Computers and Structures 315 (2025) 107761

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/cas

Deep learning-based post-earthquake structural damage level recognition

Xiaoying Zhuang [a, d, *], Than V. Tran [a], H. Nguyen-Xuan [b], Timon Rabczuk [c]

[a] Institute of Photonics, Leibniz University Hannover, Hannover 30167, Germany
[b] CIRTech Institute, HUTECH University, Ho Chi Minh City 700000, Vietnam
[c] Institute of Structural Mechanics, Bauhaus University Weimar, Weimar 99423, Germany
[d] Department of Geotechnical Engineering, College of Civil Engineering, Tongji University, Shanghai 200092, China

HIGHLIGHTS

- Evaluate and compare the efficiency of state-of-the-art pre-trained CNNs.
- Employ transfer learning and k-fold cross-validation for small datasets.
- Optimize CNN hyperparameters using Bayesian optimization.
- Use Grad-CAM to highlight key regions in structural damage images.
- Enhance and assess model generalizability on an extended dataset.

ARTICLE INFO

Keywords:
Deep learning
Damage level recognition
Image classification
Transfer learning
Convolutional neural network
Bayesian optimization

ABSTRACT

Rapid assessment of building damage levels has become very important and has received considerable attention in structural engineering. Traditional methods for this work involve manual inspection, which is often tedious and time-consuming. Deep learning technology in computer vision has developed rapidly in recent years and has proven its superiority. This paper aims to develop an efficient approach to recognize quick post-earthquake structural damage levels. First, we develop a feature extraction with seven pre-trained CNN models (Xception, InceptionV3, InceptionResNetV2, MobileNet, MobileNetV2, DenseNet121, NASNetMobile) on a small dataset of 2000 images. The CNN models are then trained by five fold cross-validation. The performance of the models is compared on a testing set, the MobileNet model demonstrated the best classifier performance with an accuracy of 90.89 %. Second, the Bayesian optimization method and the fine-tuning strategy are used to find the optimal hyperparameters of the MobileNet model. The results revealed that the performance of the MobileNet model increased significantly with an accuracy of 96.11 %. Third, Gradient-weighted class activation mapping (Grad-CAM) is used to highlight crucial regions on structural damage images for CNN's prediction. Finally, the generalizability of the MobileNet model is improved by training it on an extended dataset of 3600 images. The proposed approach demonstrates the feasibility and potential uses of deep learning in image-based structural damage level recognition.

1. Introduction

Earthquake is one of the most common disasters in the world. It can occur several times a year and negatively affects millions of people [1]. The immediate danger caused by earthquakes is reflected in damaged and collapsed structures, causing significant damage to property and human life. When an earthquake occurs, emergency responders, as well as local and government officials, need accurate information to make informed and timely decisions. Therefore, a quick and accurate assessment of structural damage to buildings affected by earthquakes plays an important role in recovering from an earthquake.

Nowadays, the post-earthquake damage inspection is done manually by dispatching a team of inspectors; this type of subjective assessment has been widely adopted due to its ease of implementation. However, such an implementation is inefficient because it is extremely susceptible to the personal experience and knowledge of the inspectors. On the

* Corresponding author at: Institute of Photonics, Leibniz University Hannover, Hannover 30167, Germany.
Email address: zhuang@iop.uni-hannover.de (X. Zhuang).

https://doi.org/10.1016/j.compstruc.2025.107761
Received 19 January 2025; Accepted 2 April 2025
Available online 25 April 2025

other hand, manual implementations are time-consuming and expensive as they require a lot of manpower, which makes them unfeasible to support emergency response planning and early recovery. Much effort has been made to automate inspection activities using computer vision-based approaches to solve problems associated with tedious and inefficient manual work [2–4]. Most of these approaches are based on traditional machine learning and image processing methods. Therefore, the aforementioned solutions are generally still time-consuming and may not be effective when applied in practice because it is inevitable that there are defects in complex real-world conditions with background noise.

In recent years, thanks to tremendous growth in computer hardware performance and a boost from the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [5], since 2012, deep learning (DL) has been widely used in many fields and achieves performance far superior to traditional methods [6–11], in which convolutional neural networks (CNNs) are at the heart of spectacular advances in DL [12–18]. Numerous works have illustrated that CNN has been becoming an increasingly popular and powerful tool for damage recognition tasks. Cha et al. [19] proposed a vision-based method using a deep CNN architecture for identifying concrete cracks. Park et al. [20] used CNN-based YOLO models to detect potholes in roads. Liang [21] proposed an image-based approach using CNN networks with Bayesian optimization to inspect reinforced concrete bridge damage after disasters. Ghosh Mondal et al. [22] proposed the use of a Faster Region-based Convolutional Neural Network (Faster R-CNN) to detect and classify damages to buildings automatically after disasters. Kim et al. [23] used a deep CNN to classify steel frame damages. Chen et al. [24] used the Inception-ResNet-V2 network to classify multiple rock structures of tunnel faces. Rosso et al. [25] proposed a CNN network with pre-trained ResNet-50 architecture and transformers for automatic road tunnel defect classification. Ni et al. [26] proposed a physics-informed CNN network to estimate traffic-induced-bridge displacement components. Xu et al. [27] proposed a multi-task learning method for the post-earthquake evaluation of RC structural components. Kim et al. [28] proposed an automatic multi-damage detection technique based on Mask R-CNN for instance segmentation. Kumar et al. [29] used the edge computing principle to propose a real-time concrete damage detection system based on YOLO-v3. Zhu et al. [30] used three object detection algorithms—Faster R-CNN, YOLOv3, and YOLOv4—for pavement distress detection. Guan et al. [31] proposed a pavement distress detection framework based on stereo vision and a modified U-net network. Liu et al. [32] proposed the Feature Pyramid Network (FPN) for asphalt pavement crack detection based on CNN and infrared thermography. Huyan et al. [33] proposed a CrackU-net based on deep CNN for pixel-wise pavement crack detection. Liu et al. [34] proposed a two-step pavement crack detection and segmentation approach based on modified YOLO-v3 and U-Net. Xu et al. [35] proposed a task-aware meta-learning approach for multi-type structural damage segmentation. Liu et al. [36] used many CNN models to classify asphalt pavement crack severity. Tan et al. [37] developed a LinkNet model incorporating the attention mechanisms for high-resolution sonar detection of local damage in underwater structures. Yang et al. [38] presented an improved real-time anchor-free damage detection method called YOLOv6s-GRE for real-time multi-class damage detection. Agyemang et al. [39] introduced ExpoDet, a comprehensive framework designed for autonomous health assessment of infrastructure. Xu et al. [40] developed a lightweight semantic segmentation approach based on DeepLabv3+ to recognize complex structural damage for actual bridges. In addition, some other notable studies have been done, such as [41–49].

To our knowledge, the CNN platform and its application in post-earthquake structural damage assessment are still limited and need improvement in terms of performance existing in the literature. Therefore, research in this field remains necessary and desirable. Nowadays, state-of-the-art models and transfer learning techniques are especially important in cases where available data is scarce and training costs and time are limited. Leveraging advanced methods and techniques should be a priority. Using Bayesian optimization will boost the model's performance for a specific task. In this paper, a fast and accurate DL image-based approach is proposed to automate the process of assessing the structural damage of buildings post-earthquakes. The outstanding contribution of this study is to consider well-established CNN architectures, which are pre-trained on large image datasets (e.g., ImageNet) and have proven effective for images, and then explore Bayesian optimization to fine-tune hyperparameters in the transfer learning process of the CNN model to improve the performance and accuracy of the model. This has not been done completely in previous studies on the classification task of the level of structural damage. The robustness and generalization of the proposed model are verified in a testing set, which is obtained from a completely independent source from the training and validation datasets. The main contributions of this paper are as follows:

- The efficiency and robustness of the state-of-the-art pre-trained CNN models are evaluated and compared with each other.
- Transfer learning strategy and k-fold cross-validation technique are used for a small dataset when training a CNN model.
- Bayesian optimization is used to automatically tune the hyperparameters of a CNN model to improve its performance.
- Gradient-weighted class activation mapping (Grad-CAM) is used to highlight crucial regions on structural damage images for CNN's predictions.
- The generalizability of the proposed model is improved and discussed on an extended dataset.

The remainder of the paper is organized as follows: Section 2 presents the methodology in detail, including the basic architecture of a CNN model, transfer learning strategy, k-fold cross-evaluation technique, Bayesian optimization method, Grad-CAM technique, and metrics to evaluate a classification model. In Section 3, the used dataset and the results of the experiments are described in detail. The results of visualizing and understanding the decision-making process of CNN using the Grad-CAM technique are shown in Section 3. Finally, conclusions and future work are outlined in Section 4.

2. Methodology

2.1. CNN architecture

CNN is a class of neural networks for DL that learns directly from data. It was first used in the 1990 by LeCun et al. [50] to solve the handwritten digit recognition task. CNN is distinguished from other neural networks by its superior performance in finding image patterns for object recognition. The CNN architecture for image classification typically consists of convolutional layers (convolution and pooling), fully connected (FC) layers, and an output layer, as shown in Fig. 1. Convolution and pooling layers are the core building blocks of CNN to extract various features from input images. While convolution layers are used for feature detection, pooling layers are used for feature selection. This reduces the spatial size of the feature map to lower computational costs. The output from the convolutional layers is fed into the FC layers. These layers are usually placed before the output layer for classification.

2.2. Transfer learning

Although DL has made significant progress in various domains in recent years, its effectiveness is still greatly influenced by the quantity and quality of the data that is available. Useful data for practical applications can occasionally be quite expensive to obtain, and it is typically only possible to gather limited amounts of data. To minimize reliance on available data size and to maximize the utilization of existing data, transfer learning (TL) has become a highly efficient approach [16]. TL utilizes stored data or knowledge that has been obtained from one or more source domains and applies it to a new target domain [51]. This is illustrated in Fig. 2, where two convolution layers are retrained.

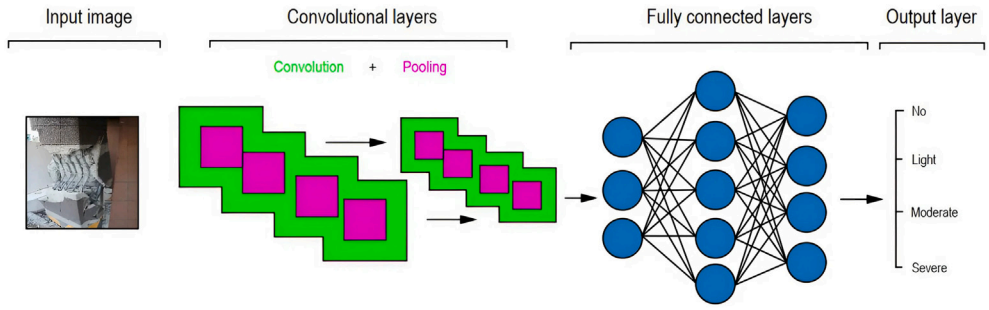


Fig. 1. Basic CNN architecture.

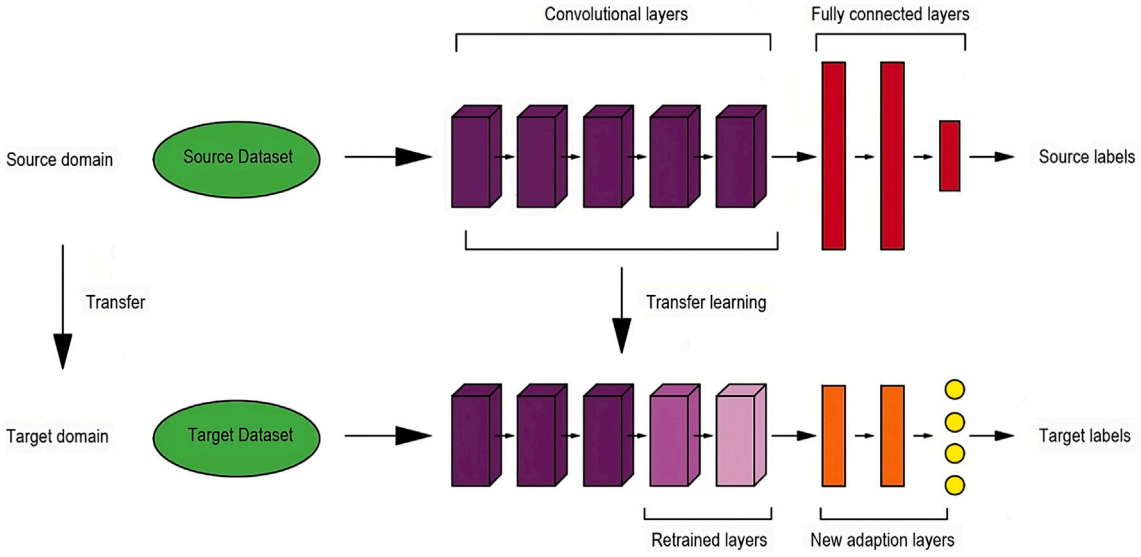


Fig. 2. Transfer learning flowchart.

The idea behind TL for image classification tasks is that if a model is trained on a large enough dataset with adequate generality, it will essentially act as a generic model of the visual world. We can then leverage these learned feature maps without having to start from scratch by training a deep CNN on a huge dataset. As a result, TL reduces the number of training parameters, thereby speeding up the model training process and saving computational resources. In addition, TL can solve the overfitting phenomenon and training difficulty (saturation of training accuracy) caused by the deficiency of training data. Consequently, the efficiency of the training process is significantly improved.

In general, feature extraction and fine-tuning are the two main approaches used when conducting TL in pre-trained deep CNN models. Which approach to use depends on two main factors: the size of the dataset in the target domain, and the similarity between the dataset in the target domain and the dataset in the source domain. Configuration and procedure for the training of the two TL strategies (feature extraction and fine-tuning) will be detailed in Section 3.

2.3. K-fold cross-validation

K-fold cross-validation is a method used to estimate the performance of machine-learning (ML) models on unseen data [52]. Usually, when training an ML model, the data is divided into three sets (training, validation, and testing). As a result, the number of samples that can be utilized to train an ML model is drastically reduced, and the outcomes of the model depend on the random selection of the training and validation sets. So, when there is scarce available data, it is advised to utilize this strategy since it is simple to comprehend. The overfitting phenomenon, which can happen when a model is trained using all of the data, can be prevented with the use of this strategy. We can "test" the model on k distinct data sets using k-fold cross-validation, which aids in ensuring that the model is generalizable.

The procedure of k-fold cross-validation with $k = 5$ is illustrated in Fig. 3. According to the basic approach, a testing set should be kept separate for final evaluation. The training set is divided into k separate smaller folds, where k–1 folds are used to train the model, and the remaining fold is used for model validation. The average of the values calculated in the loop is the performance metric supplied by k-fold cross-validation.

2.4. Bayesian optimization

Bayesian optimization is a robust method for optimizing derivative-free objective functions that are costly to assess based on scarce and noisy data. It has become extremely prevalent for tuning hyperparameters in ML algorithms, notably deep neural networks [53,54]. Bayesian optimization makes a distinction from other surrogate approaches by utilizing surrogates created using Bayesian statistics and choosing where to assess the objective using the Bayesian interpretation of these surrogates.

Bayesian optimization consists of two fundamental parts: a Bayesian statistical model, often Gaussian process regression for simulating the objective function, and an acquisition function for selecting the location of the subsequent sample, which is frequently an improvement predicted. After the objective is evaluated by an initial space-filling

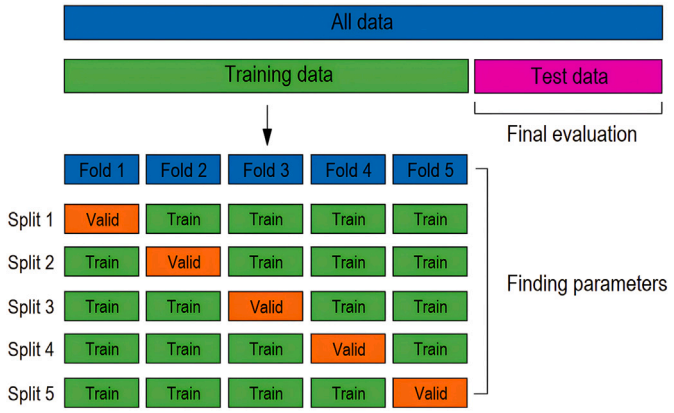


Fig. 3. Five fold cross-validation process.

experimental design, which frequently consists of points selected uniformly at random, they are used repeatedly to distribute the remaining budget for function evaluations. The process of performing Bayesian optimization is a process that is repeated until the extreme of the objective function is determined (the result is good enough, or the resources are exhausted). It can be summarized in three steps as follows:

- Optimize the acquisition function υ over the Gaussian process to determine x_l: $x_l = argmax_x \upsilon(x|K_{1:l-1})$, where $K_{1:l-1} = (x_1, y_1), (x_2, y_2), \ldots, (x_{l-1}, y_{l-1})$ are the $l-1$ accumulated observation samples drawn from the objective function g.
- Get a possibly noisy sample y_l from the objective function g: $y_l = g(x_l) + \eta_l$, with $\eta_l \sim N(0, \sigma^2_{noise})$.
- Add the data $K_{1:l} = \{K_{1:l-1}; (x_l, y_l)\}$ and update the Gaussian process.

A single iteration of Bayesian optimization utilizing the Gaussian process regression and anticipated improvement is shown in Fig. 4. Fig. 4a displays noise-free observations of the objective function g at three locations represented by green circles; an estimate of g(x) is shown as a solid orange line, and Bayesian confidence ranges for g(x) are shown as dotted orange lines. Gaussian process regression is used to derive these estimates and confidence ranges. The anticipated improved acquisition function that corresponds to this posterior is depicted in Fig. 4b. The point where the acquisition function is maximized, denoted in this case by an "x", is where Bayesian optimization decides to sample next.

2.5. Grad-CAM

Grad-CAM [55] is a generalization of CAM [56]. It is a common method for increasing the transparency of CNN-based models by displaying the input areas that are "critical" for the models' predictions or, more simply expressed, by providing visual explanations. Unlike CAM, we obtain these visualizations without modifying the base model or retraining the model. An overview of Grad-CAM for a CNN-based image classification network is shown in detail in Fig. 5.

As illustrated in Fig. 5, when an image and a target class (e.g., "damage") are provided, the image is passed through the CNN layers, followed by the FC layers, to generate a raw classification score for that class. Except for the desired class (damage), which has a gradient set to one, other classes have gradients set to zero. This information is then back-propagated into the rectified convolutional feature maps of our concern, which are integrated to calculate the rough Grad-CAM localization (heat map) that shows where the network must search to reach the specific conclusion. Lastly, the heat map was point-wise multiplied with guided

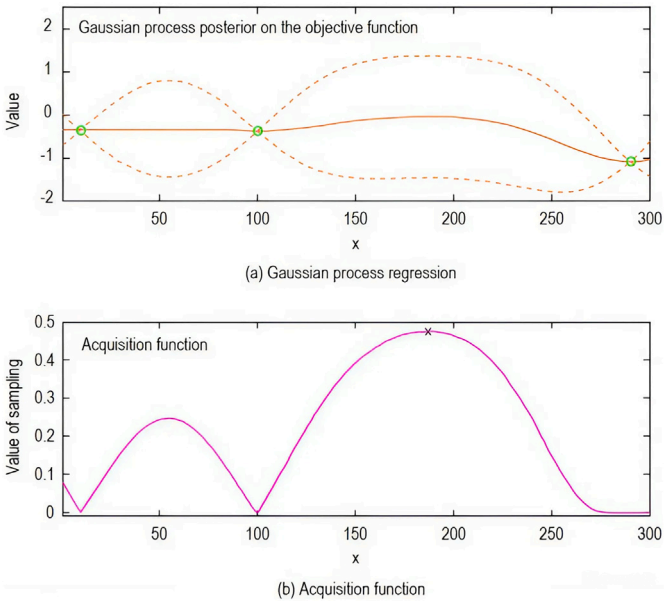


Fig. 4. Illustration of using Bayesian optimization to maximize an objective function g with a 1D continuous input: (a) Gaussian process regression, (b) acquisition function.

back-propagation to obtain Guided Grad-CAM visuals, which are both high-resolution and concept-specific.

2.6. Evaluation metrics

In classification tasks, to evaluate the model's performance, we usually use four basic metrics, namely accuracy, precision, recall, and F1-score [57]. These evaluation metrics are calculated using the basic terminology used in the confusion matrix: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), as described below:

- Accuracy is the most straightforward and widely used evaluation metric for classification tasks. This measure is simply the ratio between the number of correctly classified samples and the total number of samples. It is extremely helpful for problems where the sizes of the data classes are roughly the same.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

- When the size of the data classes is imbalanced or skewed, precision is a more advanced metric that is a better choice than accuracy. Precision is defined as the proportion of true positives (TP) to all positive classifications (TP + FP). It assesses how well the model performs in identifying just the relevant data samples.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

- Similar to precision, recall is used when the size of the data classes is imbalanced or skewed. Recall is defined as the proportion of true positives (TP) to all that are actually positive (TP + FN). It measures a model's capacity to look for all related samples within a dataset.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

- A good classification model should have both high recall and high precision, ideally as near to one as possible. To have a trade-off between precision and recall, F1-score is utilized to find the ideal blend.

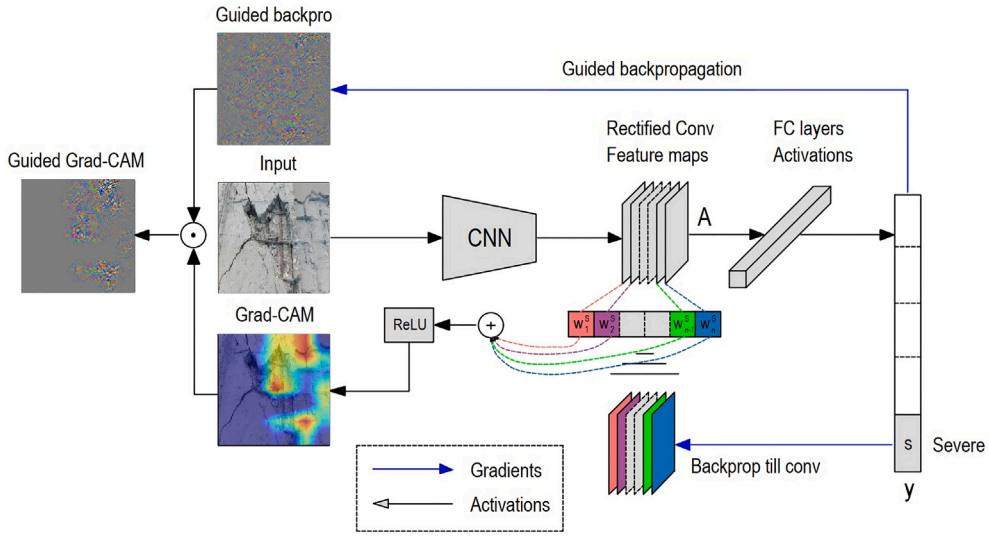


Fig. 5. Grad-CAM overview for image classification network.

It is the harmonic mean of precision and recall, which might penalize extreme values (FP and FN).

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

Along with the aforementioned metrics, the area under the curve (AUC) property for the receiver operating characteristic (ROC) curve is another useful metric for evaluating the effectiveness of ML classifiers [58]. The AUC aids in evaluating how effectively a classifier can discriminate between different classes. There, the true positive rate (TPR) and false positive rate (FPR) parameters are used to plot the ROC curve. It displays how well a classification model performs across all categorization thresholds.

3. Experiments

In this section, the dataset and the results of the experiments are described in detail. The results of visualizing and understanding the decision-making process of CNNs using the Grad-CAM technique are also depicted in this section. The choice of CNN architectures is based on consideration of various factors to ensure the model is suitable for the structural damage classification task, including analyzing the nature of the dataset (size, diversity, complexity, number of classes, etc.), using pre-trained models on large image datasets (e.g., ImageNet) when available data are scarce, considerong well-established CNN architectures that have proven effective in image classification, considering model size (number of parameters), and the efficiency of fine-tuning the model hyperparameters. The schematic of the proposed approach is illustrated in Fig. 6. First, the proposed method is developed using seven pre-trained CNN models such as Xception [59], InceptionV3

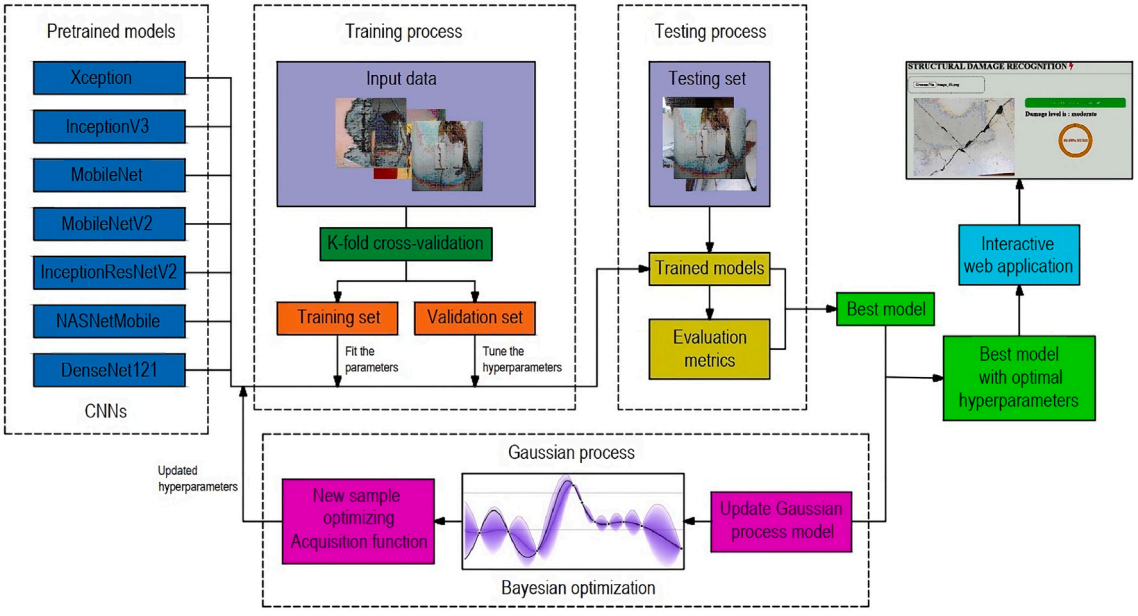


Fig. 6. The schematic of our study using pre-trained models with Bayesian optimization.

Table 1
Structural damage dataset used in this study.

Image	No damage	Light damage	Moderate damage	Severe damage
Training	300	500	500	700
Testing	45	45	45	45
Total	345	545	545	745

[60], InceptionResNetV2 [61], MobileNet [62], MobileNetV2 [63], DenseNet121 [64], and NASNetMobile [65] on a small dataset of 2000 structural damage images. The pre-trained CNN models are trained using five fold cross-validation. The effectiveness and robustness of the models are evaluated and compared with each other on the testing set. Then, the model with the best performance is improved using the Bayesian optimization method to find the optimal hyperparameters for the model. Finally, the generalizability of the MobileNet model is improved by training it on an extended dataset of 3600 images.

3.1. Dataset

In this study, the effectiveness of the proposed approach is assessed using images of structures damaged by recent earthquakes in various locations worldwide, including Haiti, Nepal, Taiwan, Ecuador, and Pohang. The dataset, provided by Ogunjinmi et al. [66], is categorized into four damage levels: no damage, light damage, moderate damage, and severe damage, following the ATC-58 guidelines. A total of 2180 images were used, with 2000 images allocated for training and validation and the remaining 180 images, from the Pohang earthquake database, used to evaluate the generalizability of the model. The categories of the used dataset and the distribution of images by category are summarized in Table 1. All images were resized to 224×224 pixels to match the input dimensions of pre-trained CNN models and normalized to a range of [01] to reduce overfitting and enhance computational efficiency. This study focuses exclusively on observable damage patterns captured from post-earthquake images for rapid assessment during emergency scenarios. Unlike methods incorporating earthquake parameters such as magnitude, epicentral distance, or ground motion duration, this approach simplifies the input requirements by relying solely on observable damage outcomes. The dataset labeling by structural engineering experts ensures real-world relevance, capturing nuanced damage levels directly from visual inspection.

Our dataset preparation and processing methodology share similarities with recent studies such as Verma et al. [67], which employed CNNs to classify electroluminescence images of silicon solar cells, and Siruvuri et al. [68], which used deep learning models trained on molecular dynamics simulations for crack analysis in photovoltaic cells. While these studies focus on micro-crack detection or material-level analyses, our work adapts these methods to the classification of post-earthquake structural damage at a building scale. The four damage categories in our study are defined as follows: light damage refers to hairline fractures; moderate damage includes broader cracks and concrete spalling; severe damage encompasses structural collapse or failure. These classifications are based on visual inspection patterns, guided by ATC-58 standards, and aim to enable rapid, image-based assessments with minimal input requirements. Representative examples for each category are shown in Fig. 7. This streamlined approach ensures practical applicability, particularly in time-sensitive disaster response scenarios, while leveraging advanced image-processing techniques inspired by related works.

3.2. Results and discussion

In this subsection, the results of three experiments are presented and discussed. In the first experiment, seven pre-trained CNN models are trained, and the models' performance is evaluated and compared. In the second experiment, the model with the best performance in the first experiment is improved using Bayesian optimization to find the optimal hyperparameters for the model. The results of the best model with optimal hyperparameters are presented and discussed in the third experiment. Data augmentation techniques such as flipping, rotation, translation, cropping, and scaling are used in the training process of the models in all three experiments [69]. These techniques are used to expand the size of the training set (especially useful when the initial dataset is small) to improve the performance and generalizability of the model. The experiments were conducted on an NVIDIA RTX 3090 GPU with 24 GB of VRAM, providing sufficient computational power for training. The CNN models required approximately 10 GB of GPU memory during training.

3.2.1. Evaluate the performance of pre-trained models

The proposed approach is developed using seven pre-trained CNN models (Xception, InceptionV3, InceptionResNetV2, MobileNet, MobileNetV2, DenseNet121, NASNetMobile) on a small dataset of 2000 structural damage images. The weights of these pre-trained models have been determined from being previously trained on the ImageNet database. This database contains more than 1.2 million images gathered from various domains and classified into 1000 object classes [5]. Given the strong generalization ability of the pre-trained models, we use these models for our dataset. Herein, the feature extraction strategy to customize these models for the structural damage classification task is implemented. The configuration of feature extraction is shown in Fig. 8.

As illustrated in Fig. 8, it is not necessary to retrain the entire model. The base convolutional and pooling layers already have features that are generally helpful for categorizing images. However, the last classifier of the pre-trained models is replaced by a new classifier with four nodes and will be retrained from scratch. With this training strategy, the number of parameters that need to be trained in the models is significantly reduced, as summarized in Table 2. This saves considerable training time and computational resources. From Table 2, it can be seen that all seven models have a total of several million to several tens of millions of parameters, of which MobileNet and MobileNetV2 models are the two models with the least total parameters compared to other models with just over 2.5 million and over 3.4 million parameters respectively. In contrast, the InceptionResNetV2 model has the largest total number of parameters, over 54 million. After implementing the feature extraction strategy, the number of trainable parameters is greatly reduced, to only a few hundred thousand parameters in all seven models. Herein, the lowest number of trainable parameters with more than 150,000 parameters is in the InceptionResNetV2 model, while the Xception model with the highest number of trainable parameters is just over 400,000 parameters. The rest of the models have more than 200,000 thousand trainable parameters.

The training of pre-trained models according to the feature extraction strategy was performed with 40 epochs for all seven models. The cross-entropy loss function is used as the objective function, which is most usually utilized for classification models that forecast probability. The Adam optimizer [70] is applied in this experiment which updates the parameters (weights) of the models to minimize the loss function. The training time required for feature extraction in CNN models is approximately 8–10 s per epoch. The performance of the models is evaluated on the testing set and it is summarized in Table 3. It can be observed that the InceptionResNetV2 model, which has the least number of trainable parameters, has the lowest performance with accuracy, precision, recall, and F1-score of 79.22 %, 84.69 %, 79.22 %, and 78.46 %, respectively. In contrast, the MobileNet model outperformed other models with accuracy, precision, recall, and F1-score of 90.89 %, 91.58 %, 90.89 %, and 90.94 %, respectively. This revealed that the MobileNet model is the best choice for the post-earthquake structural damage classification task, and it will be improved in performance in the next experiment by Bayesian optimization.

The architecture of MobileNet is illustrated in Fig. 9 [62]. Depthwise separable convolutions are used in the MobileNet model. This drastically

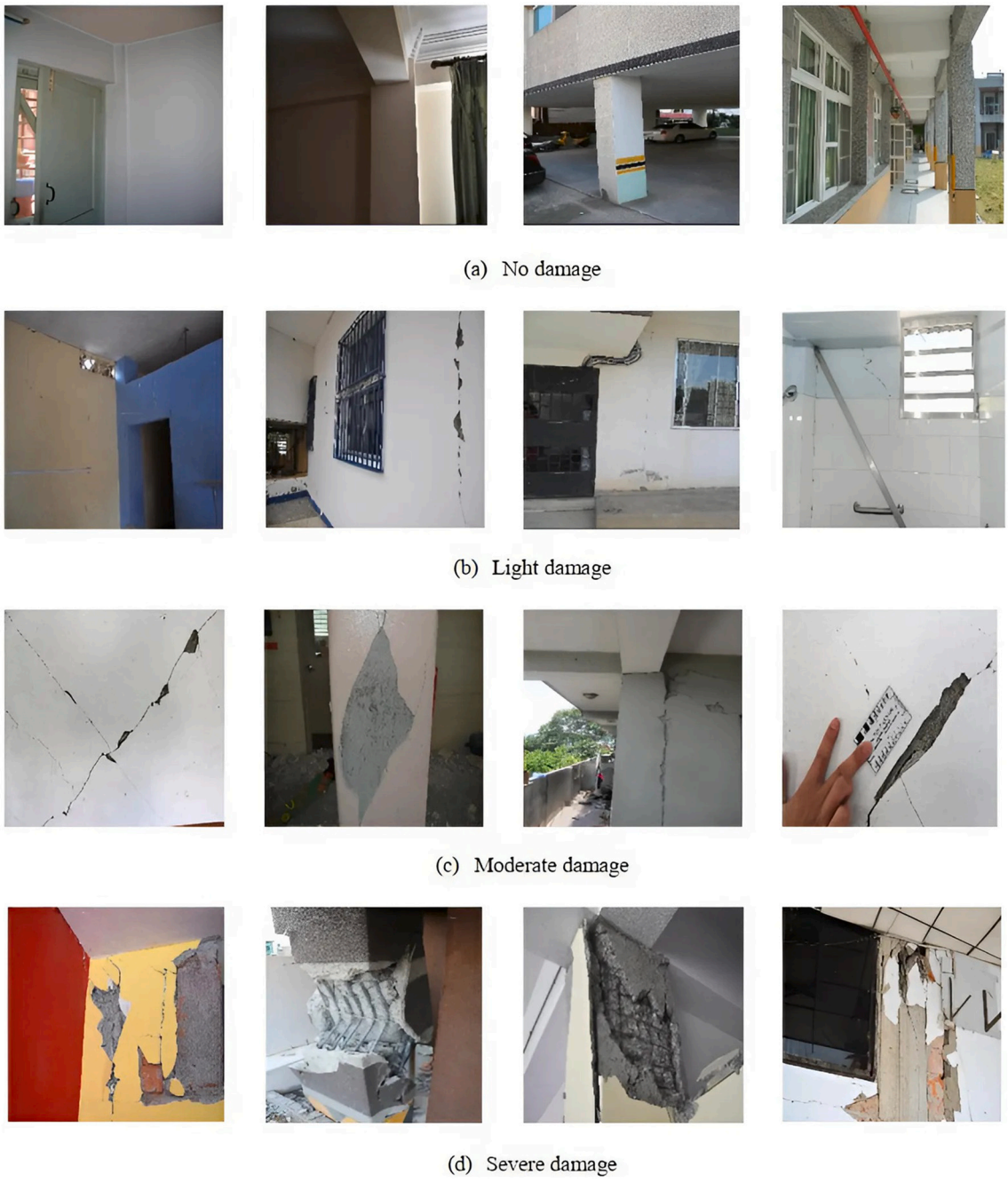


Fig. 7. Some samples of the training dataset: (a) no damage, (b) light damage, (c) moderate damage, (d) severe damage.

reduces the number of parameters compared to the networks with conventional convolutions of the same depth. As a result, the MobileNet network is a lightweight deep neural network.

3.2.2. Hyperparameters optimization result

The MobileNet model is improved using Bayesian optimization to find optimal hyperparameters, such as the number of dense layers, neurons per dense layer, activation per dense layer, dropout rate per dense layer, optimizer, learning rate, batch size, and the number of epochs. Herein, a fine-tuning strategy to customize the model for the post-earthquake structural damage classification task is implemented. The fine-tuned configuration is shown in Fig. 10. In the fine-tuning strategy, we need to unfreeze some of the top layers or the entire base model and train them together with the newly added classifier and the other adaptive layers, if any. By doing so, we can improve the higher-order feature representations in the base model and make it more useful for our structural damage classification task. It is noteworthy that the number of layers in the base model is 86, and we fine-tune from layer 78 onwards in this work. The training time required for fine-tuning MobileNet takes around 29 s per epoch.

Details of the search space for hyperparameters and the optimal results obtained from Bayesian optimization are shown in Table 4. It is clear from Table 4 that the optimal MobileNet architecture has two dense layers added between the flattened layer and the output layer. The first dense layer has 187 nodes that use the Exponential Linear Unit (ELU) activation function [71]. The second dense layer

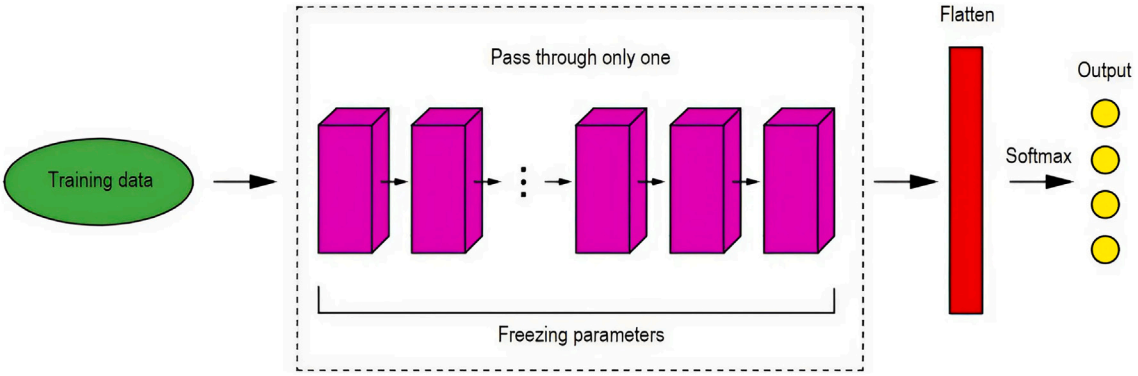


Fig. 8. Configuration of feature extraction.

Table 2
Summary of the number of parameters in models using feature extraction strategy.

Model	Non-trainable params	Trainable params	Total params
Xception	20,861,480	401,412	21,262,892
InceptionV3	21,802,784	204,804	22,007,588
InceptionResNetV2	54,336,736	153,604	54,490,340
MobileNet	3,228,864	200,708	3,429,572
MobileNetV2	2,257,984	250,884	2,508,868
DenseNet121	7,037,504	200,708	7,238,212
NASNetMobile	4,269,716	206,980	4,476,696

Table 3
Comparison of performance of various pre-trained models.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Xception	87.11	89.24	87.11	86.85
InceptionV3	86.00	87.19	86.00	85.71
InceptionResNetV2	79.22	84.69	79.22	78.46
MobileNet	90.89	91.58	90.89	90.94
MobileNetV2	88.11	89.12	88.11	88.16
DenseNet121	87.33	88.35	87.33	87.36
NASNetMobile	83.33	82.50	83.33	81.87

has 480 nodes, and Rectified Linear Unit (ReLU) is used as the activation function [72]. Neither of the dense layers uses dropout. Fig. 10 illustrates this optimal MobileNet architecture. RMSprop optimizer [73] is used with a learning rate of 10^{-4} during model training. The experiment is carried out with 70 epochs, and the batch size is 64.

3.2.3. MobileNet model with optimal hyperparameters

The MobileNet model with optimal hyperparameters obtained from Bayesian optimization will be trained in two cases. In the first case, we fine-tune the model from layer 84 (conv_pw_13) onwards to what we call 1 block + fc-layers. In the second case, we fine-tune the model from layer 78 (conv_pw_12) onwards to what we call 2 blocks + fc-layers. It should be noted that layers 78 and 84 have the most number of parameters in the base model compared to other layers with parameters of 524,288 and 1,048,576, respectively. Table 5 summarizes the number of parameters in the model using the fine-tuning strategy in the two training cases. It can be seen that the total number of parameters increased significantly from more than 3.4 million parameters (Table 2) to over 12.7 million parameters. In addition, the number of trainable parameters of the model for the case of fine-tuning one block and two blocks is more than 10.5 million and more than 11 million parameters, respectively. The difference in the number of trainable parameters in the two cases is not much (only about 500,000 parameters), but it is much larger than the number of non-trainable parameters. The main contribution to the increase in the number of parameters is the addition of two dense layers. However, with the total number of parameters at about 12.7 million, the network is still a lightweight deep neural network compared to other networks of the same depth.

Model fine-tuning is done with 70 epochs. Fig. 11 plots the loss and accuracy histories of the training and validation sets when retraining with one block and two blocks. The first observation is that the loss values decrease sharply at the early epochs and start to converge from the 20th epoch onwards, which indicates that the network is learning something. These loss values are very difficult to reduce further, this proves that the training process can be completed. Another observation is that in the accuracy values, these values increase rapidly in the early stage and start to converge from the 20th epoch onwards, and it is difficult to increase further if we continue training. Continuing to train further will easily cause overfitting. Further, it can be seen that from the 20th epoch onward, the loss curves on the training set behave very well and gradually decrease to zero, but the loss values on the validation set oscillate with the curves becoming rugged. This indicates that the overfitting phenomenon has occurred due to the lack of training data, which proves that the network is complex enough to learn something but lacks generalizability. However, it is observed that about 5 % overfitting is derived from the difference in accuracies between training and validation sets. This is a slight overfitting phenomenon. In addition, the accuracy results on the validation set are around 95 %, indicating that the degree of overfitting is still acceptable.

The confusion matrix is a common measure utilized to visualize and summarize the performance of a classification model on a testing set. It shows how many samples belong to a category, and are predicted to fall into a category. The normalized confusion matrix is more informative, and it shows the probability of true and false predictions with the values split by category. An effective classification model will have a confusion matrix with big values on the main diagonal, and small values (non-negative) on the remaining elements. The confusion matrices of the testing set for the two cases of model retraining are shown in Fig. 12. It can be observed that the values on the main diagonal are much larger compared to the values on the remaining elements in the confusion matrices. Furthermore, the values on the main diagonal in the case of retraining two blocks are slightly larger than those of one block. This demonstrates that the model retrained with two blocks gives better performance than one retrained with one block. Specifically, in the retrained model with two blocks, the model correctly predicted 42 samples (93 % correct prediction probability) out of a total of 45 samples that belong to the “light” class. The remaining 3 samples (a probability of incorrect prediction of 7 %), which the model predicts are of the “moderate” class, were misclassified by the model. The interpretation is similar for the other classes. In general, there are a few instances of FP and FN produced from the model in both retraining cases, which is easily observed from the confusion matrices.

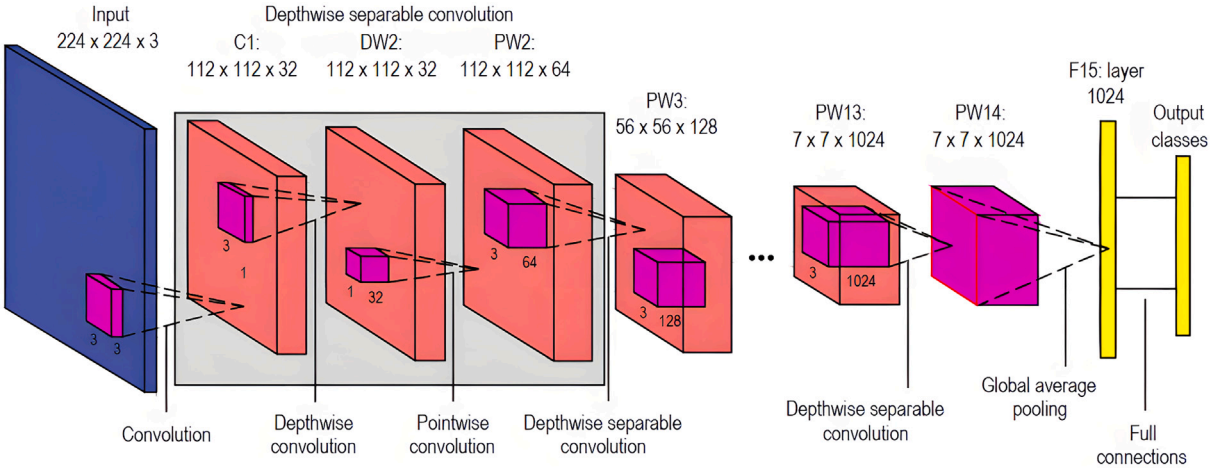


Fig. 9. MobileNet architecture.

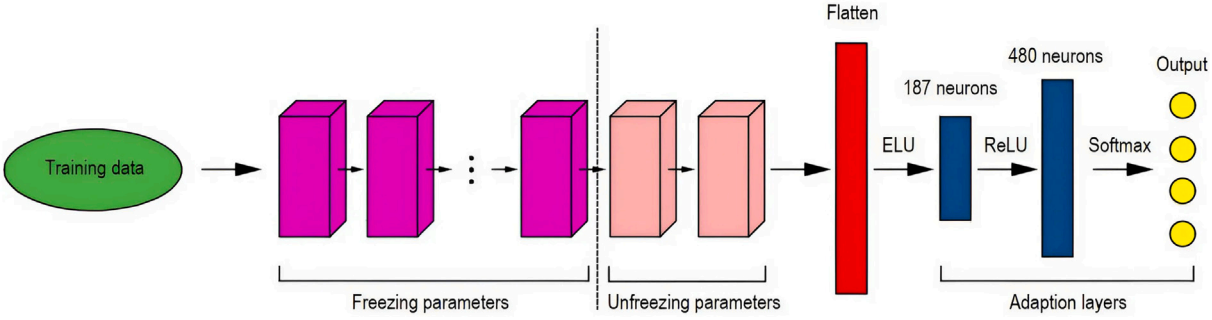


Fig. 10. Configuration of fine-tuning.

Table 4
Description of hyperparameters with search space and optimal results.

Hyperparameter	Search space	Optimal result
Number of dense layers	[0, 1, 2]	2
Neurons per dense layer	[64:512], step = 1	[187, 480]
Activation per dense layer	[sigmoid, tanh, ReLU, ELU, SELU]	[ELU, ReLU]
Dropout rate	[0, 0.1, 0.2, 0.3, 0.4, 0.5]	0
Optimizer	[SGD, AdaGrad, RMSprop, Adam]	RMSprop
Learning rate	[0.0001, 0.001, 0.01]	0.0001
Batch size	[8, 16, 32, 64]	64
Number of epoch	[30:80], step = 10	70

Table 5
Summary of the number of parameters in models using fine-tuning strategy.

Retrained blocks	Non-trainable params	Trainable params	Total params
One + FC-layers	1,636,544	11,067,583	12,704,127
Two + FC-layers	2,178,240	10,525,887	12,704,127

The ROC curves of the testing set for the two experimental cases are illustrated in Fig. 13. A classification model is effective when there is low FPR and high TPR; that is, there exists a point on the ROC curve that is close to the point with coordinates (0, 1) on the graph (in the upper left corner). The closer the curve is to that point, the more efficient the model is. AUC is a quantitative metric to evaluate the model instead of qualitatively, as in the ROC curve. This value is in the range [0, 1]; normally, it will be greater than or equal to 0.5 (equal to 0.5 when the model has learned nothing). The larger it is, the better the model. It is clear from Fig. 13 that in the model trained with one block, the AUC values of the "light", "moderate", "no", and "severe" classes are 0.99, 0.99, 1.00, and 1.00, respectively. These values are 1.00, 0.99, 1.00, and 1.00, respectively, in the model trained with two blocks. The AUC values in both experimental cases are generally large, which indicates that the classification model is good enough.

Four basic metrics are commonly used to evaluate the model's performance in classification tasks, namely accuracy, precision, recall, and F1-score, which are summarized in Table 6 for the two experimental cases. It can be seen that the accuracy, precision, recall, and F1-score values in the case of one-block retraining are 94.44 %, 94.55 %, 94.44 %, and 94.45 %, respectively, whereas in the case of two-block retraining, the accuracy, precision, recall, and F1-score values are 96.11 %, 96.15 %, 96.11 %, and 96.12 %, respectively. In general, all four metrics—accuracy, precision, recall, and F1-score—in the case of two-block training are larger than those of one-block training. As mentioned in Table 5, the difference between trainable parameters in the two cases is not much, only about 500 thousand parameters. Therefore, we choose the model fine-tuning with two blocks for further experiments.

Fig. 14 illustrates the classification of some samples of the testing set along with their prediction classes that are obtained using the trained MobileNet model. It should be noted that "A" denotes the actual, and "P" denotes the predicted. The model correctly classified most of the samples in the testing set. Some of the correct predictions are shown in Fig. 14a, where each image is a representation of each category. Of the total 180 samples of the testing set, there are seven samples with incorrect predictions (accounting for less than 4 %). The representative samples in Fig. 14b show the failure of the model to correctly classify the samples. It is observed that the model predicted moderate damage in two samples instead of the ground truth, which indicates one sample belongs to the "no" class and the other to the "light" class. Similarly, moderate

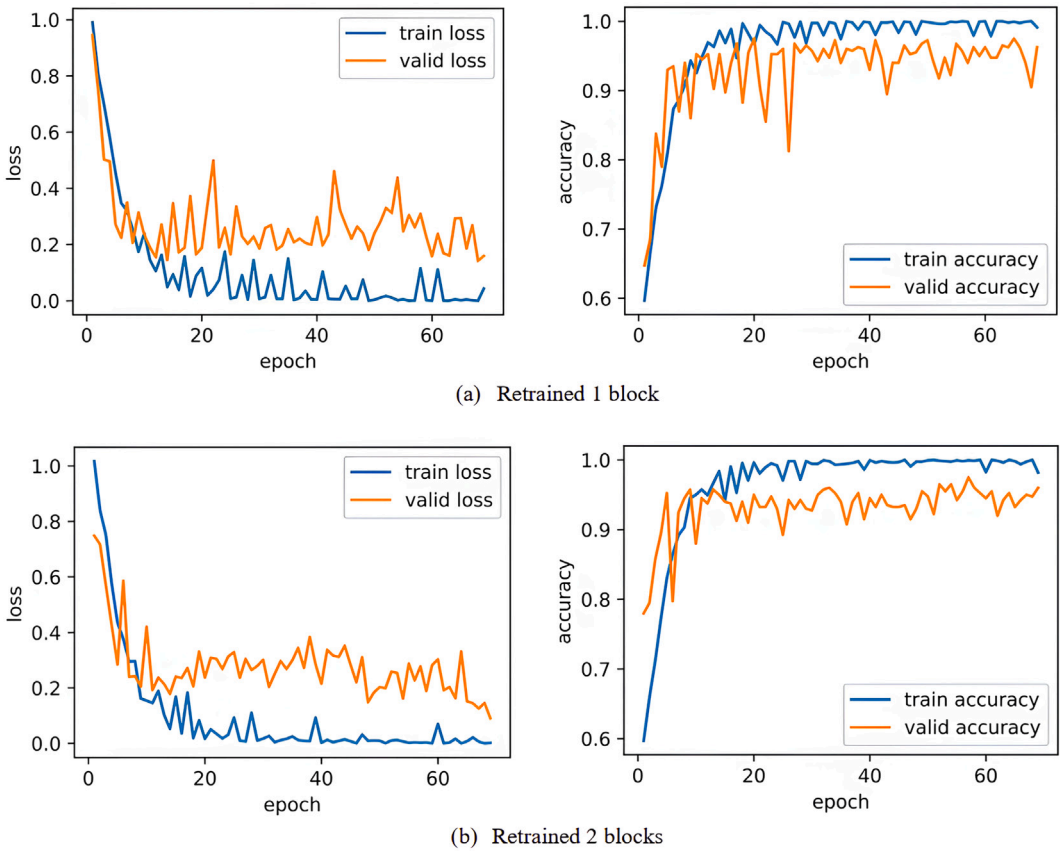


Fig. 11. Loss and accuracy histories of the training and validation datasets: (a) retrained 1 block, (b) retrained 2 blocks.

damage is occasionally misclassified as light damage or severe damage. This misclassification demonstrates that there are some similarities in the classes that confuse the classification model, which is mainly due to the complexity of the damaged structure and the background noise.

To evaluate the performance of the proposed approach, we conducted a comparative analysis of the classification accuracy of our MobileNet model with optimal hyperparameters against several state-of-the-art DL methods recently employed for structural damage classification tasks. Table 7 provides an overview of the DL methods and their classification accuracies from relevant studies. From Table 7, Gao et al. [74] and Perez et al. [75] utilized the pre-trained VGG16 model for damage recognition across various structural components and building defects, achieving accuracies of 89.7 % and 87.5 %, respectively. While effective, these approaches are limited by the relatively shallow architecture of VGG16, which struggles to generalize in scenarios with complex structural damage patterns. Mangalathu et al. [76] employed a long short-term memory (LSTM) model tailored to sequential earthquake-impact classification tasks, obtaining 86 % accuracy, but its reliance on temporal data restricts its applicability in image-only datasets. Advanced CNN-based methods such as ResNet50 [77], Xception [78], and Inception-ResNet-v2 [79] have achieved higher accuracies of 87.47 %, 94.95 %, and 97.41 %, respectively. These models benefit from deeper architectures and innovative design elements like residual connections, which enhance feature extraction capabilities. However, the computational costs associated with such models are significantly higher, potentially limiting their deployment in real-time scenarios. Our proposed MobileNet model, optimized through Bayesian hyperparameter tuning, achieved an accuracy of 96.11 %, demonstrating superior performance compared to most existing studies. This is attributed to its lightweight architecture, which effectively balances computational efficiency and accuracy. Furthermore, the proposed method surpasses other MobileNet-based approaches, such as those by Ogunjinmi et al. [66] (88.3 % accuracy) and Dais et al. [80] (95.3 % accuracy), by leveraging optimized hyperparameters and a robust fine-tuning strategy. Notably, the proposed approach provides a practical solution for real-time assessment of post-earthquake structural damage. The simplicity and speed of the MobileNet architecture, combined with its high classification accuracy, make it ideal for emergency response scenarios. Unlike traditional methods that may require extensive computational resources or manual intervention, the proposed model facilitates automated, accurate, and efficient damage recognition, which is critical for timely decision-making during disasters. The comparative analysis highlights the contribution and innovation of our work in advancing image-based structural damage classification. By addressing challenges related to computational cost, accuracy, and generalization, our method sets a new benchmark for real-time structural damage assessment in practical applications.

The proposed MobileNet model could well be the backbone for object detection and semantic segmentation tasks for post-earthquake structural damage recognition. Although interesting and ultimately valuable, attempting to accomplish all of these within a practical timeframe is unrealistic and goes beyond the intended scope of this paper. The authors believe that a more in-depth exploration of these aspects could be pursued in a separate paper, where they can be thoroughly investigated.

3.3. Visual explanations from a deep network

To visualize and understand the decision-making process of the pre-trained MobileNet model in the context of damage level classification, we use Grad-CAM to generate visual explanations for the MobileNet model's decisions, which makes it more transparent. Grad-CAM visualizations of the model predictions for moderate damage and severe

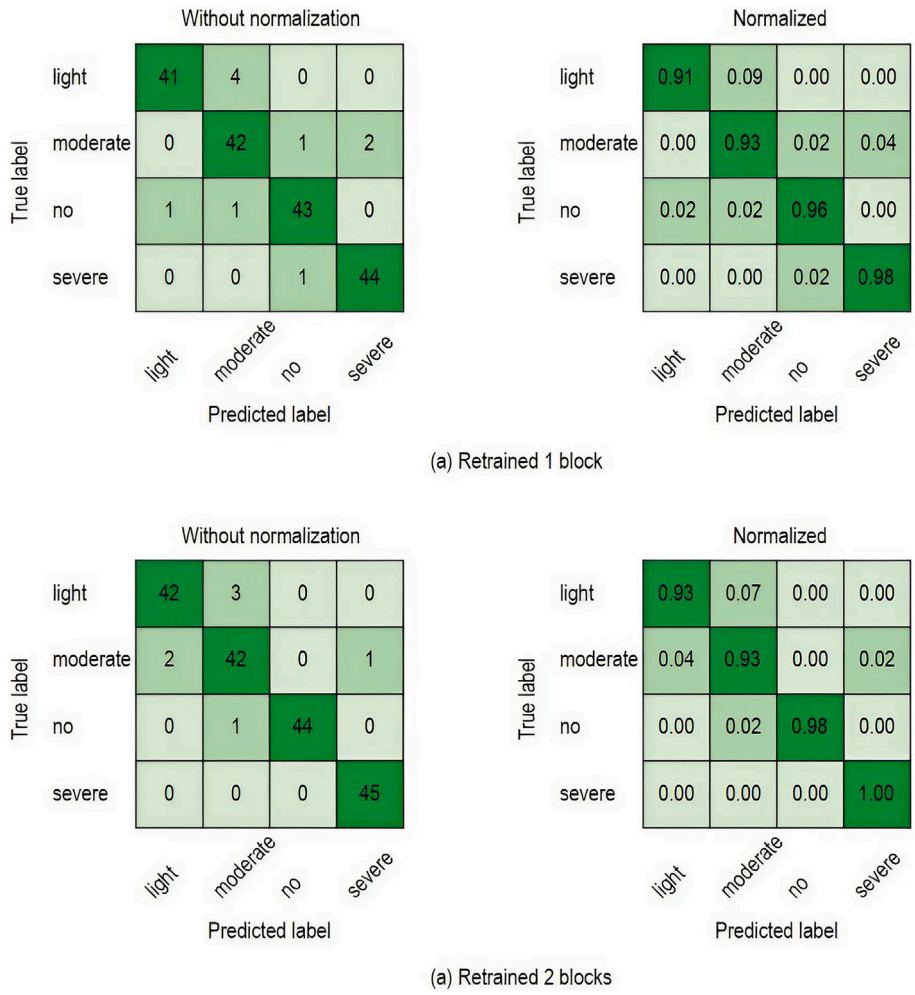


Fig. 12. Confusion matrices of the testing dataset: (a) retrained 1 block, (b) retrained 2 blocks.

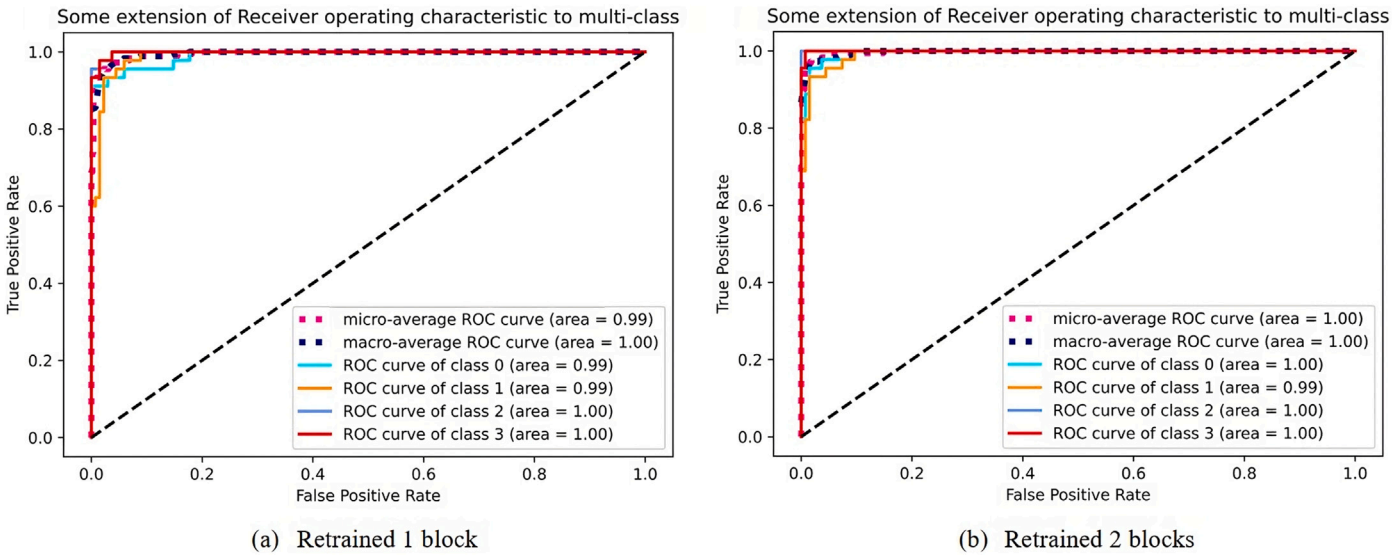


Fig. 13. ROC curves of the testing dataset: (a) retrained 1 block, (b) retrained 2 blocks.

Table 6
Comparison of performance of two models using two different fine-tuning strategies.

Retrained blocks	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
One + FC-layers	94.44	94.55	94.44	94.45
Two + FC-layers	96.11	96.15	96.11	96.12

Fig. 14. Some samples of the testing dataset with predicted cases: (a) correct predictions, (b) incorrect predictions (Note: A is actual, P is predicted).

Table 7
Comparison of the result with previous studies.

Related work	Classification task	Method(s)	Accuracy (%)
Gao et al. [74]	Damage in all structural components	VGG16	89.70
Perez et al. [75]	Damage in all structural components	VGG16	87.50
Mangalathu et al. [76]	Damage in all structural components	LSTM	86.00
Pan et al. [77]	Damage in columns only	ResNet50	87.47
Zhang et al. [57]	Damage in all structural components	Factorization machine + Deep neural network	70.30
Ogunjinmi et al. [66]	Damage in all structural components	MobileNet	88.30
Abubakr et al. [78]	Damage in all structural components	Xception	94.95
Xu et al. [81]	Damage in all structural components	Few-shot meta-learning	93.50
Wang et al. [79]	Damage in track slabs only	Inception-ResNet-v2	97.41
Dais et al. [80]	Damage in masonry surfaces only	MobileNet	95.30
Sirhan et al. [82]	Damage in asphalt surfaces only	ResNet101	87.11
Our work	Damage in all structural components	MobileNet + Bayesian optimization	96.11

damage classes are depicted in Fig. 15. It can be seen that the heat map highlights the damaged regions of the input image, which proves that the model has focused its attention on the damaged regions to make predictions.

For the light damage and no damage classes, the Grad-CAM visualizations of the model predictions are depicted in Fig. 16. Unlike the results obtained from the moderate damage and severe damage classes, where the heat map highlights damaged regions of the input image, here the model has focused its attention on various regions (either damaged or undamaged) to make predictions. This is understandable since light-damage and non-damage regions are easily confused in a dataset with background noise.

3.4. Improvement of the generalizability of the MobileNet model

As seen in Fig. 11, the overfitting phenomenon occurred due to the lack of training data. To improve the generalizability of the proposed MobileNet model, data augmentation techniques such as flipping, rotating, translating, and cropping are used to expand the size of the training set during model training. In this section, we train the proposed MobileNet model on an extended dataset, which combines the dataset described in Table 1 and partial data obtained from the Pacific Earthquake Engineering Research Center (PEER) Hub ImageNet (ϕ-Net) [83]. A total of 3780 damaged images were chosen for usage in this subsection, of which 3600 images were utilized for training and validation,

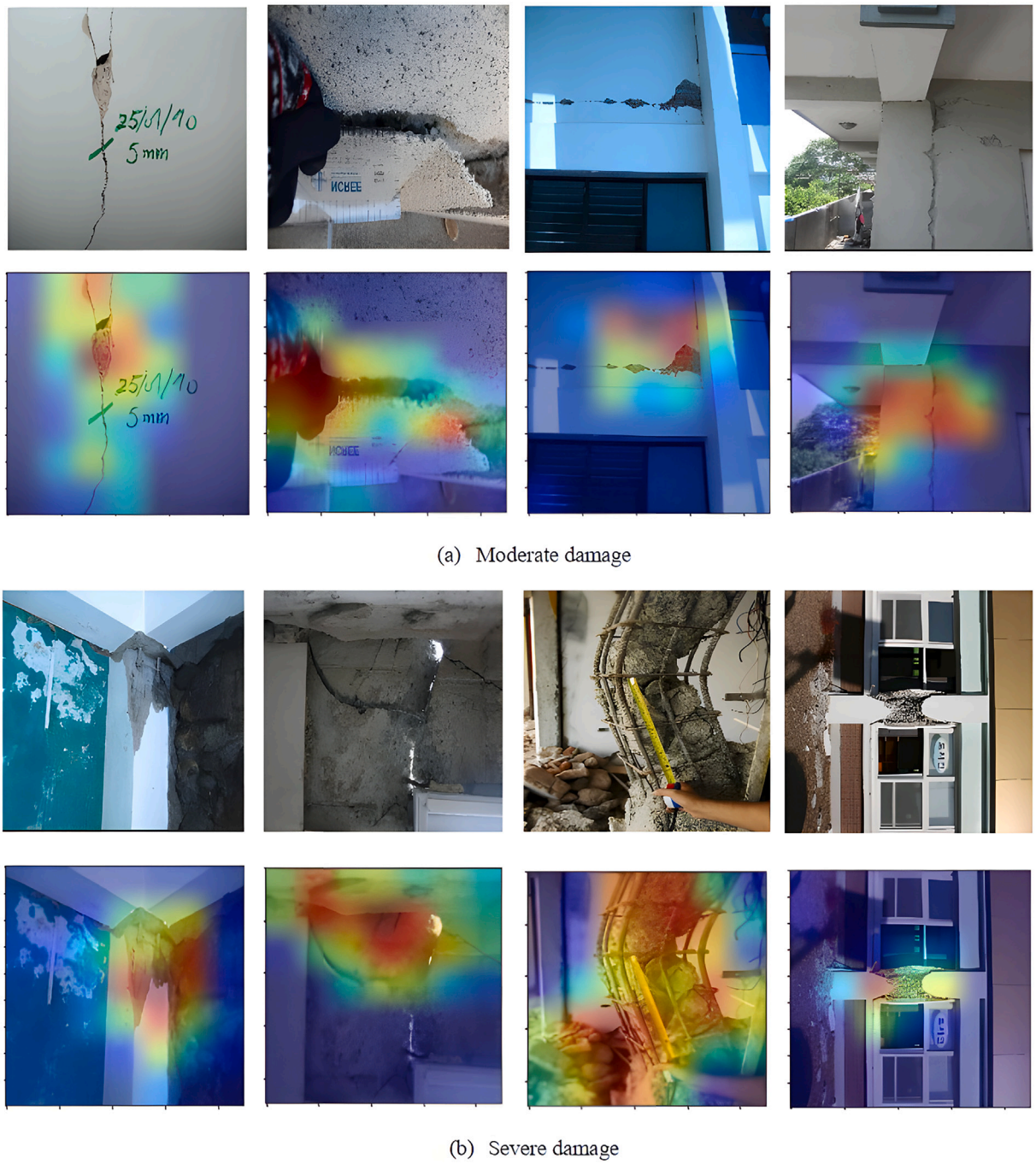
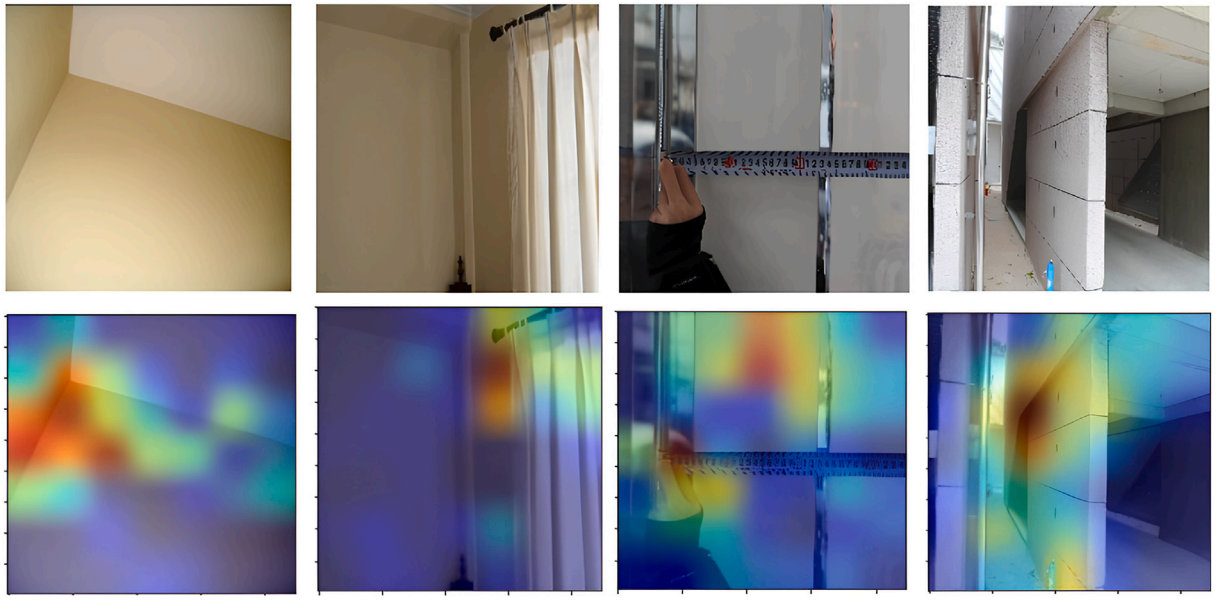


Fig. 15. Grad-CAM visualizations in damage images: (a) moderate damage, (b) severe damage.

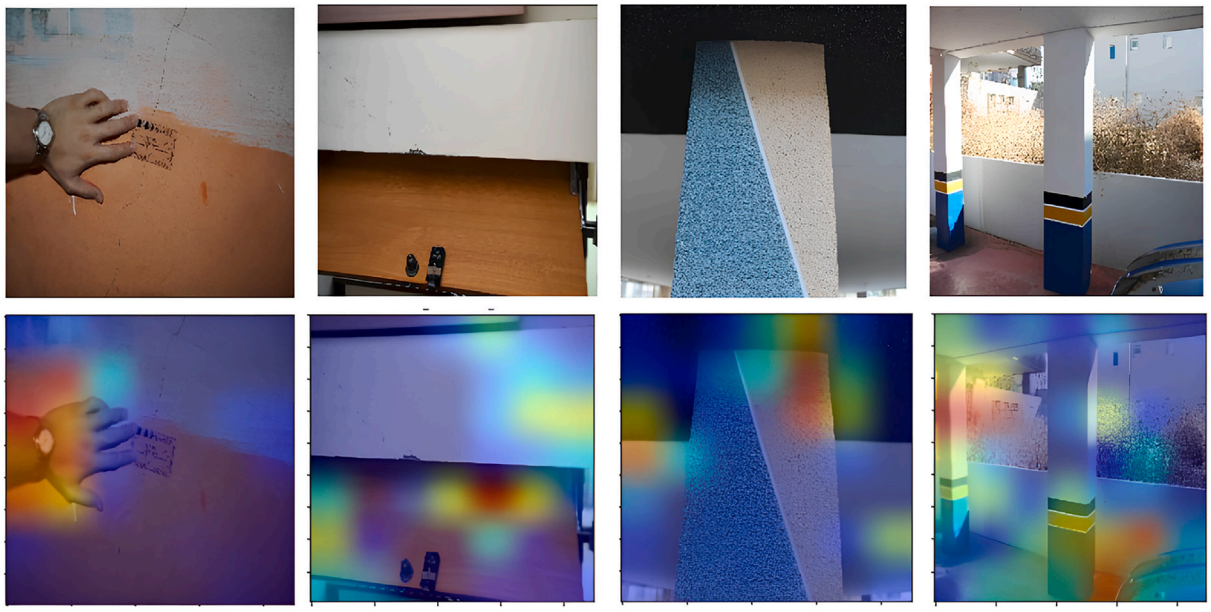
and the remaining 180 images from the Pohang earthquake database were employed in the testing phase to evaluate the generalizability of the trained model. The categories of the used dataset and the distribution of images by category are summarized in Table 8. It is worth noting that the amount of added data not only aims to increase the original dataset but also makes the balanced data categories.

Fine-tuning the MobileNet model on the extended dataset is still performed with 70 epochs. Fig. 17 illustrates the loss and accuracy histories of the training and validation sets during model training. The first observation is that the loss values decrease sharply in the first 15 epochs and then start to converge, which shows that the model is learning something that we would expect. A similar observation is seen in the accuracy values; these values increase rapidly in the early stages and start to converge from the 15th epoch onwards. An important observation is that the training process is stable, and no overfitting occurs during model training (accuracy on the training and validation sets differs by about 2 %). This proves that the model has improved generalizability on the expanded dataset.

The confusion matrices used to evaluate the performance of the MobileNet model on the testing set are shown in Fig. 18. This allows us to visualize the performance of the MobileNet model by comparing the true labels to the predicted labels. It can be seen that the values on the main diagonal are much larger compared to the values on the remaining elements in the confusion matrices. Specifically, the model correctly



(a) No damage



(b) Light damage

Fig. 16. Grad-CAM visualizations in damage images: (a) no damage, (b) light damage.

Table 8
An extended dataset of structural damage is used in this study.

Image	No damage	Light damage	Moderate damage	Severe damage
Training	300 + 600	500 + 400	500 + 400	700 + 200
Testing	45	45	45	45
Total	945	945	945	945

predicted 173 samples out of a total of 180 samples (corresponding to a correct prediction probability of 96.11 %) on the testing set. This proves that the model has good generalizability to the unseen data.

Four basic metrics commonly used to evaluate the model's performance in classification tasks are summarized in Table 9 for the two experimental cases. It can be seen that the accuracy, precision, recall, and F1-score values in the case of using the initial dataset are 96.11 %, 96.15 %, 96.11 %, and 96.12 %, respectively. In the case of using the extended dataset, the accuracy, precision, recall, and F1-score values are 96.11 %, 96.18 %, 96.11 %, and 96.09 %, respectively. All four evaluation metrics in the two training cases are similar, demonstrating that the performance of the MobileNet model is similar when evaluated on the testing set. However, the model trained on the extended dataset shows better stability and generalizability on the validation set. In short, the more data there is, the more stable the training process and the higher the model's generalizability.

MobileNet is known as a small, low-latency, low-power deep neural network [62]. At the same time, it offers the simplicity and high

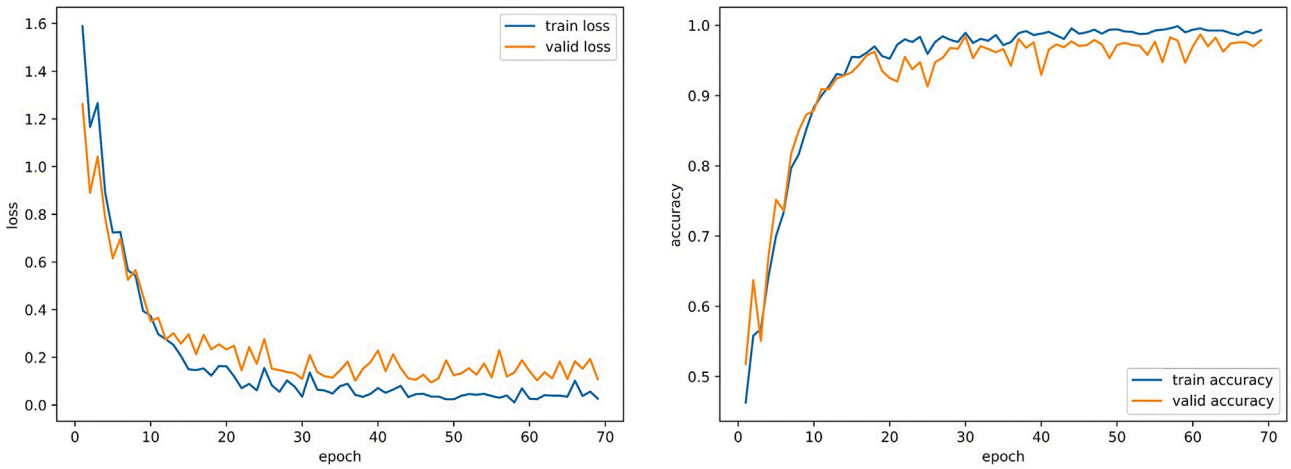


Fig. 17. Loss and accuracy histories of the training and validation on the extended dataset.

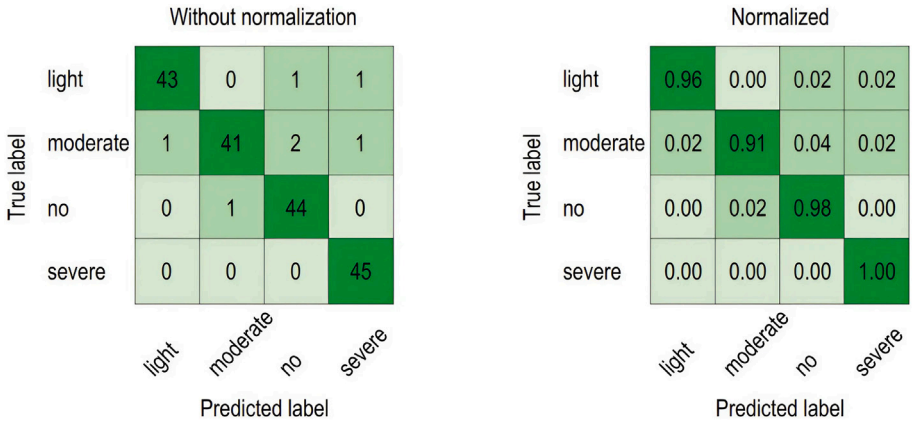


Fig. 18. Confusion matrices of the testing dataset.

Table 9
Comparison of performance of the MobileNet model on two different datasets.

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Initial dataset	96.11	96.15	96.11	96.12
Extended dataset	96.11	96.18	96.11	96.09

accuracy of the DL solution. These characteristics are necessary to implement a user-interactive portable application to facilitate real-time damage assessment. The optimal MobileNet model that is pre-trained from Python will be converted to an open standard file format (JSON) with Tensorflow.js and shared among engineers and scientists in the community for operation and further development. TensorFlow.js is a JavaScript ML library that is used to train and deploy ML models as a web application with an easy-to-use graphical user interface. Scientists and engineers may work across a variety of development platforms, thanks to the standard data interchange format. The optimal DL approach can be adopted as an engineering guideline and preserved publicly for community use in certain engineering issues. Interactive web applications can be further developed to provide fast and accurate DL solutions to users. In this work, the DL solution that allows users to quickly determine the extent of structural damage along with its probability of confidence has been generated and shared through a web-based application, named "Structural damage recognition" [84]. The deployment of the optimal MobileNet model in real-time emergency scenarios is highly feasible due to its ability to perform inference in less than 1 s per image, ensuring near-instantaneous results. This capability is critical in time-sensitive situations, such as post-earthquake damage assessment, where rapid decision-making is paramount. Furthermore, the model provides a confidence score for each prediction, reflecting the certainty of its classification. For predictions with low confidence levels (e.g., below 50 %), users are advised to carefully review the results or consult additional data sources. Additionally, in such cases, users may consider increasing the damage level by one degree as a precautionary measure to ensure safety. It is essential to note that this tool serves as an initial qualitative assessment rather than a comprehensive quantitative analysis. While it offers a fast and reliable method for prioritizing damaged structures and informing immediate actions, detailed evaluations and inspections by experts remain indispensable for final decisions. This integration of speed, simplicity, and transparency makes the proposed system an effective supporting tool in emergency response operations.

In this study, expert knowledge was effectively embedded in the data-driven model through several approaches. First, the data set used was meticulously labeled by domain experts, ensuring that the annotations accurately reflected the characteristics of the level of structural damage, thus embedding critical domain insights into the training data. Second, transfer learning was applied using pre-trained CNN architectures, leveraging general image-processing capabilities from large datasets like ImageNet and adapting them to the specific task of damage level recognition. Third, Bayesian optimization was utilized to fine-tune the hyperparameters, optimizing the model configuration to align with

the unique properties of the dataset. Finally, Grad-CAM was employed to provide visual explanations for the model's predictions, highlighting critical image regions and allowing experts to validate that the model's focus aligned with meaningful structural features. These methods collectively ensured that the model not only performed well but also remained interpretable and aligned with expert understanding, making it a reliable tool for assessing the level of structural damage.

4. Conclusion and future work

In this study, we have demonstrated the feasibility and potential of leveraging DL to automatically recognize post-earthquake structural damage levels from images, offering a rapid and efficient alternative to traditional manual inspections. Specifically, we utilized the pre-trained MobileNet model, originally trained on the ImageNet database, and optimized it using Bayesian techniques to classify four distinct damage levels: no damage, light damage, moderate damage, and severe damage. Experimental results indicate that the proposed MobileNet model achieves a remarkable overall accuracy of 96.11 %, surpassing prior studies in this domain. This high accuracy, combined with the model's lightweight architecture, underscores its practical applicability in time-sensitive scenarios, such as post-earthquake damage assessments. By integrating transfer learning and hyperparameter optimization, the model strikes an effective balance between computational efficiency and classification performance, positioning it as a robust solution for image-based structural damage recognition. However, the study also acknowledges certain limitations. The model primarily focuses on classifying damage severity levels and is not yet capable of identifying specific damage mechanisms, such as flexural or shear failures, which would require additional labeled data and model adjustments. Furthermore, the relatively small size of the initial dataset, while mitigated to some extent through transfer learning and data augmentation, still limits the model's ability to generalize to diverse and unseen earthquake scenarios. Although training on the extended dataset improved robustness, future efforts must focus on further expanding and diversifying the dataset to fully exploit the potential of this approach. Despite these challenges, the findings affirm the value of DL in structural health monitoring, showcasing its promise as a scalable, reliable tool for rapid post-disaster assessments.

In future work, we aim to address these limitations and explore enhancements aligned with the reviewers' feedback. First, we plan to significantly expand the dataset by collecting additional structural damage images to improve diversity and representation across various earthquake scenarios. This will include multi-scale views ranging from localized cracks to entire structures, and collaboration with organizations like PEER will be pursued to access larger and more diverse datasets. Second, we will investigate advanced data augmentation techniques, such as diffusion models, to synthetically generate high-quality images, introducing greater variability and addressing overfitting observed with smaller datasets. Third, embedding contextual information—such as image orientation, structural geometry, and metadata—will be explored to improve the model's ability to assess both damage severity and specific damage mechanisms (e.g., flexural, shear). Integration with physics-informed modeling and digital twin frameworks will also be prioritized to enhance the model's predictive reliability and interoperability. To further refine the classification process, especially for overlapping damage severity levels, we will evaluate hybrid approaches, such as combining CNNs with graph-based models or ensemble methods, to improve decision boundaries and accuracy. Additionally, we will explore advanced network architectures, including multi-scale and hierarchical designs or meta-learning paradigms, to enhance the model's generalizability and adaptability to complex real-world scenarios. Finally, we aim to improve the computational feasibility and environmental sustainability of the model. This includes evaluating energy consumption during training and inference, adopting energy-efficient architectures, and exploring carbon offset measures to reduce the environmental impact. These efforts will ensure that the model remains scalable, robust, and suitable for deployment in real-time emergency scenarios while maintaining low computational costs and supporting sustainable AI practices.

CRediT authorship contribution statement

Xiaoying Zhuang: Writing – review & editing, Writing – original draft, Supervision, Investigation, Funding acquisition, Conceptualization. **Than V. Tran:** Writing – original draft, Software, Investigation. **H. Nguyen-Xuan:** Writing – review & editing, Supervision, Software. **Timon Rabczuk:** Writing – review & editing, Supervision, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge the financial support of the RISE (734370) project.

Data availability

Data will be made available on request.

References

[1] Dong L, Shan J. A comprehensive review of earthquake-induced building damage detection with remote sensing techniques. ISPRS J Photogramm Remote Sens 2013;84:85–99. https://doi.org/10.1016/j.isprsjprs.2013.06.011.
[2] Feng D, Feng MQ. Experimental validation of cost-effective vision-based structural Health monitoring, Mech Syst Signal Process 2017;88:199–211. https://doi.org/10.1016/j.ymssp.2016.11.021.
[3] Yoon H, Elanwar H, Choi H, Golparvar-Fard M, Spencer BF Jr. Target-free approach for vision-based structural system identification using consumer-grade cameras. Struct Control Health Monit 2016;23:1405–16. https://doi.org/10.1002/stc.1850.
[4] Hoskere V, Park J-W, Yoon H, Spencer BF Jr. Vision-based modal survey of civil infrastructure using unmanned aerial vehicles. J Struct Eng 2019;145:04019062. https://doi.org/10.1061/(ASCE)ST.1943-541X.0002321.
[5] Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE; 2009. p. 248–55. https://doi.org/10.1109/CVPR.2009.5206848.
[6] Do DT, Lee J, Nguyen-Xuan H. Fast evaluation of crack growth path using time series forecasting. Eng Fract Mech 2019;218:106567. https://doi.org/10.1016/j.engfracmech.2019.106567.
[7] Zhang R, Chen Z, Chen S, Zheng J, Büyüköztürk O, Sun H. Deep long short-term memory networks for nonlinear structural seismic response prediction. Comput Struct 2019;220:55–68. https://doi.org/10.1016/j.compstruc.2019.05.006.
[8] Chakraborty A, Anitescu C, Zhuang X, Rabczuk T. Domain adaptation based transfer learning approach for solving PDEs on complex geometries. Eng Comput 2022;38:4569–88. https://doi.org/10.1007/s00366-022-01661-2.
[9] Guo H, Zhuang X, Chen P, Alajlan N, Rabczuk T. Stochastic deep collocation method based on neural architecture search and transfer learning for heterogeneous porous media. Eng Comput 2022;38:5173–98. https://doi.org/10.1007/s00366-021-01586-2.
[10] Liao Y, Lin R, Zhang R, Wu G. Attention-based LSTM (AttLSTM) neural network for seismic response modeling of bridges. Comput Struct 2023;275:106915. https://doi.org/10.1016/j.compstruc.2022.106915.
[11] Pham HT, Han S. Natural language processing with multitask classification for semantic prediction of risk-handling actions in construction contracts. J Comput Civ Eng 2023;37:04023027. https://doi.org/10.1061/JCCEE5.CPENG-5218.
[12] Kwasigroch A, Grochowski M, Mikołajczyk A. Neural architecture search for skin lesion classification. IEEE Access 2020;8:9061–71. https://doi.org/10.1109/ACCESS.2020.2964424.
[13] Dinh VQ, Munir F, Azam S, Yow K-C, Jeon M. Transfer learning for vehicle detection using two cameras with different focal lengths. Inf Sci (NY) 2020;514:71–87. https://doi.org/10.1016/j.ins.2019.11.034.
[14] Ho TT, Kim T, Kim WJ, Lee CH, Chae KJ, Bak SH, et al. A 3D-CNN model with CT-based parametric response mapping for classifying COPD subjects. Sci Rep 2021;11:34. https://doi.org/10.1038/s41598-020-79336-5.
[15] Wang Q, Zhuang X. A CNN-based surrogate model of isogeometric analysis in nonlocal flexoelectric problems. Eng Comput 2023;39:943–58. https://doi.org/10.1007/s00366-022-01717-3.
[16] Ho TT, Kim G-T, Kim T, Choi S, Park E-K. Classification of rotator cuff tears in ultrasound images using deep learning models. Med Biol Eng Comput 2022;60:1269–78. https://doi.org/10.1007/s11517-022-02502-6.

[17] Nguyen PD, Nguyen TQ, Tao Q, Vogel F, Nguyen-Xuan H. A data-driven machine learning approach for the 3D printing process optimisation. Virtual Phys Prototyp 2022;17:768–86. https://doi.org/10.1080/17452759.2022.2068446.

[18] Moon YS, Park B, Park J, Ho TT, Lim J-K, Choi S. Identification and risk classification of thymic epithelial tumors using 3D computed tomography images and deep learning models. Biomed Signal Process Control 2024;95:106473. https://doi.org/10.1016/j.bspc.2024.106473.

[19] Cha Y-J, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. Comput-Aided CivAnd Infrastruct Eng 2017;32:361–78. https://doi.org/10.1111/mice.12263.

[20] Park S-S, Tran V-T, Lee D-E. Application of various YOLO models for computer vision-based real-time pothole detection. Appl Sci 2021;11:11229. https://doi.org/10.3390/app112311229.

[21] Liang X. Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization. Comput-Aided Civ Infrastruct Eng 2019;34:415–30. https://doi.org/10.1111/mice.12425.

[22] 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;27:e2507. https://doi.org/10.1002/stc.2507.

[23] Kim B, Yuvaraj N, Park HW, Preethaa KS, Pandian RA, Lee D-E. Investigation of steel frame damage based on computer vision and deep learning. Autom Construct 2021;132:103941. https://doi.org/10.1016/j.autcon.2021.103941.

[24] Chen J, Yang T, Zhang D, Huang H, Tian Y. Deep learning based classification of rock structure of tunnel face. Geosci Front 2021;12:395–404. https://doi.org/10.1016/j.gsf.2020.04.003.

[25] Rosso MM, Marasco G, Aiello S, Aloisio A, Chiaia B, Marano GC. Convolutional networks and transformers for intelligent road tunnel investigations. Comput Struct 2023;275:106918. https://doi.org/10.1016/j.compstruc.2022.106918.

[26] Ni P, Li Y, Sun L, Wang A. Traffic-induced bridge displacement reconstruction using a physics-informed convolutional neural network. Comput Struct 2022;271:106863. https://doi.org/10.1016/j.compstruc.2022.106863.

[27] Xu Y, Qiao W, Zhao J, Zhang Q, Li H. Vision-based multi-level synthetical evaluation of seismic damage for RC structural components: a multi-task learning approach. Earthq Eng Eng Vib 2023;22:69–85. https://doi.org/10.1007/s11803-023-2153-4.

[28] Kim B, Cho S. Automated multiple concrete damage detection using instance segmentation deep learning model. Appl Sci 2020;10:8008. https://doi.org/10.3390/app10228008.

[29] Kumar P, Batchu S, Kota SR. Real-time concrete damage detection using deep learning for high rise structures. IEEE Access 2021;9:112312–31. https://doi.org/10.1109/ACCESS.2021.3102647.

[30] Zhu J, Zhong J, Ma T, Huang X, Zhang W, Zhou Y. Pavement distress detection using convolutional neural networks with images captured via UAV. Autom Construct 2022;133:103991. https://doi.org/10.1016/j.autcon.2021.103991.

[31] Guan J, Yang X, Ding L, Cheng X, Lee VC, Jin C. Automated pixel-level pavement distress detection based on stereo vision and deep learning. Autom Construct 2021;129:103788. https://doi.org/10.1016/j.autcon.2021.103788.

[32] Liu F, Liu J, Wang L. Asphalt pavement crack detection based on convolutional neural network and infrared thermography. IEEE Trans Intell Transp Syst 2022;23:22145–55. https://doi.org/10.1109/TITS.2022.3142393.

[33] Huyan J, Li W, Tighe S, Xu Z, Zhai J. CrackU-net: a novel deep convolutional neural network for pixelwise pavement crack detection. Struct Control Health Monit 2020;27:e2551. https://doi.org/10.1002/stc.2551.

[34] Liu J, Yang X, Lau S, Wang X, Luo S, Lee VC-S, et al. Automated pavement crack detection and segmentation based on two-step convolutional neural network. Comput-Aided Civ Infrastruct Eng 2020;35:1291–305. https://doi.org/10.1111/mice.12622.

[35] Xu Y, Fan Y, Bao Y, Li H. Task-aware meta-learning paradigm for universal structural damage segmentation using limited images. Eng Struct 2023;284:115917. https://doi.org/10.1016/j.engstruct.2023.115917.

[36] Liu F, Liu J, Wang L. Deep learning and infrared thermography for asphalt pavement crack severity classification. Autom Construct 2022;140:104383. https://doi.org/10.1016/j.autcon.2022.104383.

[37] Tan H, Zheng L, Ma C, Xu Y, Sun Y. Deep learning-assisted high-resolution sonar detection of local damage in underwater structures. Autom Construct 2024;164:105479. https://doi.org/10.1016/j.autcon.2024.105479.

[38] Yang X, Del Rey Castillo E, Zou Y, Wotherspoon L. UAV-deployed deep learning network for real-time multi-class damage detection using model quantization techniques. Autom Construct 2024;159:105254. https://doi.org/10.1016/j.autcon.2023.105254.

[39] Agyemang IO, Zhang X, Adjei-Mensah I, Acheampong D, Fiasam LD, Sey C, et al. Automated vision-based structural health inspection and assessment for post-construction civil infrastructure. Autom Construct 2023;156:105153. https://doi.org/10.1016/j.autcon.2023.105153.

[40] Xu Y, Fan Y, Li H. Lightweight semantic segmentation of complex structural damage recognition for actual bridges. Struct Health Monit 2023;22:3250–69. https://doi.org/10.1177/14759217221147015.

[41] Liu H, Zhang Y. Image-driven structural steel damage condition assessment method using deep learning algorithm. Measurement 2019;133:168–81. https://doi.org/10.1016/j.measurement.2018.09.081.

[42] Zhang Y, Sun X, Loh KJ, Su W, Xue Z, Zhao X. Autonomous bolt loosening detection using deep learning. Struct Health Monit 2020;19:105–22. https://doi.org/10.1177/1475921719837509.

[43] Naito S, Tomozawa H, Mori Y, Nagata T, Monma N, Nakamura H, et al. Building-damage detection method based on machine learning utilizing aerial photographs of the Kumamoto earthquake. Earthq Spectra 2020;36:1166–87. https://doi.org/10.1177/8755293019901309.

[44] Li X, Caragea D, Zhang H, Imran M. Localizing and quantifying infrastructure damage using class activation mapping approaches, Soc Netw Anal Min 2019;9:1–15. https://doi.org/10.1007/s13278-019-0588-4.

[45] Cheng C-S, Behzadan AH, Noshadravan A. Deep learning for post-hurricane aerial damage assessment of buildings. Comput-Aided Civ Infrastruct Eng 2021;36:695–710. https://doi.org/10.1111/mice.12658.

[46] Hoskere V, Narazaki Y, Spencer BF Jr. Physics-based graphics models in 3D synthetic environments as autonomous vision-based inspection testbeds. Sensors (Switzerland) 2022;22:532. https://doi.org/10.3390/s22020532.

[47] Jiang Y, Pang D, Li C. A deep learning approach for fast detection and classification of concrete damage. Autom Construct 2021;128:103785. https://doi.org/10.1016/j.autcon.2021.103785.

[48] Du Nguyen Q, Thai H-T. Crack segmentation of imbalanced data: the role of loss functions. Eng Struct 2023;297:116988. https://doi.org/10.1016/j.engstruct.2023.116988.

[49] Tran TV, Nguyen-Xuan H, Zhuang X. Investigation of crack segmentation and fast evaluation of crack propagation, based on deep learning, Front Struct Civ Eng 2024;1–20. https://doi.org/10.1007/s11709-024-1040-z.

[50] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998;86:2278–324. https://doi.org/10.1109/5.726791.

[51] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng 2009;22:1345–59. https://doi.org/10.1109/TKDE.2009.191.

[52] Rodriguez JD, Perez A, Lozano JA. Sensitivity analysis of k-fold cross validation in prediction error estimation. IEEE Trans Pattern Anal Mach Intell 2009;32:569–75. https://doi.org/10.1109/TPAMI.2009.187.

[53] Brochu E, Cora VM, De Freitas N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning; 2010. arXiv preprint 1012.2599. https://doi.org/10.48550/arXiv.1012.2599.

[54] Frazier PI. A tutorial on Bayesian optimization; 2018. arXiv preprint 1807.02811. https://doi.org/10.48550/arXiv.1807.02811.

[55] Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 618–26. https://doi.org/10.1109/ICCV.2017.74.

[56] Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 2921–9. https://doi.org/10.1109/CVPR.2016.319.

[57] Zhang L, Pan Y. Information fusion for automated post-disaster building damage evaluation using deep neural network. Sustain Cities Soc 2022;77:103574. https://doi.org/10.1016/j.scs.2021.103574.

[58] Hanley JA, McNeil BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 1982;143:29–36. https://doi.org/10.1148/radiology.143.1.7063747.

[59] Chollet F. Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 1251–8. https://doi.org/10.48550/arXiv.1610.02357.

[60] 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. p. 2818–26. https://doi.org/10.48550/arXiv.1512.00567.

[61] Szegedy C, Ioffe S, Vanhoucke V, Alemi A. Inception-v4, inception-ResNet and the impact of residual connections on learning. In: Proceedings of the AAAI conference on artificial intelligence, vol. 31; 2017. p. 4278–84. https://doi.org/10.1609/aaai.v31i1.11231.

[62] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. MobileNets: efficient convolutional neural networks for Mobile vision applications; 2017. arXiv preprint 1704.04861. https://doi.org/10.48550/arXiv.1704.04861.

[63] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C. MobileNetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 4510–20. https://doi.org/10.1109/CVPR.2018.00474.

[64] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 2261–9. https://doi.org/10.1109/CVPR.2017.243.

[65] Zoph B, Vasudevan V, Shlens J, Le QV. Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 8697–710. https://doi.org/10.1109/CVPR.2018.00907.

[66] Ogunjinmi PD, Park S-S, Kim B, Lee D-E. Rapid post-earthquake structural damage assessment using convolutional neural networks and transfer learning. Sensors (Switzerland) 2022;22:3471. https://doi.org/10.3390/s22093471.

[67] Verma H, Siruvuri SV, Budarapu P. A machine learning-based image classification of silicon solar cells. Int J Hydromechatron 2024;7:49–66. https://doi.org/10.1504/IJHM.2024.135990.

[68] Siruvuri SV, Budarapu P, Paggi M. Influence of cracks on fracture strength and electric power losses in silicon solar cells at high temperatures: deep machine learning and molecular dynamics approach. Appl Phys A 2023;129:408. https://doi.org/10.1007/s00339-023-06629-7.

[69] Mikołajczyk A, Grochowski M. Data augmentation for improving deep learning in image classification problem. In: 2018 international interdisciplinary PhD workshop (IIPhDW). IEEE; 2018. p. 117–22. https://doi.org/10.1109/IIPHDW.2018.8388338.

[70] Kingma DP, Ba J. Adam: a method for stochastic optimization; 2014. arXiv preprint 1412.6980. https://doi.org/10.48550/arXiv.1412.6980.

[71] Clevert D-A, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (ELUs); 2015. arXiv preprint 1511.07289. https://doi.org/10.48550/arXiv.1511.07289.
[72] Agarap AF. Deep learning using rectified linear units (ReLU); 2018. arXiv preprint 1803.08375. https://doi.org/10.48550/arXiv.1803.08375.
[73] Ruder S. An overview of gradient descent optimization algorithms; 2016. arXiv preprint 1609.04747. https://doi.org/10.48550/arXiv.1609.04747.
[74] Gao Y, Mosalam KM. Deep transfer learning for image-based structural damage recognition. Comput-Aided Civ Infrastruct Eng 2018;33:748–68. https://doi.org/10.1111/mice.12363.
[75] Perez H, Tah JH, Mosavi A. Deep learning for detecting building defects using convolutional neural networks. Sensors (Switzerland) 2019;19:3556. https://doi.org/10.3390/s19163556.
[76] Mangalathu S, Burton HV. Deep learning-based classification of earthquake-impacted buildings using textual damage descriptions. Int J Disaster Risk Reduct 2019;36:101111. https://doi.org/10.1016/j.ijdrr.2019.101111.
[77] Pan X, Yang T. Postdisaster image-based damage detection and repair cost estimation of reinforced concrete buildings using dual convolutional neural networks. Comput-Aided Civ Infrastruct Eng 2020;35:495–510. https://doi.org/10.1111/mice.12549.
[78] Abubakr M, Rady M, Badran K, Mahfouz SY. Application of deep learning in damage classification of reinforced concrete bridges. Ain Shams Eng J 2024;15:102297. https://doi.org/10.1016/j.asej.2023.102297.
[79] Wang W, Hu W, Wang W, Xu X, Wang M, Shi Y, et al. Automated crack severity level detection and classification for ballastless track slab using deep convolutional neural network. Autom Construct 2021;124:103484. https://doi.org/10.1016/j.autcon.2020.103484.
[80] Dais D, Bal IE, Smyrou E, Sarhosis V. Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. Autom Construct 2021;125:103606. https://doi.org/10.1016/j.autcon.2021.103606.
[81] Xu Y, Bao Y, Zhang Y, Li H. Attribute-based structural damage identification by few-shot meta learning with inter-class knowledge transfer. Struct Health Monit 2021;20:1494–517. https://doi.org/10.1177/1475921720921135.
[82] Sirhan M, Bekhor S, Sidess A. Multilabel CNN model for asphalt distress classification. J Comput Civ Eng 2024;38:04023040. https://doi.org/10.1061/JCCEE5.CPENG-5500.
[83] Gao Y, Mosalam KM. PEER Hub ImageNet: a large-scale multiattribute benchmark data set of structural images. J Struct Eng 2020;146:04020198. https://doi.org/10.1061/(ASCE)ST.1943-541X.0002745.
[84] Tran TV. Structural damage recognition; 2024. https://vanthantran.github.io/than.github.io/. [Accessed 30 December 2024].