, whereas flexural cracks usually spread in vertical or horizontal direction. This will facilitate finer level of categorization of various damages commonly observed in RC buildings post earthquake events. Additionally, it will help quantifying the severity of damage through autonomous evaluation of crack thickness and spalling area. Future studies should also explore the possibility of improving the detector performance by implementing Bayesian data fusion as proposed by Chen and Jahanshahi.[24] Aggregating the detection scores of a damaged region photographed from disparate camera positions may eliminate some of the false detections leading to an improved detection accuracy. Future studies should also focus on the practical implementation of this detection algorithm by integrating it with UAVs or inspection robots for real hands-on experiments.

ORCID

Tarutal Ghosh Mondal https://orcid.org/0000-0003-2091-7046
Mohammad R. Jahanshahi https://orcid.org/0000-0001-6583-3087

REFERENCES

1. Yamaguchi T, Hashimoto S. Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Mach Vis Appl*. 2010;21(5):797-809.
2. German S, Brilakis I, DesRoches R. Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety assessments. *Adv Eng Inform*. 2012;26(4):846-858.
3. Abdel-Qader I, Abudayyeh O, Kelly ME. Analysis of edge-detection techniques for crack identification in bridges. *J Comput Civ Eng*. 2003;17(4):255-263.
4. Amhaz R, Chambon S, Idier J, Baltazart V. Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection. *IEEE Intell Transp Syst*. 2016;17(10):2718-2729.
5. Avila M, Begot S, Duculty F, Nguyen TS. 2d image based road pavement crack detection by calculating minimal paths and dynamic programming. In: Image processing (icip), 2014 ieee international conference on IEEE; 2014:783-787.
6. Buza E, Omanovic S, Huseinovic A. Pothole detection with image processing and spectral clustering; 2013:48-53.
7. Koch C, Brilakis I. Pothole detection in asphalt pavement images. *Adv Eng Inform*. 2011;25(3):507-515.
8. Ying L, Salari E. Beamlet transform-based technique for pavement crack detection and classification. *Comput Aided Civ Infrastruct Eng*. 2010;25(8):572-580.
9. Zalama E, Gómez-García-Bermejo J, Medina R, Llamas J. Road crack detection using visual features extracted by gabor filters. *Comput Aided Civ Infrastruct Eng*. 2014;29(5):342-358.
10. Zou Q, Cao Y, Li Q, Mao Q, Wang S. Cracktree: Automatic crack detection from pavement images. *Pattern Recogn Lett*. 2012;33(3):227-238.
11. Sinha SK, Fieguth PW, Polak MA. Computer vision techniques for automatic structural assessment of underground pipes. *Comput Aided Civ Infrastruct Eng*. 2003;18(2):95-112.
12. Zhang W, Zhang Z, Qi D, Liu Y. Automatic crack detection and classification method for subway tunnel safety monitoring. *Sensors*. 2014;14(10):19307-19328.
13. Chen P-H, Shen H-K, Lei C-Y, Chang L-M. Support-vector-machine-based method for automated steel bridge rust assessment. *Autom Constr*. 2012;23:9-19.
14. Cord A, Chambon S. Automatic road defect detection by textural pattern recognition based on adaboost. *Comput Aided Civ Infrastruct Eng*. 2012;27(4):244-259.
15. Ouma YO, Hahn M. Pothole detection on asphalt pavements from 2d-colour pothole images using fuzzy c-means clustering and morphological reconstruction. *Autom Constr*. 2017;83:196-211.
16. O'Byrne M, Schoefs F, Ghosh B, Pakrashi V. Texture analysis based damage detection of ageing infrastructural elements. *Comput Aided Civ Infrastruct Eng*. 2013;28(3):162-177.
17. Gopalakrishnan K, Khaitan SK, Choudhary A, Agrawal A. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Constr Build Mater*. 2017;157:322-330.
18. Zhang A, Wang KC, Li B, Yang E, Dai X, Peng Y, Fei Y, Liu Y, Li JQ, Chen C. Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network. *Comput Aided Civ Infrastruct Eng*. 2017;32(10):805-819.
19. Zhang L, Yang F, Zhang YD, Zhu YJ. Road crack detection using deep convolutional neural network. In: Image processing (icip), 2016 ieee international conference on IEEE; 2016:3708-3712.
20. Cha Y-J, Choi W, Suh G, Mahmoudkhani S, Büyüköztürk O. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Comput Aided Civ Infrastruct Eng*. 2017;33:731-747.
21. Cha Y-J, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. *Comput Aided Civ Infrastruct Eng*. 2017;32(5):361-378.
22. Yeum CM, Dyke SJ, Ramirez J. Visual data classification in post-event building reconnaissance. *Eng Struct*. 2018;155:16-24.
23. Kim H, Ahn E, Shin M, Sim S-H. Crack and noncrack classification from concrete surface images using machine learning. *Structural Health Monitoring*. 2018;18:725-738.
24. Chen F-C, Jahanshahi MR. NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion. *IEEE Trans Ind Electron*. 2018;65(5):4392-4400.
25. Hoskere V, Narazaki Y, Hoang T, Spencer Jr B. Vision-based structural inspection using multiscale deep convolutional neural networks. arXiv preprint arXiv:1805.01055; 2018.
26. Gao Y, Mosalam KM. Deep transfer learning for image-based structural damage recognition. *Comput Aided Civ Infrastruct Eng*; 33:748-768.
27. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the ieee conference on computer vision and pattern recognition; 2016:2818-2826.
28. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the ieee conference on computer vision and pattern recognition; 2016:770-778.
29. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, inception-resnet and the impact of residual connections on learning. In: Aaai, Vol. 4; 2017:12.

30. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems; 2015:91-99.
31. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the ieee conference on computer vision and pattern recognition; 2014:580-587.
32. Girshick R. Fast R-CNN. In: Proceedings of the ieee international conference on computer vision; 2015:1440-1448.
33. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: Proceedings of the ieee conference on computer vision and pattern recognition; 2015:1-9.
34. Shah P, Pujol S, Puranam A. Database on performance of high-rise reinforced concrete buildings in the 2015 Nepal earthquake. https://datacenterhub.org/resources/242; 2015.
35. Shah P, Pujol S, Puranam A, Laughery L. Database on performance of low-rise reinforced concrete buildings in the 2015 Nepal earthquake. https://datacenterhub.org/resources/238; 2015.
36. Sim C, Song C, Skok N, Irfanoglu A, Pujol S, Sozen M. Database of low-rise reinforced concrete buildings with earthquake damage. https://datacenterhub.org/resources/123; 2015.
37. Sim C, Villalobos E, Smith JP, Rojas P, Pujol S, Puranam AY, Laughery L. Performance of low-rise reinforced concrete buildings in the 2016 Ecuador earthquake. https://datacenterhub.org/resources/14160; 2016.
38. Abadi M, Agarwal A, Barham P, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467; 2016.
39. Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122; 2015.
40. Hosang JH, Benenson R, Schiele B. Learning non-maximum suppression.Cvpr pp. 6469-6477; 2017.
41. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P., Zitnick CL. Microsoft coco: Common objects in context. In: European conference on computer vision Springer; 2014:740-755.
42. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics; 2010:249-256.
43. Bottou L. Large-scale machine learning with stochastic gradient descent. In: Y Lechevallier, G Saporta, eds. *Proceedings of compstat'2010*: Springer; 2010:177-186.
44. Wu R-T, Singla A, Jahanshahi MR, Bertino E, Ko BJ, Verma D. Pruning deep convolutional neural networks for efficient edge computing in condition assessment of infrastructures. *Computer-Aided Civil and Infrastructure Engineering*. 2019;34:774-789.

How to cite this article: Ghosh Mondal T, Jahanshahi MR, Wu R-T, Wu ZY. Deep learning-based multi-class damage detection for autonomous post-disaster reconnaissance. *Struct Control Health Monit*. 2020;e2507. https://doi.org/10.1002/stc.2507