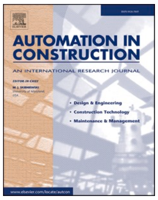
Automation in Construction 132 (2021) 103941

Contents lists available at ScienceDirect

Automation in Construction

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



Investigation of steel frame damage based on computer vision and deep learning

Bubryur Kim [a], N. Yuvaraj [b], Hee Won Park [c], K.R. Sri Preethaa [d], R. Arun Pandian [e], Dong-Eun Lee [c,f,*]

[a] *KyungPook National Univ., Department of Robot and Smart System Engineering, DaeGu, Republic of Korea.*
[b] *Dong-A Univ., Department of Architectural Engineering, Busan, Republic of Korea*
[c] *KyungPook National Univ., School of Architecture, Civil, Environment and Energy Engineering, DaeGu, Republic of Korea*
[d] *KPR Institute of Engineering and Technology, Dept. of Artificial Intelligence and Data Science, Coimbatore, India*
[e] *KPR Institute of Engineering and Technology, Artificial Intelligence Laboratory, Coimbatore, India*
[f] *KyungPook National Univ., School of Architecture, Civil, Environment and Energy Engineering, 1370. Sangyegk-Dong, Buk-Gu, DaeGu 702-701, Republic of Korea*

ARTICLE INFO

Keywords:
Steel frame damage
Deep learning
Computer vision
Deep convolutional neural network
Steel structure monitoring

ABSTRACT

Visual damage inspection of steel frames by eyes alone is time-consuming and cumbersome; therefore, it produces inconsistent results. Existing computer vision-based methods for inspecting civil structures using deep learning algorithms have not reached full maturity in exactly locating the damage. This paper presents a deep convolutional neural network-based damage locating (DCNN-DL) method that classifies the steel frame images provided as inputs as damaged and undamaged. DenseNet, a DCNN architecture, was trained to classify the damage. The DenseNet output was upscaled and superimposed on the original image to locate the damaged part of the steel frame. The DCNN-DL method was validated using 144 training and 114 validation sets of steel frame images. DenseNet, with an accuracy of 99.3%, outperformed MobileNet and ResNet with accuracies of 96.2% and 95.4%, respectively. This case study confirms that the DCNN-DL method effectively facilitates the real-time inspection and location of steel frame damage.

1. Introduction

Many civil structures (such as bridges, buildings, dams, and highways) in use today were built decades ago. These structures eventually approach the end of their service life and undergo aging and deterioration. The American Society of Civil Engineers claims that approximately 56,000 bridges require rehabilitation owing to their structural deficiencies in USA [1]. The periodic monitoring of critical structures plays a vital role in ensuring safety. Structural health monitoring (SHM) is the process of implementing strategies and systems for structural damage detection [2]. SHM safeguards the operational safety of civil structures through various types of sensor deployment, periodic structural inspection, assessment, and maintenance [3] and includes the periodic continuous monitoring of structural parameters and analysis of data to determine the health, performance, and integrity of the structure [4]. SHM provides critical information to increase the safety of civil structures and infrastructures, thereby supporting informed maintenance decisions and actions [5].

Steel frames are used to construct civil structures with various forms of structural members and are either blended with steel or made entirely of steel [6]. Assuring the quality of streel frames is closely associated with ensuring the safety and longevity of structures. Steel surface defects that may be attributed to various factors, such as external strain, climatic changes, and chemical reactions, degrade the appearance, performance, and durability of steel [7]. Steel products are available in various forms, among which slabs, plates, and hot/cold strips are uniformly surfaced steel products, and rods/bards, angles, and channels are heavy structural surfaces. Various types of surface defects may occur in these products. However, no standards have been established to classify these defects [8]. The continuous external load applied to the steel member causes fatigue and leads to structural damage. Continuous external or cyclic loads applied on a steel member cause fatigue or cracks, which eventually results in structural damage, devastating events, and loss of lives and property [9,10].

Existing studies related to steel surface inspections are mainly confined to either the industry associated with manufacturing steel

[*] Corresponding author.
E-mail address: dolee@knu.ac.kr (D.-E. Lee).

https://doi.org/10.1016/j.autcon.2021.103941
Received 11 May 2021; Received in revised form 30 August 2021; Accepted 1 September 2021
Available online 9 September 2021

members or academia. Stress spectra are generally obtained through theoretical stress analyses based on structural and traffic-loading models. However, fatigue life has been predicted inaccurately owing to errors in both structural and loading modeling [11]. Advances in sensing, computing, data acquisition, and communication have promoted the development of SHM technology in steel bridges. However, limitations in deploying sensors, specifically for in-situ conditions in places where fatigue cracks are expected, hinder safety and pose difficulties in monitoring these structures [12]. Various manual inspection methods that are labor-intensive, time-consuming, and cumbersome are employed to assess the conformity of building plumbing systems and steel-blended (or enabled) structural components [13]. When a repetitive linear mega project (for example, high-rise buildings) is underway, these methods are particularly error-prone because of the tedious data collection, processing, and reporting tasks. Indeed, the repetition of production units exacerbates error probability, thereby causing a mishandling of the nonconformity (i.e., defects or deformations) [14].

The manual detection of structural damage remains a challenging and time-consuming process. Various SHM technologies have been proposed to evaluate the integrity of structures [15–17]. A sensor-based SHM is capable of monitoring structural and temperature-, humidity-, and wind-induced deformations [18]. Non-destructive technologies that use ultrasonic waves and acoustic emission techniques have been applied for damage detection [19–21]. Several SHMs have been utilized to identify damage that occur on the surfaces of large-scale steel structures. Sensor-based technologies depend on implementing numerous sensors and involve complex data processing mechanisms [22,23]. Therefore, it is necessary to develop an efficient method to process data collected from multiple sources. Machine learning (ML) models enable learning from data and are capable of predicting structural damage from the learned knowledge [24]. A support vector machine (SVM)-based kernel function has been used for structural damage detection [25]. In addition, artificial neural networks (ANNs) have been widely applied for damage detection in trusses, steel frames, and steel bridges [26–28]. Detecting damage from structural images using ML models involves complex feature engineering tasks, as it depends on a large number of features [29,30].

Computer vision techniques have been developed to address these challenges. Mobile manipulator imaging systems [31], hybrid image segmentation methods [32,33], and edge detection methods [34,35] are popular techniques for damage detection. Computer vision technologies have been used to identify spatiotemporal vehicle load distribution patterns in long-span steel structures [36]. For example, the image enhancement technique combined with the principal component analysis (PCA) algorithm has been used to identify damage in cable-stayed steel structures [37]. However, computer vision techniques have not reached full maturity in identifying damage in small-scale steel structures, as utilizing sensors to identify damage on such structures remains a challenge. This study fills this research gap by developing a novel computer vision model to identify damage to small-scale steel structures.

2. Current state of damage detection methods based on computer vision

2.1. Literature review

Developments in optics and computer vision have enabled structural inspections over long noncontact distances based on high-precision structural monitoring [38]. Computer vision integrated with deep learning (DL), which detects nonconformity, has enhanced the automation of quality checking and has attracted intensive attention owing to the increasing demand for surface quality assurance [39]. Detection methods may include stress/strain monitoring, structural displacement measurement, crack or damage inspection, and classification [40]. The horizontal and vertical displacements of structural components have been identified based on various image processing algorithms (such as CamShift tracking [41], mean-shift tracking [42], digital image correlation [43], and the Lucas–Kanade method [44]).

Unshaped or deformed object detection and classification is one of the most difficult tasks in computer vision and pattern recognition [45]. The detection of nonconformities on steel surfaces is a type of unshaped object detection problem because a region with defects cannot be predefined. Various methods for detecting and classifying steel frame surface defects have been proposed [46]. They include image processing techniques (or algorithms) that identify surface defects based on their morphology and features (i.e., color, texture, and shape) by investigating the outer surfaces of steel structures [47], classifying the deformed and nondeformed surfaces using texture descriptors obtained based on a gray-level co-occurrence matrix [48], and classifying the defects by hybridizing image processing algorithms and self-organization maps [45]. In addition, neural networks are used to classify steel structures and identify the deformation of steel structures [46]. Genetic algorithms, multivariate discriminant functions, and support vector machines are conventional methods that detect and classify whether the defects or deformations on steel frame surfaces undergo a multiparadigm shift that hybridizes computational intelligence (i.e., genetic algorithms, multivariate discriminant function, support vector machines, ANNs, Gabor filters, entropy filters, and deep neural networks) with computer vision [49–58].

Conventional computer vision techniques have been augmented through integrations with deep neural networks to detect deformations and defects [59]. Accordingly, several DCNN architectures (such as AlexNet [60], VGG16 [61], and Resnet50 [62]) have been trained using a large image dataset [63]. These architectures achieve an acceptable accuracy for general image classification because the performance of the existing DCNN model depends on the quality and quantity of the input data used for training [64]. However, not all DCNN architectures effectively perform tasks pertaining to the handling of real-world cases in the domain of computer vision. Insufficient image data is a major issue in identifying nonconformities (i.e., damage, defect, or deformation) in steel frames [65]. Given insufficient data, a DCNN model that classifies nonconformities encounters a vanishing gradient or overfitting. Therefore, it would be beneficial to choose the best DCNN architecture operable with a minimum number of parameters and optimum trade-offs [66]. Existing studies investigated the competence of the DCNN architecture by training models using a small amount of data; this is a significant drawback of existing studies. In addition, the model may be trained on irrelevant features of an image, rather than on targeted features. This inappropriate processing adds more challenges to classifying a nonconformity (or damage) on the surface of steel frames. Indeed, data augmentation methods may facilitate the development of an image classification model that achieves the objective using a minimum amount of data [56].

Existing DCNN studies propose damage detection methods based on bounding boxes and semantic segmentation [67]. Damage to steel members causes shape deformation and/or changes in dimensions. Because the image of the shape-deformed spot can be captured from various angles and distances, identical shape deformations may have different appearances in the images obtained [68]. The bounding box method can localize the deformation and damage to steel frames. However, it cannot exactly locate damaged spots. The enhanced performance of DCNNs encourages researchers to utilize the DCNN architecture to conduct semantic segmentation [69]. This facilitates the implementation of fully convolutional networks (FCNs) for semantic segmentation. The FCN that was used to detect roads in self-driving vehicles and aerial imagery was used to detect the damage to concrete structures based on semantic segmentation [70–72]. The semantic segmentation method outperformed the mapping of coarse and fine interferences. However, it cannot appropriately detect the deformation of structural members.

In the literature, DL models are configured for various computer

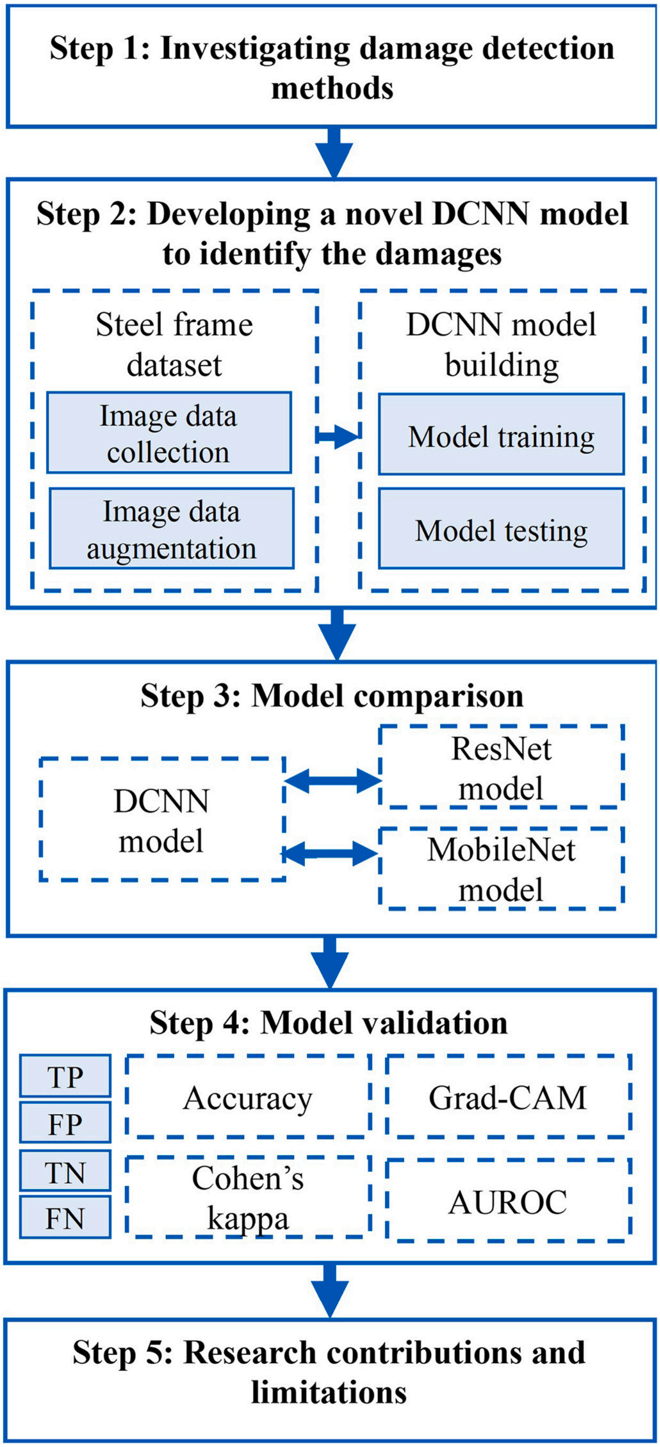


Fig. 1. Research method.

vision-based applications [73]. Among these, convolutional neural network-(CNN) based models have been used as a standard in computer vision for decades. Over the course of development, CNN model architectures have acquired increasingly more hidden layers and large parameter sets, thus providing incredibly accurate models [74,75]. Along with this development, there arises a data bias issue in which the model might learn the complexity in the training images, resulting in overfitting. The greater the presence of functions in the CNN, the greater the chances of overfitting [76–78]. Residual blocks, batch normalization, and skip connections have been proposed to address the issue and retain deep CNN architectures. Residual blocks are effective in handling overfitting issues. However, adding skip-level connections and batch normalization may further increase the complexity of the architecture.

Steel bar damage detection involves training the model on adversarial images. However, configuring a robust model for these images is difficult [79]. The competence of DL models is heavily reliant on the data with which the model is trained. Hence, the quality of data used for building the DL model plays a crucial role in determining its efficiency. It is challenging to build a distinct dataset of small-scale steel structures by considering all essential parameters. It is noteworthy that DL models can apply the advantages of data augmentation techniques to increase the diversity of the training data [80]. In addition, DL algorithms are black boxes and have low interpretability. The black box nature restricts the visualization of the activated perceptron of the deployed DL algorithm [81]. However, gradient-weighted class activation mapping (Grad-CAM) visualization techniques may be employed to visualize the activated perceptron to detect the exact damaged location on the steel frames.

Compared with various pretrained DCNN models, the DenseNet, ResNet, and MobileNet architectures are accepted as the most robust because of their distinctive architectural features [69,70]. DenseNet is outstanding for building an image classification model when the amount of image data is limited [71,81]. The number of parameters to be trained by DenseNet is smaller than that of the other pretrained DCNN models [82,83]. DenseNet enables innovation to avoid pitfalls, such as vanishing gradients and overfitting [84]. Grad-CAM has been used to overcome the limitations of damage detection methods based on bounding boxes and semantic segmentation [85]. Grad-CAM can make a particular decision of interest by using the gradient feature vector that is input into the last convolutional layer of the DCNN. Two distinct images can be classified based on the Grad-CAM visualization [86]. In addition, Grad-CAM visualization does not require a modification to the existing CNN architecture, unlike other object detection models. An object detection model usually requires the localization and annotation of each object to command an additional task. Grad-CAM is a class activation map in which visualization is based on the gradients generated during back propagation. Grad-CAM is the best technique for understanding the intuition of the developed model during the training process. It can also be used as a class activation map for object detection models without modifying the layer stack of the core architecture.

2.2. Research method

The flow of the proposed research method is illustrated in Fig. 1. The entire workflow is organized into five steps, and each step operates on a set of criteria.

First, the performances of the existing damage detection methods were investigated through a literature survey to identify the major limitations of these methods and, therefore, to recapitulate their distinctive research contributions. Second, a novel DCNN model was developed to identify damage on steel frames. A benchmark dataset consisting of a collection of steel frame images was developed by taking different parameters into consideration. The diversity of the dataset was further enhanced by applying the data augmentation techniques. The proposed DCNN model was trained and tested over the augmented dataset.

Third, the novel method was evaluated using a dataset, to compare operations with other existing DL models for image classification. The best performing image classification models in the literature, namely ResNet and MobileNet, were implemented with the steel frame dataset to identify the damage. The performances of the models were compared with that of the proposed DCNN architecture. Fourth, the learning rates of the DL models were measured during the model training and testing phases. The performance of the model was validated by analyzing additional performance metrics, namely accuracy, and Cohen's Kappa and area under the receiver operating characteristics (AUROC) scores. Further, the Grad-CAM visualization method was applied to visualize

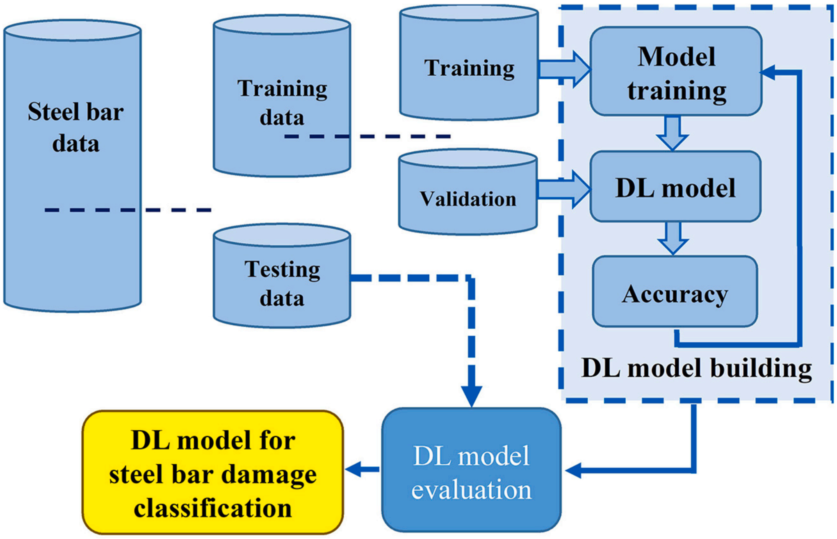


Fig. 2. Deep learning model building architecture.

the exact location of damage on the surface of the steel frame. The validity and effectiveness of the proposed method were verified by performing a series of case studies, along with the results and discussions of the novel method. Finally, the research contributions, limitations, and conclusions were presented.

3. Hybridization of the DCNN method with an image augmentation technique

The method used to classify the damage on steel members is discussed in this section, along with the system database that serves to build the DL model and the process of data augmentation.

3.1. Computational processes

CNNs, which are deep neural networks comprising a multilayer network wherein the neurons in each layer are connected to the neurons in the next layer, are widely used for analyzing visual imagery and solving image classification problems. The proposed method was implemented to identify damaged steel frames. It classifies the steel frame image into either nonconformity (or damaged) or conformity (or undamaged) based on CNNs trained using a supervised learning approach [32]. Various deep CNN models that were trained using the training dataset (i.e., undamaged and damaged steel frame images) were employed for the experiment. A DenseNet-based DCNN model was implemented to classify the damage to the steel frame. A diversified steel frame dataset was populated by varying experimental parameters. An image data augmentation technique that can increase the magnitude of the dataset was implemented. The model was trained using a data augmentation technique to obtain a more efficient steel frame damage classification. The performance of the model was validated by comparing it with that of existing pretrained models (i.e., MobileNet and ResNet). In addition, the exact damage location was visualized using the Grad-CAM approach.

The steel frame data were divided into training and testing datasets, as shown in Fig. 2. The images in the training set were used to build the DL model, which was validated and fine-tuned to enhance its accuracy in each iteration of the model building. The DL model was then validated by analyzing its performance using the available testing data. Thereafter, the performances of the trained DL models that were developed using different deep CNN architectures, relative to the various evaluation metrics, were confirmed using an exclusive testing dataset.

The DL model validated for classifying the real-time damage inspection of steel structures captures the area to be inspected, inputs the captured image data into the DCNN, and analyzes and classifies the image data into damaged and undamaged steel frames, as shown in

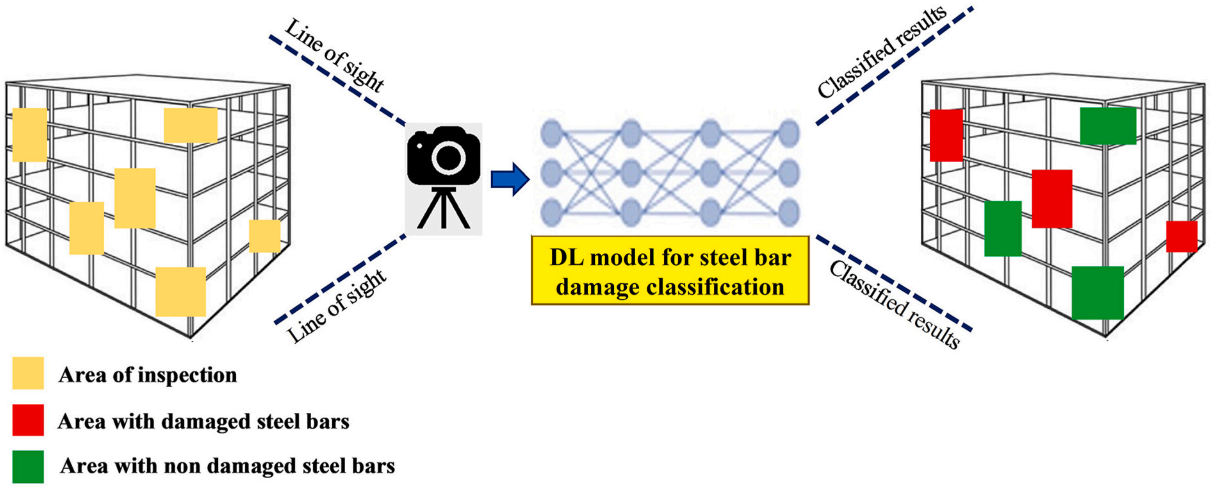

Line of sight
Line of sight
DL model for steel bar damage classification
Classified results
Classified results
Area of inspection
Area with damaged steel bars
Area with non damaged steel bars

Fig. 3. Steel frame assessment.

Fig. 3 (a), (b), and (c).

Table 1
Manipulated parameters for populating the dataset.

Parameters	Values
Dimension of steel frame (S)	50 × 50 mm and 75 × 75 mm
Height of camera placement (H)	50 cm, 75 cm, 100 cm
Distance between steel frame and camera (D)	5 cm, 10 cm, 20 cm, 30 cm, 40 cm, 50 cm
Measurement angle (θ)	0°, 10°, 20°, 30°, 40°, 50°, 60°, 70°, 80°, 90°
Image dimension (I)	2304 × 1728

3.2. *Populating the steel frame image dataset*

Dataset collection for computer-vision modeling is an extremely delicate task that dictates the scope of the research project. In this study, data attributes were carefully selected to achieve usable and valid model training. The computer vision-based model, which classified the damage on steel frames, used various attributes associated with image data. These attributes (or parameters) included the dimensions of the steel frame cross-section, height of the camera placement, distance between the steel frame and camera, and measurement angle. The images were captured using a higher dimensional camera to produce high-resolution images, which were then cropped and labeled separately. The steel frame images captured with different angles, lighting, and textures formed a large dataset based on permutations of these conditions. This dataset provided high-quality images for training the CNN model. The data collection process increased the chances of generating a greater number of distinct images for model training by data augmentation. The generated dataset augmented the robustness of the model and provided a better deployment accuracy. A proprietary dataset of steel frame images was obtained by considering these parameters for a simulated job site. A series of controlled experiments were performed by manipulating these parameters, as shown in Table 1, while maintaining the laboratory setup arrangement shown in Fig. 4.

Samples of damaged and undamaged steel bars of different sizes (S) were collected to build a steel bar data repository, and images were collected from the selected samples by considering the parameters. A digital camera was used to capture a total of 720 images of 2304 × 1728 resolution that equally contributed to two classes: damaged and undamaged. The dataset contained images obtained based on all the permutations of the manipulated parameters (i.e., the height of camera placement (H), dimensions of the steel frame (S), distances between the steel frame and camera (D), and measurement angles (θ)) by manipulating one of the parameters. The collected images were stored in an operational data repository for classification tasks. The sample images of the damaged and undamaged steel frames presented in Figs. 5 and 6, respectively, were randomly mixed and stored in the repository.

The DL model was constructed in two phases: model training and testing. The model was trained on the available steel frame dataset for damage classification. During testing, the trained model was validated using the performance metrics. Each training and testing phase used separate exclusive image datasets. In addition, another validation dataset was used to verify whether the model overfitted the training dataset during the model training phase. The image labels of the damaged and undamaged steel frames were partitioned into training, validation, and testing datasets, which had 462, 144, and 144 images, respectively. Both the testing and validation datasets were exclusively stored for evaluation during training and hyperparameter-tuning.

3.3. *Augmenting the image data*

It is important to provide image data in an algorithm-understandable format because the performance of the DL model is closely related to the quantity and quality of the supplied data. Building an effective DL model by supplying quality data may contribute to reducing the validation and training errors. Reducing these errors may lead the model under study to be generalized, thereby widening its applicability and usability. Several

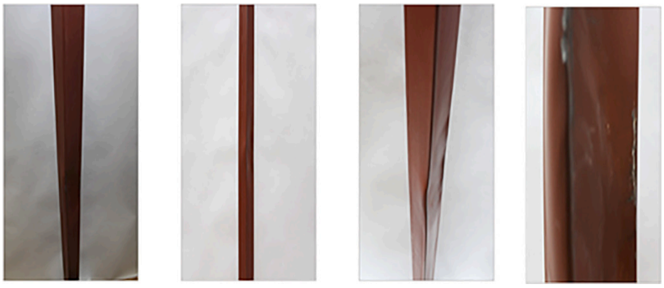


Fig. 5. Damaged steel frame images.

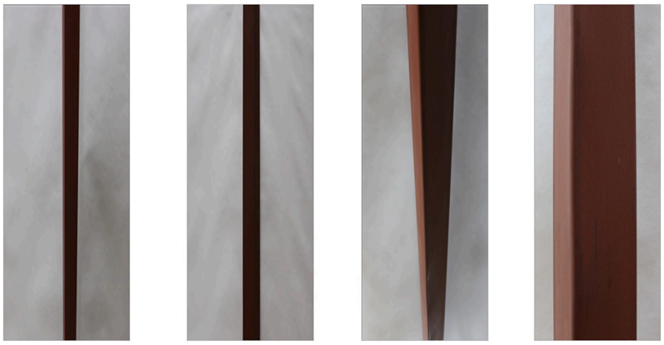


Fig. 6. Undamaged steel frame images.

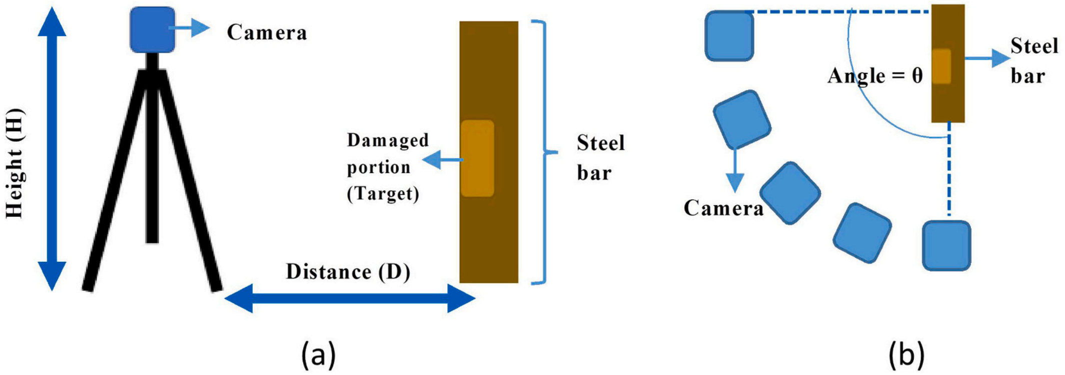


Fig. 4. Data collection arrangement: (a) Camera position; (b) Angle of measurement.

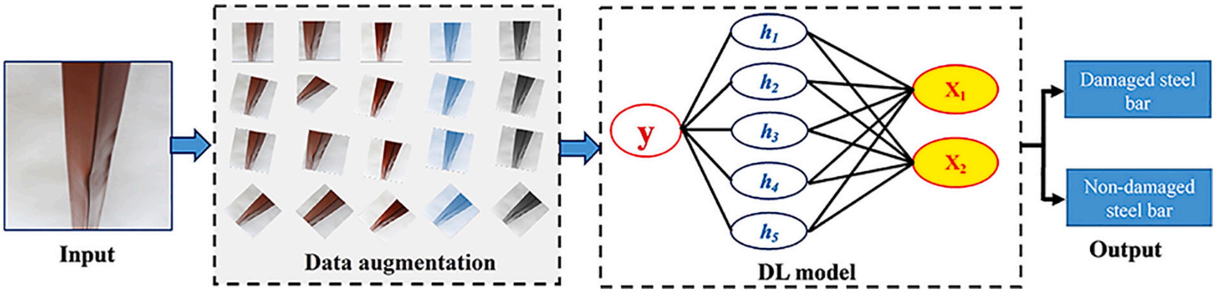


Fig. 7. Image data augmentation.

regularization methods are employed to prevent a model from overfitting the training data. Specifically, data augmentation complements the overfitting issue.

To overcome the limitations involved in collecting a more diverse dataset, image augmentation was performed to synthesize a larger sample set, as using diversified images may increase the robustness of the trained model. The factors affecting the quality of an image (such as distortion, blur, and over or under exposure) were anticipated in the edge devices where the model was employed. Image augmentation was implemented on the training dataset to ensure that the model performed well by learning relevant and significant features from the images. Image augmentation synthesizes more image samples in the training set by manipulating the original image samples through random image effects. The augmented training set included both the original sample and synthesized images. Data augmentation reduces the uniformity in the dataset and optimizes the model based on the actual characteristics of the object. This may decrease the model's chances of bias to the dataset source. The data augmentation of images facilitates the acquisition of competent models, even when a limited amount of data is provided for training and evaluation purposes. The augmented data represent a more comprehensive set than the data points obtained; therefore, the distance between the training and validation sets, as well as that between either of these sets and additional testing sets obtainable in the future, are minimized [56].

Data augmentation was implemented by referring to existing studies (i.e., classifying handwritten digits based on LeNet-5 [57], increasing the dataset size by 2048 in the ImageNet dataset for AlexNet CNN [3], and reducing the error rate of the DL model by solving the overfitting issue [4]). It was performed by cropping 224 × 224 image patches from the original image and flipping them horizontally and vertically in a random fashion, tilting them to a maximum angle of up to 45°, adding Gaussian noise to them, and manipulating their attributes (i.e., brightness, contrast, and hue). After increasing the magnitude of the training dataset four-fold, as shown in Fig. 7, the DL model was trained using the augmented set of steel frames.

3.4. *Pretraining deep CNN models*

CNN architectures typically have multiple convolutional layers for extracting features from input images. Fully connected neural networks are used to classify the input images using feature vectors. The initial CNN models, such as LeNet-5 [2], had five convolution layers. Subsequently, CNN architectures incorporated a deeper and larger number of convolutional layers that could effectively learn the spatial characteristics of an image. Owing to the advancement in innovative methods and architectures, image classification has become more sophisticated and accurate. It is well accepted that DCNN models are often overfitted to the training dataset, resulting in a higher validation loss, and advanced deep architectures are enabled with more functionalities, such as batch normalization, regularization, dropout, and residual connections, resulting in more generalized DCNN models.

3.4.1. *Pretrained deep CNN models for rebar image classification*

Three DCNN models (DenseNet, ResNet, and MobileNet) were trained to evaluate their performances in rebar damage classification in this study. Recent technical advancements have made DCNNs efficient in image classification and segmentation tasks [57]. The overall performance of a DNN depends on the depth and width of the network architecture. However, the deeper and wider the networks, the poorer the convergence, overfitting, and gradient disappearance performances of such comprehensive architectures. Intensive optimizations of hyperparameters and layer arrangements have been performed to address these issues [58].

ResNet outperforms other deep CNN models (i.e., VGG16 and AlexNet) because it learns more complex objects and classifies them efficiently. It is built with residual blocks that are effective at learning an identity function; thus, they can learn identity functions without suffering from vanishing gradients. The ResNet architecture implemented in the proposed model had 50 deep layers to provide a good balance of simplicity and ability. It was relatively less deep compared with the other models and was sufficiently capable of fitting the irregularities among images. MobileNet, which is a spatial CNN, makes a model comparatively shallow and learns efficiently from the input image. It uses depth-wise convolution in its neural architecture, thus making it efficient in learning the spatial features of images across channels. Because it has a smaller neural network layer depth, it reduces training parameter sets, resulting in a smaller model size, and achieves computational economies compared to other deep CNN models. DenseNet is a deeply connected neural layer architecture in which the layers are deeply connected with its previous layers and adopt skip connections from residual blocks. During backpropagation, the parameters of the neural network that are closer are often not tuned, thereby affecting model learning. However, deeply interconnected neural networks generate gradients that enable the DenseNet model to learn more effectively, and it is considerably smaller and more efficient in computations than ResNet and MobileNet. Considering these advantages, DenseNet, ResNet, and MobileNet models were selected to develop a steel frame damage detection model.

DenseNet contains a transition layer between two consecutive dense blocks. The transition layer reduces the number of input feature maps by using a 1 × 1 convolution kernel and halves the number of input feature maps by using a 2 × 2 average pooling layer. These two operations can reduce the computational load on the network [59]. ResNet consists of 152-layer-deep residual nets that address the issue of learning complex functions and vanishing gradients. With ResNet, the gradients can flow directly through the skip connections backward from the latter layers to the initial filters. Moreover, each layer follows the same pattern. MobileNet is a network model that uses depth-wise separable convolution as its core. The depth-wise convolution and point convolution layers are considered as two separate convolution layers in MobileNet. The input feature maps of each depth-wise convolution layer in the dense block are the superposition of the output feature maps in the previous convolution layer [60].

The DenseNet architecture shown in Fig. 8 was implemented to

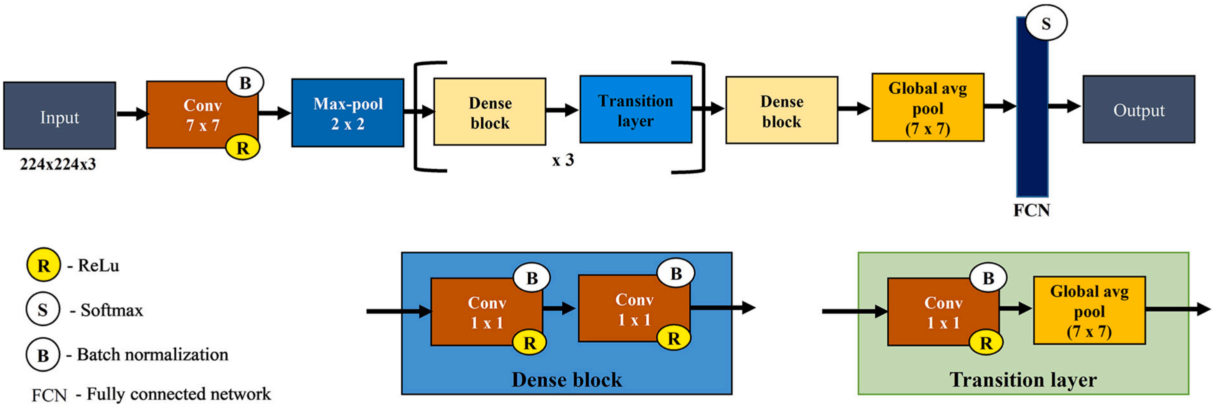


Fig. 8. DenseNet layer stack.

Table 2
Proposed pretrained CNN model comparison.

Model	No. of parameters	Convolution layers	Inference time	Size
DenseNet	6 M+	121	1.321 s	39 Mb
MobileNet	4 M+	88	1.381 s	16 Mb
ResNet	25 M+	50	2.243 s	98 Mb

resolve the issue of using a CNN in DL. When a model is extremely layered and deep, the information provided to the input layers may be lost before it reaches the output layer, and the model parameters may be highly susceptible to exploding or vanishing gradients. Note that the DenseNet architecture simplifies the connectivity proposed earlier by ResNet. DenseNet layers are narrow, implying a smaller number of parameters to learn, thereby reducing the size of the model. The DenseNet architecture minimizes the learning of redundant feature maps [61].

In MobileNet, batch normalization is conducted in each convolution layer; accordingly, the number of feature maps is reduced. In addition, MobileNet reduces the size of the feature map by using a convolution layer instead of a pooling layer [62]. The performance of each pretrained model with different model parameters are shown in Table 2. DenseNet concatenates the output feature map with the incoming feature map, while the others add them. ResNet features less inference time compared with the other models by keeping the parameter learning concise among the layers.

3.4.2. DenseNet for steel frame damage identification

A DCNN architecture with learnable parameters was implemented to build the model. Parameter weights from the pretrained CNN model of the proposed architecture trained using ImageNet data were considered as the input. The proposed DenseNet model for rebar damage identification, shown in Fig. 9, consisted of two major parts: a CNN layer stack arrangement and damage visualization.

The computational operations of the CNN layer stack arrangement and Grad-CAM visualization method for damage classification are explained in the following sections.

3.4.2.1. Steel frame damage classification. The CNN layer stack arrangement extracts features from the input image and generates feature vectors. The vectors are forwarded to an FCN for classification. The CNN layer arrangement and FCN are called the feature extractor and damage classifier, respectively. In the steel frame damage classification, the FCN is set to generate two softmax classes: damaged and undamaged steel rebars. For feature extraction, the input image is passed onto a

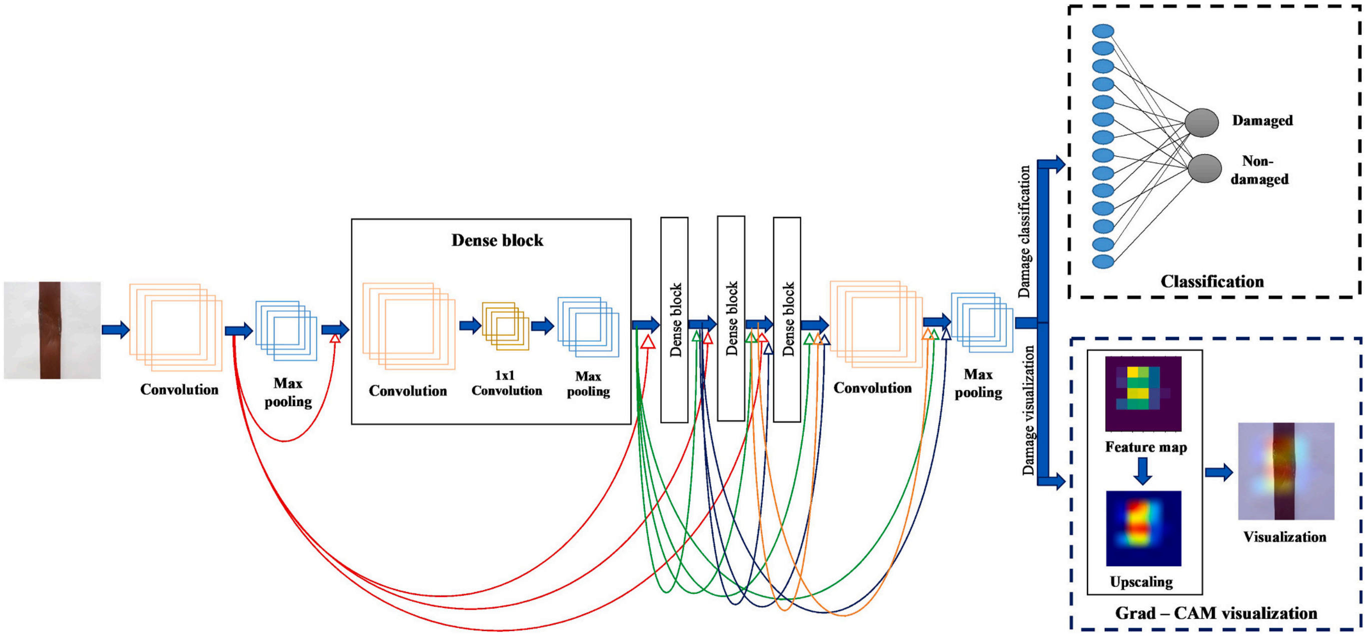


Fig. 9. Proposed DenseNet model.

Table 3
DenseNet algorithm for steel frame damage classification.

```
DenseNet algorithm

Programming Language used for implementation: Python
Libraries used for DenseNet model building: Tensorflow and Keras
Libraries used for image augmentation: OpenCV and computer vision library.
Libraries used for visualizations: Matplotlib and 2D graph tool
1. Let X be the input image of batch, y be the label for the image X.
2. PerformFeature Extraction on the image using DenseNet CNN Algorithm.
a. Obtain feature maps of the 1st layer $a_0$ after passing the image into convolution
   layer of 32 filters of dimension 7 × 7 and apply batch normalization function along
   with ReLU function.
b. Apply Max Pooling function to $a_0$.
c. for$r = 6, 12, 24$ do
   i. Dense Block, repeat for $r$ times
      1. Obtain$x_{l-1}$ output feature map from previous layer.
      2. Pass$x_{l-1}$ to convolution layer of 128 filters of dimension 1 × 1 with batch
         normalization and ReLU activation and obtain $y_l$.
      3. Pass$y_l$ to convolution layer of 32 filters of dimension 3 × 3 with batch
         normalization and ReLU activation function to obtain $y_l$.
      4. Concatenate$x_{l-1}$ and $y_l$ to get $x_l$.
   ii. Transition Layer
      1. Obtain output of Dense Block $x_{l-1}$ and pass it to convolutional layer with
         filter of dimension 1 × 1, followed by batch normalization and ReLU
         activation function to obtain $x_l$.
      2. Pass the $x_l$ to Global Average Pooling Algorithm and store it in $x_l$.
end for

d. Dense Block, repeat 16 times
   i. Obtain$x_{l-1}$ output feature map from previous layer.
   ii. Pass$x_{l-1}$ to convolution layer of 128 filters of dimension 1 × 1 with batch
       normalization and ReLU activation and obtain $y_l$.
   iii. Pass$y_l$ to convolution layer of 32 filters of dimension 3 × 3 with batch
        normalization and ReLU activation function to obtain $y_l$.
   iv. Concatenate$x_{l-1}$ and $y_l$ to obtain $x_l$ feature map.
e. Flatten the $x_l$ feature map to get feature vectors.

3. Run Feature Classification Network on the feature vector.
   a. Initialize weights $w$ and bias $b$ arrays of Linear Network with 256 Neural
      Nodes.
   b. Perform$z = w.\ a_{feature} + b$.
   c. Perform ReLU Activation Function $a^l = \max(z, 0)$.
   d. Initialize weights $w$ and bias $b$ arrays of Linear Network with two Neural
      Nodes.
   e. Perform$z = w.\ a^l + b$.
   f. Perform ReLU Activation Function $a^l = \max(z, 0)$.
   g. Apply softmax function on $a^l$ to obtain probability distribution of the two
      classes: damaged and undamaged rebar.
```

convolution layer of 32 filters with a 7 × 7 dimension, and a batch normalization function was applied along with the ReLU function. The features extracted from the input rebar images by the convolutional layer were used by the FCN for damage classification. The convolutional layers in the dense block had weights loaded from the pretrained model.

Images from the steel frame dataset were segmented into mini-batches with a batch size of 32 images. These batches were fed through a data augmentation layer, which manipulated the input images using multiple parameters, as described in Section 3.3. The augmented images were used to train the model. During the training process, a batch of images was passed through a queue of the model. The model generated predictions after processing the images sequentially and individually. These predictions were compared with the original label, and the losses were calculated using the loss function and binary categorical cross entropy; the losses were thereafter used to calculate the gradients for each layer of the model. The Adam optimization function was used to update the parameters of the model layers during back propagation [40]. The training process continued using all batches of the dataset, and the model was updated. The computational algorithm of DenseNet for feature extraction and feature classification is illustrated as a pseudocode in Table 3.

The model was evaluated by calculating the validation loss obtained using the model predictions with the validation dataset. This training and validation constituted one epoch. The number of epochs was manipulated to ensure the prediction quality. The CNN model performances with different architectures were evaluated by manipulating the number of epochs. It was confirmed that no back propagation would occur while validating the model. Each model architecture demonstrated a distinctive mechanism wherein the convolutional networks extracted the features.

3.4.2.2. Steel frame damage visualization. A novel Grad-CAM visualization method that identified the exact damage location on the surface of the steel frame was implemented and validated in this study. Training a CNN-based image classification model is challenging and can become counterintuitive if the training process is not done carefully. Deep CNN models have become more powerful in solving the complexities of image features. It is important to ensure that the developed model is capable of learning relevant features during training and its predictions are validated. The efficiency of the DL model can be ensured by decoding the activated perceptron. Grad-CAM is one such method for visualizing activated perceptrons by generating a heatmap that provides insight into steel bar damage classification. The activated neurons of the Grad-CAM visualization provide insights into the damaged and undamaged locations of the steel bars. Thus, this visualization technique was used to validate the efficiency of the model. The output from the final convolution and maxpool layers of the proposed DenseNet architecture generated a downscaled 7 × 7 feature map. The downscaled feature map from the DenseNet architecture was upscaled according to the original size of the input image. The internal operation and heat map visualization for the damage identification are shown in Fig. 10. Once the feature map was upscaled, the Grad-CAM visualization superimposed an upscaled feature map over the original input image [64]. The superimposing of images facilitated the identification of the damaged spot on the surface of the steel frame.

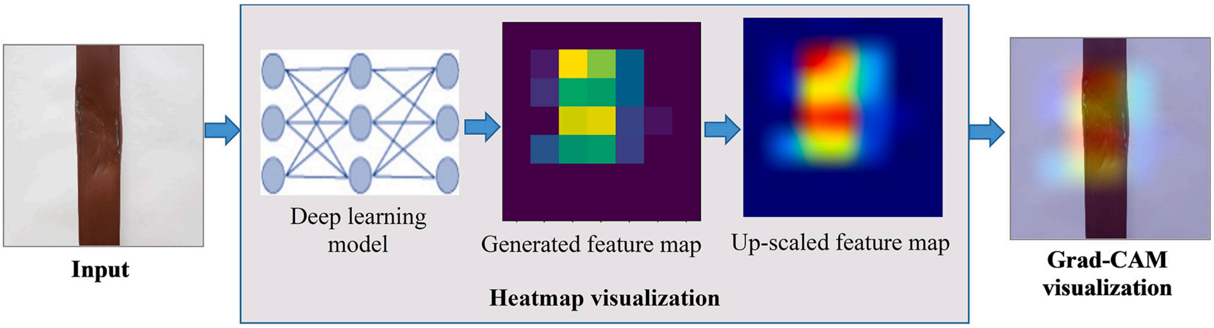


Fig. 10. Grad-CAM visualization for steel frame damage identification.

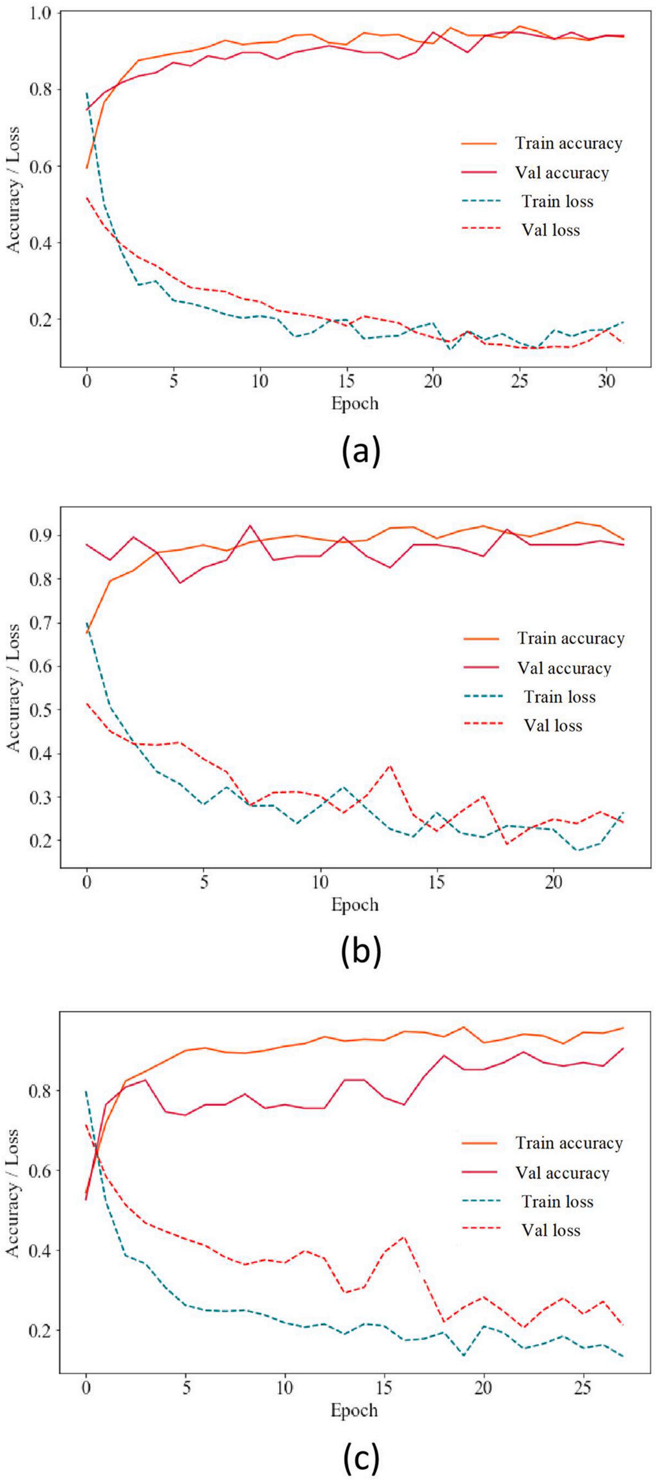


Fig. 11. Learning curve of models during training and validation phase: (a) DenseNet; (b) MobileNet; (c) ResNet.

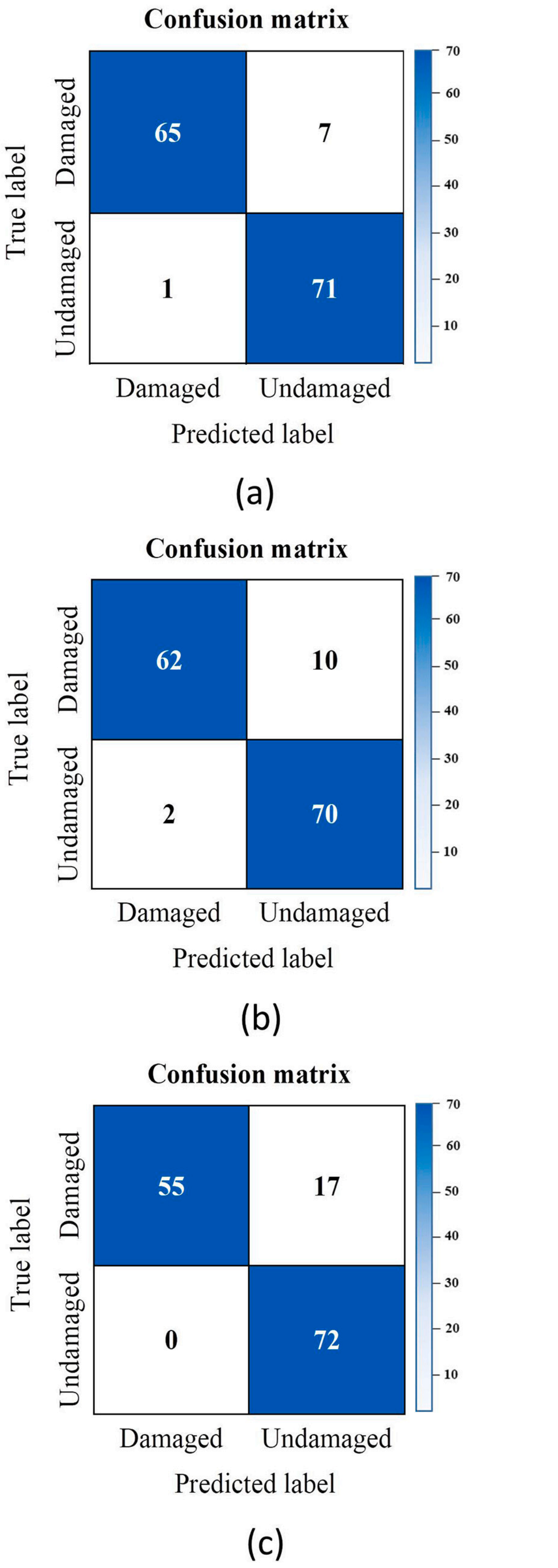


Fig. 12. Confusion matrix for damage classification: (a) DenseNet; (b) MobileNet; (c) ResNet.

4. Results and discussions

4.1. Steel frame classification using different pretrained models

The model classifiers were trained using batches of 32 images for each iteration of 30 epochs. The classification accuracies of the models were evaluated using a confusion matrix that provided a correlation

Table 4
Score metric of the models.

Model	Balanced accuracy	Cohen's Kappa score	AUROC score
DenseNet	94.8%	97.8%	99.1%
MobileNet	92.2%	95.6%	97.9%
ResNet	90.4%	89.4%	93.8%

between the prediction of the model and the actual class label of the data points. The learning curves of the DL models during the training and testing phases are shown in Fig. 11(a), (b), and (c). During model training, the ResNet50, MobileNet, and DenseNet-121 classifiers achieved maximum classification accuracies of 94.5%, 93.9%, and 99.3%, respectively. DenseNet-121 outperformed the other models in terms of classification accuracy during the testing phase.

The DenseNet-121 classifier achieved almost perfect classification scores with only seven false positives (FPs) and one false negative (FN), whereas MobileNet obtained 10 FPs and 70 FNs for 144 test images of the test set, as shown in the confusion matrix in Fig. 12(a) and (b). The ResNet classifier achieved the lowest accuracy as compared with the other classifiers. It produced 17 FPs and 72 FNs as shown in the confusion matrix in Fig. 12(c).

The performances of the models were assessed using additional metrics (i.e., balanced accuracy, and Cohen's Kappa and ROCAUC scores). The balanced accuracy metric evaluates the performance of a model using imbalanced datasets and classifies the balance accuracy in binary using Eq. 1.

$$balanced\ accuracy = \frac{1}{2}\times\left(\frac{TP}{TP+FN}+\frac{TN}{TN+FP}\right), \tag{1}$$

where TP is the total number of positives, FN is the number of false negatives, TN is the total number of negatives, and FP is the number of false positives.

A model with a balanced accuracy score greater than 0.5 is considered to outperform the others. The Cohen's Kappa score k, which is a metric that compares observed and expected accuracies, was calculated using Eqs. 2, 3, and 4.

$$\mathrm{k} = \frac{p_o - p_e}{1 - p_e}, \tag{2}$$

where p_o is the relative observed accuracy among classifiers, and p_e is the hypothetical probability of chance agreement.

$$p_0 = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

$$p_e = \frac{(TN+FP)*(TN+FN)+(FN+TP)*(FP+TP)}{(TP+TN+FP+FN)*(TP+TN+FP+FN)} \tag{4}$$

The Cohen's Kappa score was always less than or equal to 1. The greater the Cohen's Kappa score, the greater the accuracy of the model. The Cohen's Kappa score of DenseNet (97.7%) was greater than that of MobileNet (95.6%) and ResNet (89.4%), as shown in Table 4.

The AUROC curve represents the performance of the classification models, given various thresholds. The ROC is a probability curve plotting the true positive rate over the false positive rate. The AUC is the

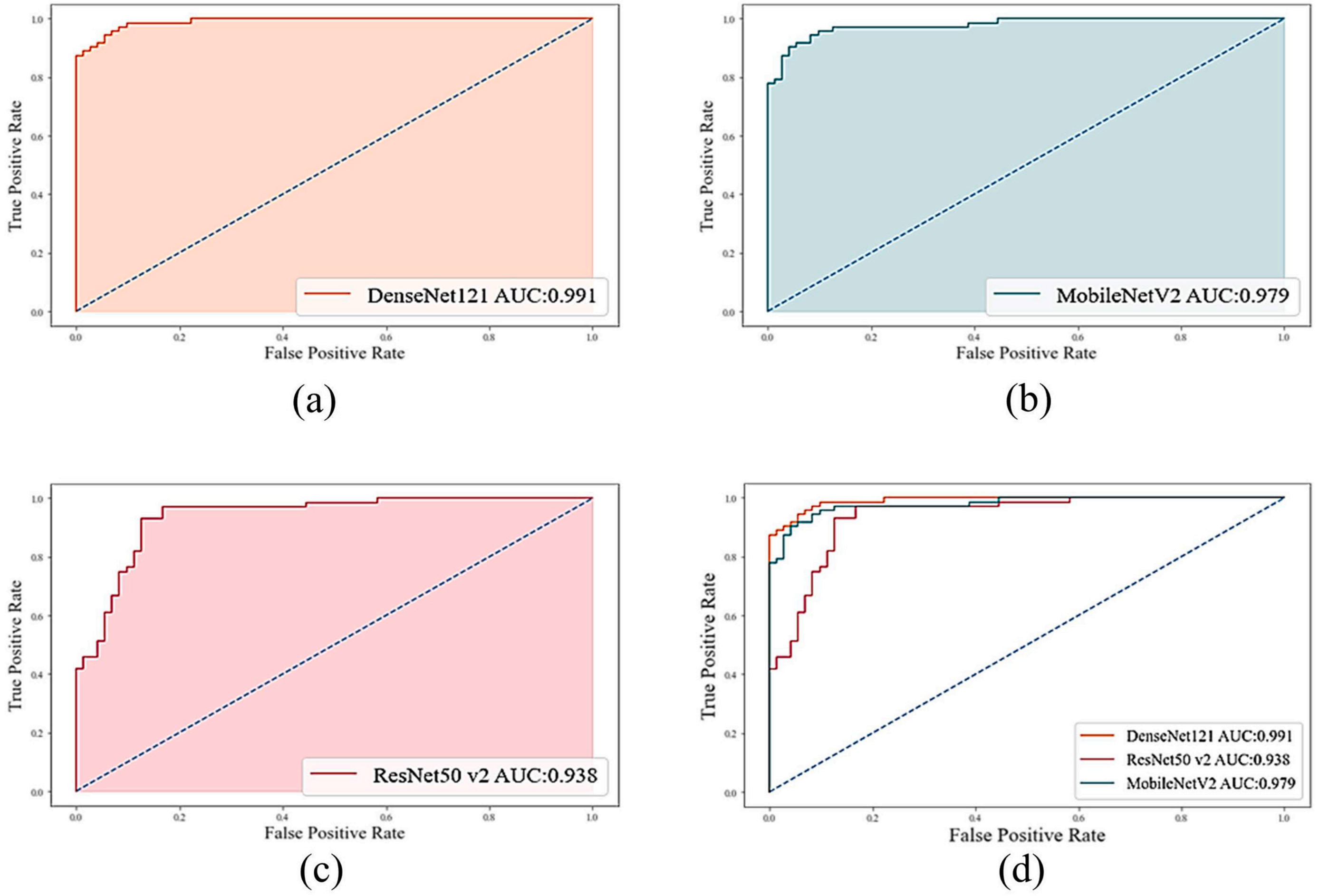


Fig. 13. AUROC of models for steel bar damage classification: (a) DenseNet; (b) MobileNet; (c) ResNet; (d) AUC comparison of the models.

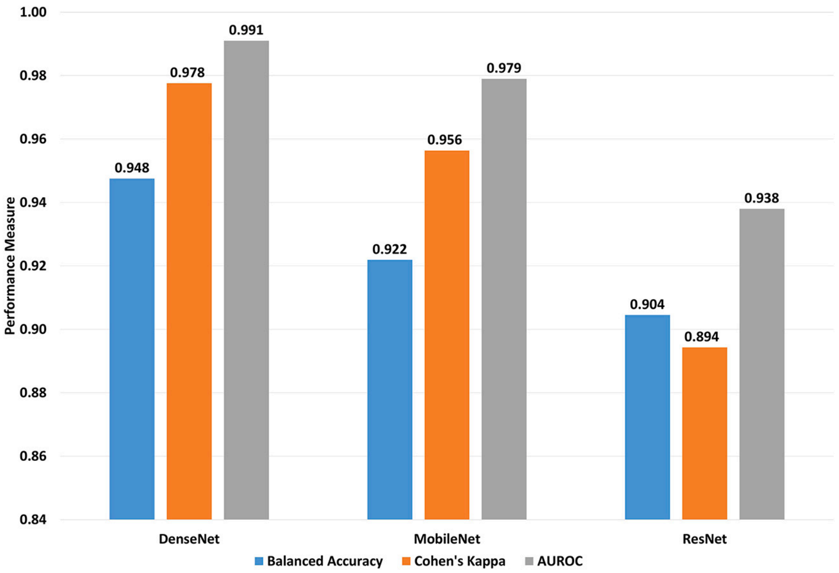


Fig. 14. Performance comparison of classification models.

degree of separability that ranges from 1 to 0. The closer the AUC value of the model is to 1, the more suitable the model is for classification. The AUC scores of DenseNet-121, MobileNet, and ResNet were 99.1%, 97.9%, and 93.8%, respectively, as shown in Fig. 13(a), (b), and (c). Indeed, DenseNet-121 outperforms the other classifiers, based on the AUC score.

The AUROC score was considered for measuring model performance in terms of accuracy over both the classes equally. The higher probability value of DenseNet indicates that the model was able to classify the damaged and undamaged steel frames perfectly. In deep connected networks, the flow of gradients from one layer to another is essential to enhance model performance. Densely connected layers in the DenseNet improves the flow of information and gradients throughout the network and enables access to the gradients at each layer of the DenseNet. By using this information, the DenseNet model may reduce overfitting and maintain the maximum AUROC score.

A comparison of the evaluation metric values for the classification model performances is shown in Fig. 14. The results confirmed that the DenseNet model outperformed the MobileNet and ResNet models in terms of identifying steel frame damage. The outstanding metric scores of balanced accuracies, and Cohen's Kappa and AUROC scores, which were 94.8%, 97.8%, and 99.1%, respectively, provide evidence that DenseNet was the best model, followed by MobileNet. Indeed, the fully connected convolution layers that received all the preceding layers as input and featured a strong gradient flow enabled DenseNet to outperform the other classifiers.

The Grad-CAM visualization outputs obtained by using the DenseNet model are presented in Fig. 15. The steel frame image that was input to the DenseNet model generated a downscaled 7 × 7 feature set. To obtain the Grad-CAM visualization, the lower-end feature set was upscaled to the pixel size of the original input image. The upscaled feature set was superimposed on the input image to locate the damaged spot on the surface of the steel frame. It was found that the damaged steel frame images were clearly identified by overlaying the damage on the original image, and the superimposed image did not vary considerably with the original image when the undamaged steel frame was under study as shown in Fig. 15.

5. Conclusion

The main contribution of this study is the development of a DCNN-based architecture that classifies steel frames into damaged and undamaged by identifying the damaged location on the surface of the steel frame. Given that the dataset used to train the models in this study was limited, the research findings provide evidence that the DCNN model achieves the purpose of this study with minimum trade-offs and number of parameters. The densely connected CNN layers in the DenseNet architecture, which were used to implement this study, mitigated the vanishing gradient problem and contributed to achieving superior classification performance despite being trained with fewer parameters. The robustness of the DenseNet architecture was further augmented by integrating the Grad-CAM visualization technique, thereby making the damage classification model more efficient. The integration realized object localization by transferring the knowledge acquired by the model to detect damaged steel frame images. The image augmentation techniques that were implemented to include more adversaries among the training images ensured enhanced model learning. The novel technique implemented in the proposed model may detect damage through knowledge transfer and an intuitive understanding of the CNN model. The validity of the DenseNet model was confirmed by a maximum accuracy of 99.3% obtained through a minimum computation of 30 epochs. The DenseNet model outperformed the other pretrained DL models (MobileNet and ResNet).

This study featured a unique stack arrangement of the DenseNet layer that densely connected all the layers to reduce the problem associated with redundant layers, thereby improving the overall accuracy compared with other DCNN models. This method advances the ongoing research on damage identification and location methods and the body of knowledge in construction management at a job site because it helps project managers achieve the goals involved in quality control. A limitation of this study was the degree of computerization. Because the method was not completely computerized, it may not control the damage to steel frames by detecting them and providing their locations without the involvement of managers in real time. It would be possible to further optimize the hyperparameters of the proposed architecture by compressing the size of the model to develop a fully functional autonomous system. The efficiency of DenseNet runtime may be enhanced using model optimization techniques, which may facilitate the integration of the model into an embedded device. Furthermore, the integration of steel frame damage classification models with autonomous UAVs and/or ground vehicles is intended for a future study.

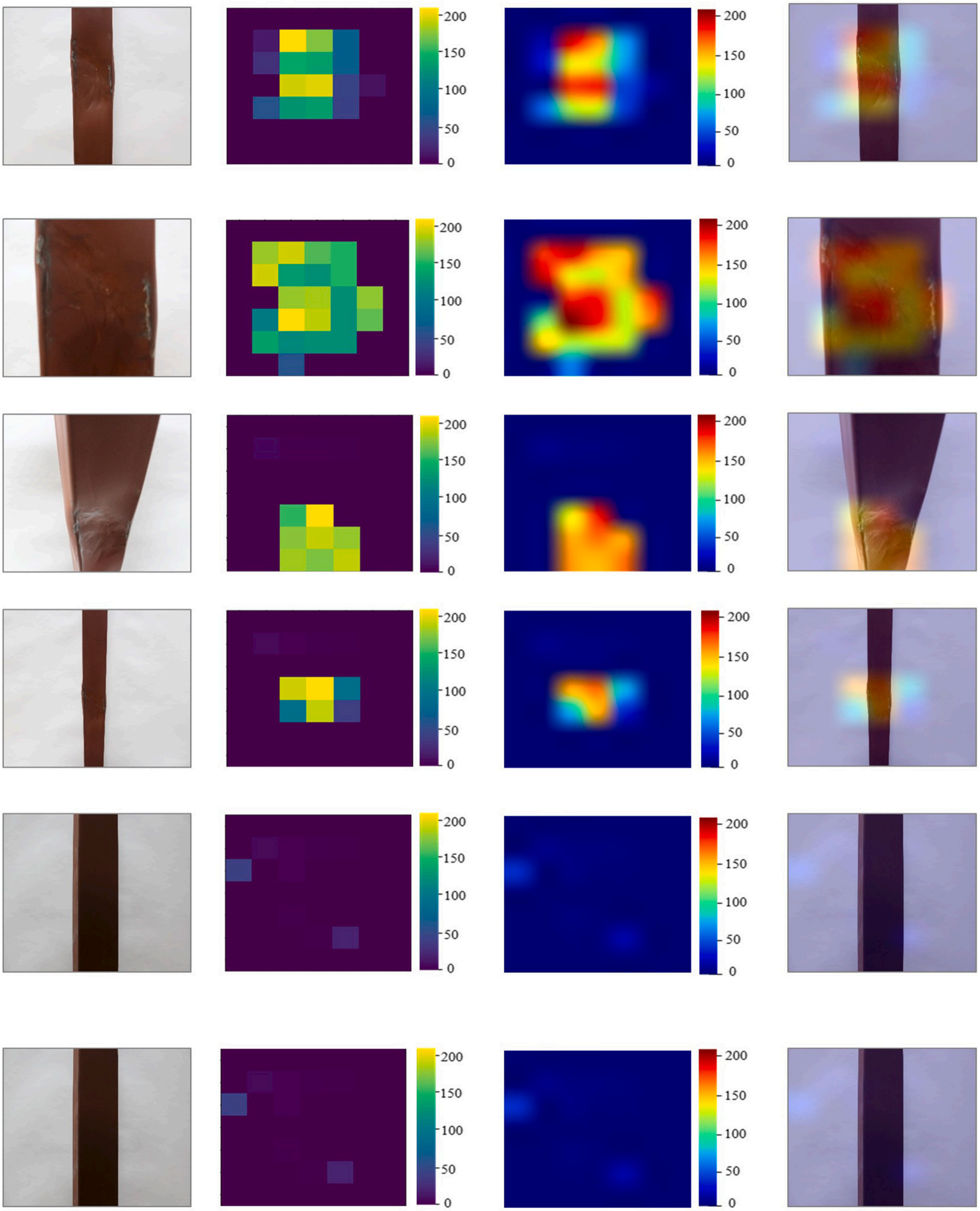


Fig. 15. Results of steel frame classification using the model.

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

This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean Government [NRF-2018R1A5A1025137]. The contributions of the Ministry of Science, ICT, and Future Planning (MSIP) are gratefully acknowledged.

References

[1] J. Seo, L. Duque, J. Wacker, Drone-enabled bridge inspection methodology and application, Autom. Constr. 94 (2018) 112–126, https://doi.org/10.1016/j.autcon.2018.06.006.

[2] B.F. Spencer, V. Hoskere, Y. Narazaki, Advances in computer vision-based civil infrastructure inspection and monitoring, Engineering 5 (2019) 199–222, https://doi.org/10.1016/j.eng.2018.11.030.

[3] M. Alsharqawi, T. Zayed, S.A. Dabous, Integrated condition rating and forecasting method for bridge decks using visual inspection and ground penetrating radar, Autom. Constr. 89 (2018) 135–145, https://doi.org/10.1016/j.autcon.2018.01.016.

[4] R.K. Soman, B. Raphael, K. Varghese, A system identification methodology to monitor construction activities using structural responses, Autom. Constr. 75 (2017) 79–90, https://doi.org/10.1016/j.autcon.2016.12.006.

[5] R. Chacon, R. Zorrilla, Structural health monitoring in incrementally launched steel bridges: patch loading phenomena modeling, Autom. Constr. 58 (2015) 60–73, https://doi.org/10.1016/j.autcon.2015.07.001.

[6] X.W. Ye, Y.Q. Ni, K.Y. Wong, J.M. Ko, Statistical analysis of stress spectra for fatigue life assessment of steel bridges with structural health monitoring data, Eng. Struct. 45 (2012) 166–176, https://doi.org/10.1016/j.engstruct.2012.06.016.

[7] H.K. Shen, P.H. Chen, L.M. Chang, Automated steel bridge coating rust defect recognition method based on color and texture feature, Autom. Constr. 31 (2013) 338–356, https://doi.org/10.1016/j.autcon.2012.11.003.

[8] B.G. Erkal, J.F. Hajjar, Laser-based surface damage detection and quantification using predicted surface properties, Autom. Constr. 83 (2017) 285–302, https://doi.org/10.1016/j.autcon.2017.08.004.

[9] C.V. Dung, H. Sekiya, S. Hirano, T. Okatani, C. Miki, A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks, Autom. Constr. 102 (2019) 217–229, https://doi.org/10.1016/j.autcon.2019.02.013.

[10] R.S. Adhikari, O. Moselhi, A. Bagchi, Image-based retrieval of concrete crack properties for bridge inspection, Autom. Constr. 39 (2014) 180–194, https://doi.org/10.1016/j.autcon.2013.06.011.

[11] S.H. Kim, S.W. Lee, H.S. Mha, Fatigue reliability assessment of an existing steel railroad bridge, Eng. Struct. 23 (2001) 1203–1211, https://doi.org/10.1016/S0141-0296(01)00038-4.

[12] H. Hasni, P. Jiao, A.H. Alavi, N. Lajnef, S.F. Masri, Structural health monitoring of steel frames using a network of self-powered strain and acceleration sensors: a numerical study, Autom. Constr. 85 (2018) 344–357, https://doi.org/10.1016/j.autcon.2017.10.022.

[13] H.N. Li, L. Ren, Z.G. Jia, T.H. Yi, D.S. Li, State-of-the-art in structural health monitoring of large and complex civil infrastructures, J. Civ. Struct. Heal. Monit. 6 (2016) 3–16, https://doi.org/10.1007/s13349-015-0108-9.

[14] D.Y. Yun, D. Kim, M. Kim, S.G. Bae, J.W. Choi, H.B. Shim, et al., Field measurements for identification of modal parameters for high-rise buildings under construction or in use, Autom. Constr. 121 (2021) 103446, https://doi.org/10.1016/j.autcon.2020.103446.

[15] S. Jeong, R. Hou, J.P. Lynch, H. Sohn, K.H. Law, An information modeling framework for bridge monitoring, Adv. Eng. Softw. 114 (2017) 11–31, https://doi.org/10.1016/j.advengsoft.2017.05.009.

[16] K. Worden, E.J. Cross, On switching response surface models, with applications to the structural health monitoring of bridges, Mech. Syst. Signal Process. 98 (2018) 139–156, https://doi.org/10.1016/j.ymssp.2017.04.022.

[17] G. Zhou, A. Li, J. Li, M. Duan, Structural health monitoring and time-dependent effects analysis of self-anchored suspension bridge with extra-wide concrete girder, Appl. Sci. 8 (2018) 115, https://doi.org/10.3390/app8010115.

[18] A. Guo, A. Jiang, J. Lin, X. Li, Data mining algorithms for bridge health monitoring: Kohonen clustering and LSTM pre-diction approaches, J. Supercomput. 76 (2020) 932–947, https://doi.org/10.1007/s11227-019-03045-8.

[19] N.K. Mutlib, S. Baharom, A. El-Shafie, M.Z. Nuawi, Ultrasonic health monitoring in structural engineering: buildings and bridges, Struct. Control. Health Monit. 23 (2015) 409–422, https://doi.org/10.1002/stc.1800.

[20] Q. Han, J. Xu, A. Carpinteri, G. Lacidogna, Localization of acoustic emission sources in structural health monitoring of masonry bridge, Struct. Control. Health Monit. 22 (2014) 314–329, https://doi.org/10.1002/stc.1675.

[21] T. Yamaguchi, S. Hashimoto, Fast crack detection method for large-size concrete surface images using percolation-based image processing, Mach. Vis. Appl. 21 (2010) 797–809, https://doi.org/10.1007/s00138-009-0189-8.

[22] J. Yan, A. Downey, A. Cancelli, S. Laflamme, A. Chen, J. Li, F. Ubertini, Concrete crack detection and monitoring using a capacitive dense sensor array, Sensors 19 (2019) 1843, https://doi.org/10.3390/s19081843.

[23] B. Kim, N. Yuvaraj, K.R. Sri Preethaa, G. Hu, D.E. Lee, Wind-induced pressure prediction on tall buildings using generative adversarial imputation network, Sensors 21 (2021) 2515, https://doi.org/10.3390/s21072515.

[24] B. Kim, N. Yuvaraj, T. Tse, D.E. Lee, G. Hu, Pressure pattern recognition in buildings using an unsupervised machine-learning algorithm, J. Wind Eng. Ind. Aerodyn. 214 (2021) 104629, https://doi.org/10.1016/j.jweia.2021.104629.

[25] R. Ghiasi, P. Torkzadeh, M. Noori, A machine-learning approach for structural damage detection using least square support vector machine based on a new combinational kernel function, Struct. Control. Health Monit. 15 (2016) 302–316, https://doi.org/10.1177/1475921716639587.

[26] M. Mehrjoo, N. Khaji, H. Moharrami, A. Bahreininejad, Damage detection of truss bridge joints using artificial neural networks, Expert Syst. Appl. 35 (2008) 1122–1131, https://doi.org/10.1016/j.eswa.2007.08.008.

[27] M.P. Gonzalez, J.L. Zapico, Seismic damage identification in buildings using neural networks and modal data, Comput. Struct. 86 (2008) 416–426, https://doi.org/10.1016/j.compstruc.2007.02.021.

[28] P.J. Chun, H. Yamashita, S. Furukawa, Bridge damage severity quantification using multipoint acceleration measurement and artificial neural networks, Shock. Vib. 2015 (2015) 789384, https://doi.org/10.1155/2015/789384.

[29] Z. Farahmandpour, M. Seyedmahmoudian, A. Stojcevski, I. Moser, J.G. Schneider, Cognitive service virtualisation: a new machine learning-based virtualization to generate numeric values, Sensors 20 (2020) 5664, https://doi.org/10.3390/s20195664.

[30] Z. Liang, C. Wang, Z. Duan, H. Liu, X. Liu, K.U.J. Khan, A hybrid model consisting of supervised and unsupervised learning for landslide susceptibility mapping, Remote Sens. 13 (2021) 1464, https://doi.org/10.3390/rs13081464.

[31] P.C. Tung, Y.R. Hwang, M.C. Wu, The development of a mobile manipulator imaging system for bridge crack inspection, Autom. Constr. 11 (2002) 717–729, https://doi.org/10.1016/S0926-5805(02)00012-2.

[32] A. Akagic, E. Buza, S. Omanovic, A. Karabegovic, Pavement crack detection using Otsu thresholding for image segmentation, in: Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2018, pp. 1092–1097, https://doi.org/10.23919/MIPRO.2018.8400199.

[33] J. Tang, Y. Gu, Automatic crack detection and segmentation using a hybrid algorithm for road distress analysis, in: Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, 2013, pp. 3026–3030, https://doi.org/10.1109/SMC.2013.516.

[34] L.R. Salim, H.M. La, S. Zeyong, S. Weihua, Developing a crack inspection robot for bridge maintenance, in: Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, 2011, pp. 6288–6293.

[35] J.W. Kim, S.B. Kim, J.C. Park, J.W. Nam, Development of crack detection system with unmanned aerial vehicles and digital image processing, in: Proceedings of the 2015 World Congress on Advances in Structural Engineering and Mechanics (ASEM15), Incheon, Korea, 2015. W3F.2.MS582_2545F1.pdf (i-asem.org).

[36] Z. Chen, H. Li, Y. Bao, N. Li, Y. Jin, Identification of spatio-temporal distribution of vehicle loads on long-span bridges using computer vision technology, Struct. Control. Health Monit. 23 (2015) 517–534, https://doi.org/10.1002/stc.1780.

[37] H.N. Ho, K.D. Kim, Y.S. Park, J.J. Lee, An efficient image-based damage detection for cable surface in cable-stayed bridges, NDT E Int. 58 (2013) 18–23, https://doi.org/10.1016/j.ndteint.2013.04.006.

[38] Y.J. Cha, K. You, W. Choi, Vision-based detection of loosened bolts using the Hough transform and support vector machines, Autom. Constr. 71 (2016) 181–188, https://doi.org/10.1016/j.autcon.2016.06.008.

[39] K. Bubryur, N. Yuvaraj, K.R.S. Preethaa, R.A. Pandian, Surface crack detection using deep learning with shallow CNN architecture for enhanced computation, Neural Comput. & Applic. 33 (2021) 9289–9305, https://doi.org/10.1007/s00521-021-05690-8.

[40] J.J. Lee, S.J. Cho, M. Shinozuka, C.B. Yun, C.G. Lee, W.T. Lee, Evaluation of bridge load carrying capacity based on dynamic displacement measurement using real-time image processing techniques, Int. J. Steel Struct. 6 (2006) 377–385.

[41] S. Yen, C.H. Wang, J. Chien, Accurate and robust ROI localization in a camshift tracking application, Multimed. Tools Appl. 74 (2014) 10291–10312, https://doi.org/10.1007/s11042-014-2167-z.

[42] X.W. Ye, C.Z. Dong, T. Liu, Image-based structural dynamic displacement measurement using different multiobject tracking algorithms, Smart Struct. Syst. 17 (2016) 935–956, https://doi.org/10.12989/sss.2016.17.6.935.

[43] H.W. Schreier, M.A. Sutton, Systematic errors in digital image correlation due to under matched subset shape functions, Exp. Mech. 42 (2002) 303–310, https://doi.org/10.1177/001448502321548391.

[44] J. Guo, C. Zhu, Dynamic displacement measurement of large-scale structures based on the Lucas-Kanade template tracking algorithm, Mech. Syst. Signal Process. 66 (2016) 425–436, doi:10.1016%2Fj.ymssp.2015.06.004.

[45] P. Asadi, M. Gindy, M. Alvarez, A. Asadi, A computer vision-based rebar detection chain for automatic processing of concrete bridge deck GPR data, Autom. Constr. 112 (2020) 103106, https://doi.org/10.1016/j.autcon.2020.103106.

[46] P. Martinez, R. Ahmad, M. Al-Hussein, A vision-based system for pre-inspection of steel frame manufacturing, Autom. Constr. 97 (2019) 151–163, https://doi.org/10.1016/j.autcon.2018.10.021.

[47] X.W. Ye, C.Z. Dong, T. Liu, Force monitoring of steel cables using vision-based sensing technology: Methodology and experimental verification, Smart Struct. Syst. 18 (2016) 585–599, https://doi.org/10.12989/sss.2016.18.3.585.

[48] F.N.S. Medeiros, G.L.B. Ramalho, M.P. Bento, L.C.L. Medeiros, On the evaluation of texture and color features for non-destructive corrosion detection, EURASIP J. Adv. Signal Process. 817473 (2010), https://doi.org/10.1155/2010/817473.

[49] A.A. Nik, F.M. Nejad, H. Zakeri, Hybrid PSO and GA approach for optimizing surveyed asphalt pavement inspection units in massive network, Autom. Constr. 71 (2016) 325–345, https://doi.org/10.1016/j.autcon.2016.08.004.

[50] P.J. Chun, T. Yamane, S. Izumi, T. Kameda, Evaluation of tensile performance of steel members by analysis of corroded steel surface using deep learning, Metals 9 (2019) 1259, https://doi.org/10.3390/met9121259.

[51] H.S. Park, B.K. Oh, Real-time structural health monitoring of a supertall building under construction based on visual modal identification strategy, Autom. Constr. 85 (2018) 273–289, https://doi.org/10.1016/j.autcon.2017.10.025.

[52] M. Mangal, J.C.P. Cheng, Automated optimization of steel reinforcement in RC building frames using building information modeling and hybrid genetic algorithm, Autom. Constr. 90 (2018) 39–57, https://doi.org/10.1016/j.autcon.2018.01.013.

[53] F. Marzoughi, T. Arthanari, D. Askarany, A decision support framework for estimating project duration under the impact of weather, Autom. Constr. 87 (2018) 287–296, https://doi.org/10.1016/j.autcon.2017.11.001.

[54] G. Moradi, M. Shamsi, M.H. Sedaaghi, S. Moradi, Using statistical histogram based EM algorithm for apple defect detection, Am. J. Signal Process. 2 (2012) 10–14, https://doi.org/10.5923/j.ajsp.20120202.02.

[55] G. Li, X. Zhao, K. Du, F. Ru, Y. Zhang, Recognition and evaluation of bridge cracks with modified active contour model and greedy search-based support vector machine, Autom. Constr. 78 (2017) 51–61, https://doi.org/10.1016/j.autcon.2017.01.019.

[56] C.V. Dung, L.D. Anh, Autonomous concrete crack detection using deep fully convolutional neural network, Autom. Constr. 99 (2019) 52–58, https://doi.org/10.1016/j.autcon.2018.11.028.

[57] B. Zhong, H. Wu, L. Ding, P.E.D. Love, H. Li, H. Luo, et al., Mapping computer vision research in construction: developments, knowledge gaps and implications for research, Autom. Constr. 107 (2019) 102919, https://doi.org/10.1016/j.autcon.2019.102919.

[58] T.C. Huynh, Vision-based autonomous bolt-looseness detection method for splice connections: design, lab-scale evaluation, and field application, Autom. Constr. 124 (2021) 103591, https://doi.org/10.1016/j.autcon.2021.103591.

[59] W. Fang, L. Ding, P.E.D. Love, H. Luo, H. Li, F. Pena-Mora, et al., Computer vision applications in construction safety assurance, Autom. Constr. 110 (2020) 103013, https://doi.org/10.1016/j.autcon.2019.103013.

[60] W. Fang, B. Zhong, N. Zhao, P.E.D. Love, H. Luo, J. Xue, et al., A deep learning-based approach for mitigating falls from height with computer vision: convolutional neural network, Adv. Eng. Inform. 39 (2019) 170–177, https://doi.org/10.1016/j.aei.2018.12.005.

[61] B. Zhong, X. Xing, P. Love, X. Wang, H. Luo, Convolutional neural network: deep learning-based classification of building quality problems, Adv. Eng. Inform. 40 (2019) 46–57, https://doi.org/10.1016/j.aei.2019.02.009.

[62] W. Fang, L. Ding, B. Zhong, P.E. Love, H. Luo, Automated detection of workers and heavy equipment on construction sites: a convolutional neural network approach, Adv. Eng. Inform. 37 (2018) 139–149, https://doi.org/10.1016/j.aei.2018.05.003.

[63] H. Oliveira, P.L. Correia, Automatic road crack detection and characterization, IEEE Trans. Intell. Transp. Syst. 14 (2013) 155–168, https://doi.org/10.1109/tits.2012.2208630.

[64] Y. Xu, Y.Q. Bao, J.H. Chen, W.M. Zuo, H. Li, Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images, Struct. Health Monit. 18 (2019) 653–674, https://doi.org/10.1177/1475921718764873.

[65] Y.J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, Comput.-Aided Civ. Inf. Eng. 32 (2017) 361–378, https://doi.org/10.1111/mice.12263.

[66] A. Zhang, K.C. Wang, B. Li, E. Yang, X. Dai, Y. Peng, et al., Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network, Comput.-Aided Civ. Inf. Eng. 32 (2017) 805–819, https://doi.org/10.1111/mice.12297.

[67] S. Zhou, W. Song, Concrete roadway crack segmentation using encoder-decoder networks with range images, Autom. Constr. 120 (2020) 103403, https://doi.org/10.1016/j.autcon.2020.103403.

[68] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, P. Fieguth, A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure, Adv. Eng. Inform. 29 (2015) 196–210, https://doi.org/10.1016/j.aei.2015.01.008.

[69] J.N. Yang, Y. Xia, C.H. Loh, Damage identification of bolt connections in a steel frame, J. Struct. Eng. 140 (2014) 1–9, https://doi.org/10.1061/(ASCE)ST.1943-541X.0000831.

[70] U.H. Billah, H.M. La, A. Tavakkoli, Deep learning-based feature silencing for accurate concrete crack detection, Sensors 20 (2020) 4403, https://doi.org/10.3390/s20164403.

[71] Q. Yang, W. Shi, J. Chen, W. Lin, Deep convolution neural network-based transfer learning method for civil infrastructure crack detection, Autom. Constr. 116 (2020) 103199, https://doi.org/10.1016/j.autcon.2020.103199.

[72] T.S.R. Mhathesh, J. Andrew, K.M. Sagayam, L. Henesey, A 3D convolutional neural network for bacterial image classification, in: J. Peter, S. Fernandes, A. Alavi (Eds.), Intelligence in Big Data Technologies Beyond the Hype, Advances in Intelligent Systems and Computing 1167, Springer, Singapore, 2020, pp. 419–431, https://doi.org/10.1007/978-981-15-5285-4_42.

[73] Q.D. Cao, Y. Choe, Building damage annotation on post-hurricane satellite imagery based on convolutional neural networks, Nat. Hazards 103 (2020) 3357–3376, https://doi.org/10.1007/s11069-020-04133-2.

[74] B. Kim, N. Yuvaraj, K.R.S. Preethaa, R.A. Pandian, Surface crack detection using deep learning with shallow CNN architecture for enhanced computation, Neural Comput. & Applic. 33 (2021) 9289–9305, https://doi.org/10.1007/s00521-021-05690-8.

[75] M. Dong, H. Wu, H. Hu, R. Azzam, L. Zhang, Z. Zheng, et al., Deformation prediction of unstable slopes based on real-time monitoring and DeepAR model, Sensors 21 (2020) 14, https://doi.org/10.3390/s21010014.

[76] D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, DeepAR: probabilistic forecasting with autoregressive recurrent networks, Int. J. Forecast. 36 (2020) 1181–1191, https://doi.org/10.1016/j.ijforecast.2019.07.001.

[77] F. Huang, J. Zhang, C. Zhou, Y. Wang, J. Huang, L. Zhu, A deep learning algorithm using a fully connected sparse autoencoder neural network for landslide susceptibility prediction, Landslides 17 (2020) 217–229, https://doi.org/10.1007/s10346-019-01274-9.

[78] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: high confidence predictions for unrecognizable images, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 427–436, https://doi.org/10.1109/CVPR.2015.7298640.

[79] D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network dissection: quantifying interpretability of deep visual representations, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3319–3327. https://arxiv.org/abs/1704.05796.

[80] L. Perez, J. Wang, The effectiveness of data augmentation in image classification using deep learning, Comput. Vis. Pattern Recognit. (2017) 1–8. arXiv:1712.04621 .

[81] C.W. Zhang, M.Y. Yang, H.J. Zeng, J.P. Wen, Pedestrian detection based on improved LeNet-5 convolutional neural network, J. Algorithms Comput. Technol. 13 (0) (2019) 1–10, https://doi.org/10.1177/1748302619873601.

[82] L. Wang, K. Kawaguchi, P. Wang, Damaged ceiling detection and localization in large-span structures using convolutional neural networks, Autom. Constr. 116 (2020) 103230, https://doi.org/10.1016/j.autcon.2020.103230.

[83] J. Zhang, C. Lu, J. Wang, L. Wang, X. Yue, Concrete cracks detection based on FCN with dilated convolution, Appl. Sci. 9 (2019) 2686, https://doi.org/10.3390/app9132686.

[84] Y. Ren, J. Huang, Z. Hong, W. Lu, J. Yin, L. Zou, et al., Image-based concrete crack detection in tunnels using deep fully convolutional networks, Constr. Build. Mater. 234 (2020) 117367, https://doi.org/10.1016/j.conbuildmat.2019.117367.

[85] K.R. Sri Preethaa, A. Sabari, Intelligent video analysis for enhanced pedestrian detection by hybrid metaheuristic approach, Soft. Comput. 24 (2020) 12303–12311, https://doi.org/10.1007/s00500-020-04674-5.

[86] B. Kim, N. Yuvaraj, K.R. Sri Preethaa, R. Santhosh, A. Sabari, Enhanced pedestrian detection using optimized deep convolution neural network for smart building surveillance, Soft. Comput. 24 (2020) 17081–17092. https://link.springer.com/article/10.1007/s00500-020-04999-1.