Heliyon 10 (2024) e38104

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Deep neural networks for automated damage classification in image-based visual data of reinforced concrete structures

Ching-Lung Fan

Department of Civil Engineering, Republic of China Military Academy, Kaohsiung, Taiwan

ARTICLE INFO

Keywords:
Damage images
Convolutional neural networks
Fully convolutional networks
Computer vision

ABSTRACT

Significant strides in deep learning for image recognition have expanded the potential of visual data in assessing damage to reinforced concrete (RC) structures. Our study proposes an automated technique, merging convolutional neural networks (CNNs) and fully convolutional networks (FCNs), to detect, classify, and segment building damage. These deep networks extract RC damage-related features from high-resolution smartphone images (3264 × 2448 pixels), categorized into two groups: damage (exposed reinforcement and spalled concrete) and undamaged area. With a labeled dataset of 2000 images, fine-tuning of network architecture and hyperparameters ensures effective training and testing. Remarkably, we achieve 98.75 % accuracy in damage classification and 95.98 % in segmentation, without overfitting. Both CNNs and FCNs play crucial roles in extracting features, showcasing the adaptability of deep learning. Our promising results validate the potential of these techniques for inspectors, providing an effective means to assess the severity of identified damage in image-based evaluations.

1. Introduction

Reinforced concrete (RC) structures undergo natural aging, resulting in issues such as exposed reinforcement and spalled concrete, which can compromise overall integrity and structural behavior. As primary materials in modern constructions, RC requires vigilant Structural Health Monitoring (SHM) to ensure civil infrastructure safety. SHM systems have been implemented [1,2], requiring numerous sensors. Analyzing the complexity and heterogeneity of sensor data is time-consuming and labor-intensive; therefore, effective implementation of SHM systems is challenging. The combination of numerical imaging and computer vision techniques has demonstrated considerable potential for assisting in analyzing sensor data, offering new opportunities for automating the assessment of damage in RC structures.

A study investigated the application of image-based damage detection methods in SHM systems for RC structures [3]. Image-based damage detection methods offer several advantages over traditional damage detection methods, including reduced inspection time and cost. Images can be collected using various types of cameras, including security cameras [4], smartphone cameras [5], digital cameras [6], and cameras mounted on unmanned aerial vehicles [7,8]. Cameras are mobile and versatile and enable efficient data collection, and image-based sensors and sensing platforms enable efficient and cost-effective visual inspection [9]. Image data have become increasingly easy to collect, and computer vision and machine learning (ML) techniques have increased the efficiency with which image data can be analyzed. These developments have improved the potential for leveraging image-based approaches in SHM. ML is a class of data-driven algorithms designed to leverage high-dimensional features from perceptual data for damage detection [10]. Deep

E-mail address: p93228001@ntu.edu.tw.

https://doi.org/10.1016/j.heliyon.2024.e38104
Received 10 September 2024; Received in revised form 14 September 2024; Accepted 17 September 2024
Available online 19 September 2024

learning, a modern ML approach, has been used to effectively achieve feature representations in a range of computer vision tasks [11].

Computer vision–based inspection techniques are cost-effective and easy to implement. They do not require physical contact, making them efficient and practical. Additionally, these techniques can be used to interpret data intuitively, which has contributed to their widespread acceptance [12,13]. Computer vision technology involves multidisciplinary techniques, with optical imaging, image processing, artificial intelligence, and ML combined to extract meaningful information from images or videos. Computer vision–based damage inspection techniques are faster, cheaper, and more accurate than traditional techniques are [14]. Computer vision–based damage detection methods are effective; however, opportunities remain for increasing their precision and accuracy. According to Spencer Jr. et al. [15], automatically and reliably transforming image or video data into actionable information by using computer vision technology is challenging. Deep neural networks can be used to improve the accuracy of computer vision–based image analysis, bridging the gap between machine and human vision. Convolutional neural networks (CNNs) are popular deep learning frameworks that have revolutionized computer vision and led to advancements in image recognition in various domains.

CNN possesses the capability to automatically learn hierarchical features from input images, and this architectural framework has become a suitable platform for extracting image features [16]. They have gained popularity in image recognition and shown remarkable capabilities in detecting damage in images. Several studies have demonstrated the effectiveness of CNN models for this purpose [17–19]. CNNs have been applied for damage detection and classification purposes with promising results [20]. They can classify different types of damage in RC structures by using images [21,22]. Moreover, extended CNN models, such as fully convolutional networks (FCNs), have been developed that improve upon the feature extraction abilities of traditional CNN models. FCNs enable semantic segmentation of objects in images.

Computer vision and deep learning have been used extensively in damage detection systems to detect cracks in concrete; however, they have not been extensively used to detect exposed steel reinforcement or concrete spalling. Steel reinforcement can be exposed when concrete cover degrades, and structures with exposed steel reinforcement are considered unsafe. Spalling, on the other hand, refers to the areas where the wall surface is exposed due to flaking and the detachment of large solid or surface-broken material fragments. Studies on the detection and quantification of spalling mainly concentrate on structural elements of civil structures and buildings, as severe spalling on structural components can lead to sudden collapse or failure of the entire structure [23]. The presence of exposed reinforcement and concrete spalling indicates fatigue reactions and serves as an early warning sign for potential structural failure. The integrity and durability of a structure can rapidly degrade, and therefore, monitoring, maintenance, and timely intervention are crucial for preventing further deterioration and extending the life of an RC structure.

When various damages occur in RC structures, a robust detection method can effectively classify the types and extent of damages based on the shape, size, and location of the damage images. Classification and segmentation techniques provide technical support for the rapid detection of surface damages, with both types of inspections offering further detailed information. The present study has developed a robust method for detecting damage in RC structures. By combining computer vision-based deep learning and image processing techniques, the method accurately determines the type and extent of damage. The approach integrates CNN and FCN algorithms to automatically detect exposed reinforcement and concrete spalling.

2. Literature review

Advancements in digital camera, smartphone, and unmanned aerial system technologies have revolutionized the collection of damage images for assessing the condition of civil infrastructure. Unmanned aerial systems are particularly useful for detecting damage in difficult-to-reach areas [24,25]. However, research has yet to determine the most suitable method for classifying images. Classification methods must extract numerous features and carefully select the best feature data set to achieve accurate classification results. In recent decades, advancements in high-resolution imaging technology paved the way for ML techniques, which are now widely used for image classification. Traditional ML algorithms often rely on a large number of labeled samples to train models and achieve high prediction accuracy [26]. Classification algorithms, or classifiers, are used to predict discrete class labels, such as damage categories, by training using sample data [27]. Commonly used ML algorithms for image damage classification include support vector machine [28,29], random forest [30], K-nearest neighbors [31], and K-means [32]. These methods are used to extract damage features from images and then classify the extracted features.

Artificial neural networks (ANNs) are commonly used to detect cracks and other damage in concrete [33,34]. However, the simple architecture of ANNs limits their computational capability, preventing them from accurately capturing local features in images. Traditional ML techniques rely on manual extraction of low-dimensional features; however, this can be time consuming and ineffective because of background noise [35]. Furthermore, images often have uneven regions, and the objects being extracted may be complex; traditional classification and feature extraction algorithms may struggle to detect deep semantic features. Therefore, researchers have turned to other ANN-based methods for damage detection. CNNs have become popular for this purpose [36,37]. CNNs use multiple layers to model features and are applicable for image classification, object detection, and image segmentation.

CNN architectures comprise convolutional layers, pooling layers, and fully connected layers. These components enable networks to learn features for predicting outcomes on the basis of images or object classes. The process involves convolving the input image with filters in the convolutional layer to obtain feature maps with higher-level semantic information. A pooling layer then reduces the dimensionality of these feature maps and identifies more effective features. Finally, a fully connected layer extracts features from the maps and derives predictions through function operations. CNNs offer clear advantages for advanced feature representation and have considerable potential for use in image recognition tasks. For example, Tong et al. [38] used CNN-based patch classifiers to identify cracks and determine their lengths. Gao and Mosalam [39] used the VGG-16 network to train a classifier to identify spalling in concrete. Santos et al. [40] used an AlexNet-based CNN and transfer learning to detect exposed reinforcement.

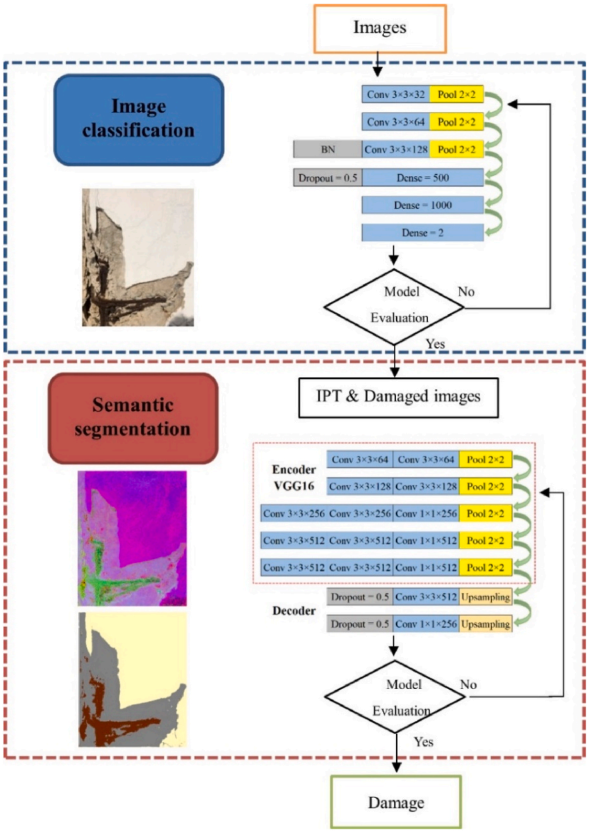


Fig. 1. Image classification and damage segmentation workflow.

CNNs now widely serve as the foundation for recognition and segmentation networks. Various CNN-based models have been developed. Long et al. [41] developed an FCN that efficiently partitions objects of the same category, providing a crucial basis for semantic segmentation. FCNs are end-to-end methods that combine pixel-level classification with an upsampling and downsampling architecture in the encoder–decoder framework. FCNs extend traditional CNNs by transforming class predictions into semantic segmentation images. In the pixel-level classification approach, a class is assigned to each pixel in an image, which effectively separates features from the background to enable extraction of specific characteristics [42]. Other models commonly used for image segmentation include U-Net [43], SegNet [44], and the various models from the DeepLab series [45].

FCNs have been proven effective in extracting specific objects and have been practically applied for semantic segmentation of damage images. A range of FCN-based methods have been developed that can detect damage of various types. Yang et al. [46] trained an FCN with multiple types of cracks to semantically identify and segment cracks of different scales. Dung and Anh [47] developed a deep FCN-based method for detecting cracks in concrete structures, achieving an average accuracy of approximately 90 %. Li et al. [42] proposed a SegNet-based FCN for detecting different forms of damage (e.g., cracks, spalling, efflorescence, and holes) in concrete structures. Rubio et al. [48] used an FCN based on the VGG-16 network as a backbone to detect delamination and exposed steel rebars, achieving average detection accuracies of 89.7 % and 78.4 %, respectively. Wang and Cheng [49] integrated a deep CNN with dense conditional random field techniques to improve semantic segmentation accuracy. They assigned defect labels to each pixel to facilitate automatic severity assessment of sewer pipeline defects and were able to obtain information on defect types, locations, and geometries. Zhang et al. [50] designed an FCN for detecting three types of concrete surface defects: cracks, spalling, and exposed steel rebars, achieving semantic segmentation accuracy of 75.5 %. Fu et al. [51] developed a crack detection method based on an improved DeepLabv3+ semantic segmentation algorithm, achieving an average intersection ratio of 82.37 %. Finally, Wang et al. [52] constructed a data set containing 2446 manually annotated images to train five deep networks for semantic segmentation performance evaluation purposes. Xu et al. [53] proposed a semantic segmentation method based on an improved DeepLabv3+ model, which integrates a MobileNetV2 backbone network and an enhanced ASPP module, for the identification of bridge and structural damage.

CNNs and FCNs have been employed in a wide variety of computer vision tasks, including image segmentation and object recognition tasks, in medicine and autonomous driving. The detection of damage in RC structures is a critical task in civil engineering. Deep learning methods have performed well when used for damage detection in civil engineering [54,55], but several challenges must be overcome in the application of deep learning for image-based damage detection in RC structures. This study used CNNs to extract damage-related features and used image processing techniques to enhance the segmentation capabilities of FCNs, allowing for the type and scope of damage to be precisely identified. This novel application of deep learning in civil engineering is useful for the

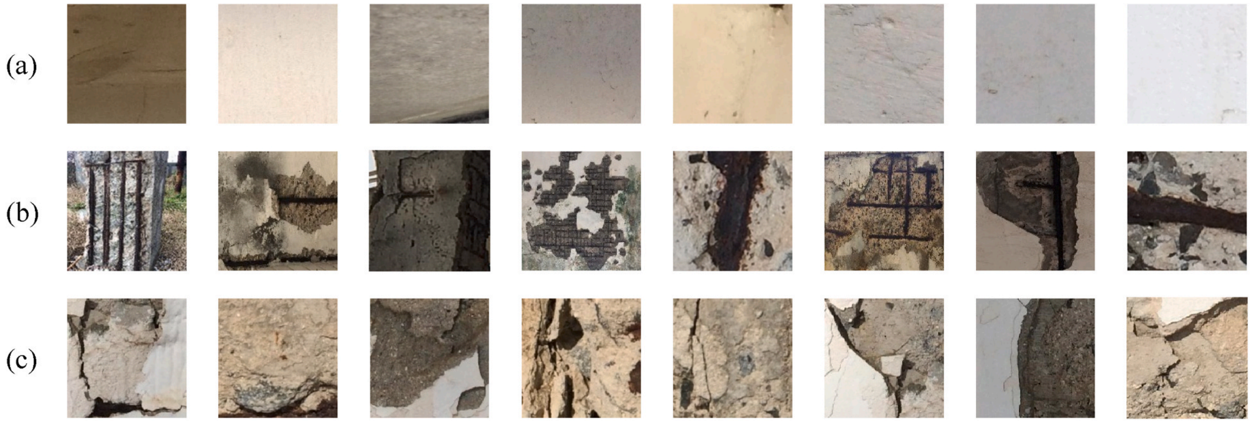


Fig. 2. Image dataset of undamaged and damaged RC: (a) Undamaged areas, (b) Exposed reinforcement, (c) Spalled concrete.

identification of exposed reinforcement and concrete spalling damage.

Large and high-quality datasets are key in image classification. Larger datasets yield greater model accuracy and generalizability given their variety, and high-quality datasets are key to meaningful feature extraction. Models for detecting damage in RC structures must be robust to variations in lighting, shadow, and surface conditions [56]; lighting conditions affect overall visibility, shadows can obscure damage, and rough surfaces introduce noise. Models must also be capable of distinguishing between different types of damage, such as cracks, spalling, efflorescence, and exposed reinforcement, on the basis of their subtle but important visual characteristics. Models are thus difficult to train and test given these demands. Various deep learning methods and other techniques have been developed to address these specific challenges. According to Cha et al. [57], combining deep learning models and image processing techniques can further enhance model performance in RC structure damage detection.

3. Research method

The hyperparameters of the proposed model were adjusted to ensure effective feature extraction and segmentation. The hyperparameters that were adjusted included convolutional and pooling layer quantities, convolutional kernel sizes and strides, activation functions, and image data augmentation. Images were classified by a CNN and then input to an FCN that identified exposed reinforcement and spalled concrete in the backgrounds of the images. A primary advantage of FCN-based damage detection is its integration of identification and localization functionalities, which provide more comprehensive results. Additionally, FCN-based damage detection eliminates the need for applying the sliding window technique, as it enables the processing of the entire image without dividing it into smaller patches [58].

In the present study, damage detection involved two steps: image classification and semantic segmentation. In the first step: a CNN was trained to classify images, and then the model was tested. The parameters of the CNN were fine-tuned until the CNN could distinguish between damaged and undamaged images in cluttered and unstructured data. This allowed for identification of specific forms of damage, such as exposed reinforcement and spalled concrete. The second step involved semantic segmentation. The results from the classification step were processed (image processing technology, IPT) and input into an FCN to determine the scope of damage. The hyperparameters of the FCN were fine-tuned, and the model was trained accordingly. Ultimately, this enabled precise segmentation and localization of damage (Fig. 1).

3.1. Image data set

Model creation involves construction of a data set and training and validation of the model. In RC structures, common damage includes exposed steel rebars and spalling, which have distinct characteristics in terms of color, shape, and area. In the present study, a diverse collection of images capturing various forms of damage in RC structures was used as the data set. The data set contained images of exposed reinforcement and spalled concrete, representing common forms of damage found in civil infrastructure.

RC damage detection is a key part of SHM. The diversity and scale of data used for model training must be considered when constructing a dataset to ensure the accuracy and generalization ability of the trained model. The frequency and effect of different damage types vary; therefore, dataset construction should cover various damage types and scenarios to ensure that the characteristics and diversity of damage are fully represented in the dataset. This approach not only improves the model's damage recognition ability but also enhances its adaptability to different environments. The present study expanded the database to include images showing exposed reinforcement and spalled concrete in RC structures. High-resolution images (3264 × 2448 pixels) were taken using a smartphone under a variety of lighting conditions. To ensure dataset accuracy and representativeness, a rigorous data collection process was implemented. Images were taken at several locations and of different environments, including indoor and outdoor environments. Special attention was given to taking images of damage under different lighting conditions, at different angles, and from a range of distances.

All labeled images were resized to 256 × 256 pixels to improve computational efficiency. Images were cropped to represent either exposed reinforcement, spalled concrete, or undamaged areas. This decision considered the scale of the existing dataset and the need for real-time processing in specific application scenarios. In the context of limited computational resources, using this resolution facilitates efficient model training and inference. To maximize the retention of key features and avoid information loss, we performed precise cropping during image adjustment, ensuring the accuracy of damage locations and minimizing potential errors introduced during cropping that could negatively impact model training. Pixel-level labels indicating the boundaries of exposed reinforcement and spalled concrete were added to the images. Such pixel-level annotation is essential for FCNs to accurately identify and delineate damage features during semantic segmentation. This approach allows us to maintain the representativeness of the image content and the reliability of the model while achieving efficient computation. Additionally, we further enhance the model's adaptability to different environments and damage types through techniques such as data augmentation, ensuring high detection accuracy across a broader range of application scenarios. This study used a dataset containing 2000 labeled images that showed both undamaged and damaged (exposed reinforcement and spalled concrete) RC structures (Fig. 2).

The present study's image data set was divided into training, validation, and test sets. The training set was used for model training; the validation set was used for parameter tuning; and the test set was used for evaluating the model's performance. In total, 80 % of the images were used to train and validate the model, and 20 % of the images were used to test the model. Specifically, 1000 images were randomly selected from the pool of 2000 images and used for training; 600 of the remaining images were used for validating; and the final 400 images were used for testing. The test set was used to evaluate the model's ability to classify exposed reinforcement and spalled concrete and to assess the effectiveness of the classification results in guiding FCN model segmentation.

This study utilized random sampling to divide the dataset into training, validation, and testing subsets. During the dataset partitioning process, we followed an 80:20 ratio, which is a common practice in deep learning research, aimed at ensuring a reasonable proportion between the training set and the validation/testing sets to enhance the model's generalization capability. The quantity of training samples, class balance, and the completeness of different damage categories significantly affect accuracy [59]. To ensure representativeness and balance among the subsets, we made sure that each subset contained various types of damage images. This partitioning method aids in the model's generalization during training and facilitates performance evaluation during validation and testing. When selecting images, we adhered to the following criteria: firstly, diversity was crucial, ensuring that each subset included a variety of damage types to comprehensively reflect various real-world conditions, including different damage locations, severities, and environmental conditions. Secondly, to ensure balance, we aimed to maintain a balance of damage types within each subset to prevent bias in model evaluation. Finally, representativeness was also a key factor; the selected images covered all common types of damage in the dataset, ensuring that each subset effectively represented the main features of the entire dataset.

3.2. Deep learning technology

CNNs comprise convolutional layers, pooling layers, and fully connected layers. Convolution involves using a fixed-size sliding window (filter) that sequentially moves across an input image, convolving with the receptive field's tensor via element-wise multiplication and summation with a bias term [60]. This process generates multiple feature maps for learning image features. Pooling reduces image size to control computational costs and prevent overfitting while retaining essential features. Fully connected layers compute class probabilities based on feature maps using function operations, typically applying the Softmax function for multiclass classification. CNNs handle tasks like image recognition, feature detection, and classification. Performance optimization involves dataset selection and adjusting hyperparameters such as filter size, layer count, depth, dropout rates, and batch normalization.

FCNs, derived from CNNs, specialize in semantic segmentation and localization compared to CNNs, which are limited to image classification. Both models utilize convolutional and pooling layers; however, FCNs replace fully connected layers with convolutional layers for dense pixel-wise prediction and semantic segmentation. Traditional CNNs conclude with fully connected layers for one-dimensional predictions, lacking spatial information needed for image segmentation tasks requiring pixel-level classification. FCNs address this by employing deconvolutional (or upsampling) layers to expand feature maps from the final convolutional layer to match the original image size, enabling pixel-level predictions. This capability allows FCNs to predict class labels for each pixel, effectively delineating regions and achieving detailed segmentation. The approach generates a segmentation map with class labels for precise object detection and classification by assigning pixels to specific categories. FCNs integrate encoder-decoder architectures using convolutions for feature extraction and deconvolutions for spatial restoration, facilitating end-to-end semantic segmentation with accurate pixel-level classification.

In this study, we chose to use FCNs instead of more complex models such as Transformer models (e.g., Swin Transformer [61], Segformer [62]), encoder-decoder architectures (e.g., DeepLabv3+ [63], Unified Perceptual Parsing (UPP) [64]), hybrid methods (e.g., Dual Path Network (DPNet) [65]), and weakly-supervised and semi-supervised models (e.g., Wegformer [66]), primarily due to its multiple advantages in specific application scenarios. Firstly, FCNs offer lower computational complexity and memory usage compared to these advanced models, providing significant efficiency advantages when real-time processing or deployment on resource-constrained devices is required. Secondly, FCNs are a well-established architecture that has undergone years of optimization and extensive application, offering rich practical experience and tool support, which facilitates rapid deployment. Additionally, FCNs have proven effective in extracting local features in many practical applications, particularly in detecting low-level features such as textures, which is crucial for image segmentation tasks related to RC damage. While Transformer models have unique advantages in capturing global features, their complexity and high computational overhead may not be suitable for certain application scenarios. Finally, the relatively simple structure of FCNs results in a more stable training process and greater model interpretability, making it easier to analyze and optimize. These factors make FCNs a more appropriate choice for this study.

The objective of this study is to propose an effective solution for RC structural damage detection by combining validated CNN and FCN architectures for image classification and semantic segmentation. These methods have demonstrated stability and effectiveness in practical applications. Although these approaches are not novel, we emphasize their effectiveness and applicability in specific scenarios, particularly in environments with resource constraints and the need for efficient processing, where FCNs continue to offer unique advantages.

3.3. Image processing technology

Environmental conditions, such as lighting, weather, and background colors, affect the performance of damage detection models that rely on color information. Lighting conditions strongly influence the recognition of spectral features—different types of damage can appear spectrally similar in visible light due to variations in illumination and the presence of shadows. In RC structures, two or more different types of damage (coupled damage) may coexist and interact with each other. Features of coupled damage are not easily distinguishable in red, green, and blue (RGB) images, and using only the three color channels of RGB images may result in detection errors even with deep learning methods. Because the perceived colors in RGB images are affected by lighting conditions, converting the RGB color space to a different color space can help compensate for changes in illumination. Jahanshahi and Masri [67] noted that damage detection models that use the RGB color space do not achieve the most accurate results. Image processing techniques should be used to compensate for the disadvantages of RGB images. The conversion of RGB images into a non-RGB color space can enhance classification performance.

The hue, saturation, and value (HSV) color space facilitates easier color segmentation compared with the RGB color space because HSV components can be processed independently, allowing for segmentation of objects with specific hues while disregarding changes in value. Consequently, HSV models are widely used for color segmentation and detection tasks in image processing and computer vision. Color separation (and therefore image segmentation) is easier in the HSV space than in the RGB space. HSV transformations can serve as an alternative for shadow correction in image equalization algorithms and can yield superior results (especially in areas still affected by shadows) and validating independent analysis as effective [7]. Effective RC damage classification tasks must account for the diversity of RC surfaces and the effects of environmental conditions. HSV-based image processing techniques enhance the recognition capabilities of deep learning models and enable more accurate damage segmentation and assessment.

4. Experimental results and analysis

Increasing the size of a training data set, although often challenging, is generally beneficial for improving detection performance. Using a small training data set is not ideal. Detection models often have deeper structures and more parameters, which may lead to redundancy when identifying damage. Determining the appropriate depth of a model is challenging. To address these challenges, the present study developed a CNN architecture consisting of three convolutional layers, three pooling layers, and one fully connected layer (three dense layers), with a moderate number of sliding windows in each layer. This deep architecture effectively extracts complex features from damage images. The CNN has a different structure and fewer layers than do typical large neural networks with multiple objectives and large data sets for multiclass classification. Therefore, the CNN demonstrates high efficiency with respect to training time while remaining well-suited for damage identification tasks, providing a balance between accuracy and computational efficiency.

4.1. CNN architecture and hyperparameters

The CNN used in this study is a supervised learning deep neural network. The architecture of the network, the number and size of sliding windows, affects the accuracy of predictions and computational efficiency. The architecture and hyperparameters used are presented in Appendix. Each input image is cropped to a size of $150 \times 150 \times 3$, representing the height, width, and RGB channels. (1) All sliding windows have a size of 3×3 in convolutional layers. The number of filters starts at 32 and doubles in each subsequent convolutional layer until reaching 128 filters. The ReLU activation function is used for all convolutions. A stride of 1 is applied to all convolutions, with zero-padding ensuring the output size matches the input size. Max pooling with a size of 2×2 is used for image reduction. (2) The fully connected layer consists of a flatten layer followed by two hidden layers with 500 and 1000 neurons, respectively. The ReLU activation function is applied to these layers. The output layer has two neurons with Softmax activation function for predicting the presence of damage features. (3) Data augmentation techniques are applied to increase the size of the data set and reduce the risk of overfitting. According to Dung et al. [68], data augmentation can improve classification accuracy by up to 5 %. Dropout was also used to prevent overfitting. Batch normalization and Adam optimization were implemented for faster network convergence and training.

The process of passing all images in the data set through the neural network for forward and backward propagation is referred to as batch processing. During this learning process, the model's predictions are compared with actual values. The weights in the network are optimized using gradient descent, with the goal of minimizing the error on the training set and achieving convergence in processed batches. The validation set is used to ensure the model does not overfit solely to the training set and to perform necessary hyperparameter tuning. Hyperparameters are adjustable parameters that can be set manually to ensure the best possible output. The test set comprises data that has not been used during training set and validation set. After the CNN has generated satisfactory results on the training and validation sets, it is evaluated on new data in the testing set. A good and random testing set is crucial because it enables an

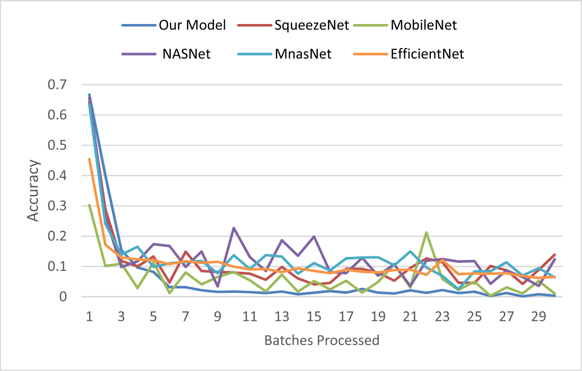


Fig. 3. Convergence process of loss on the training set for six models.

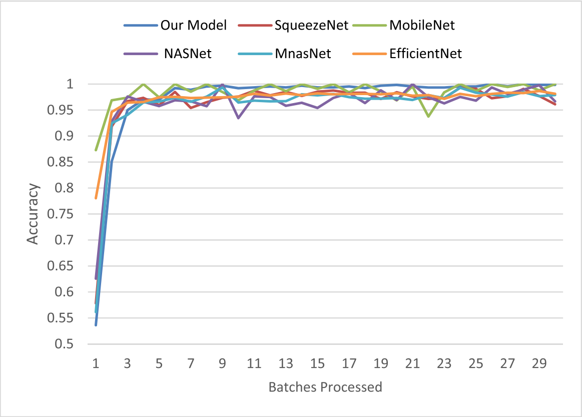


Fig. 4. Convergence process of accuracy on the training set for six models.

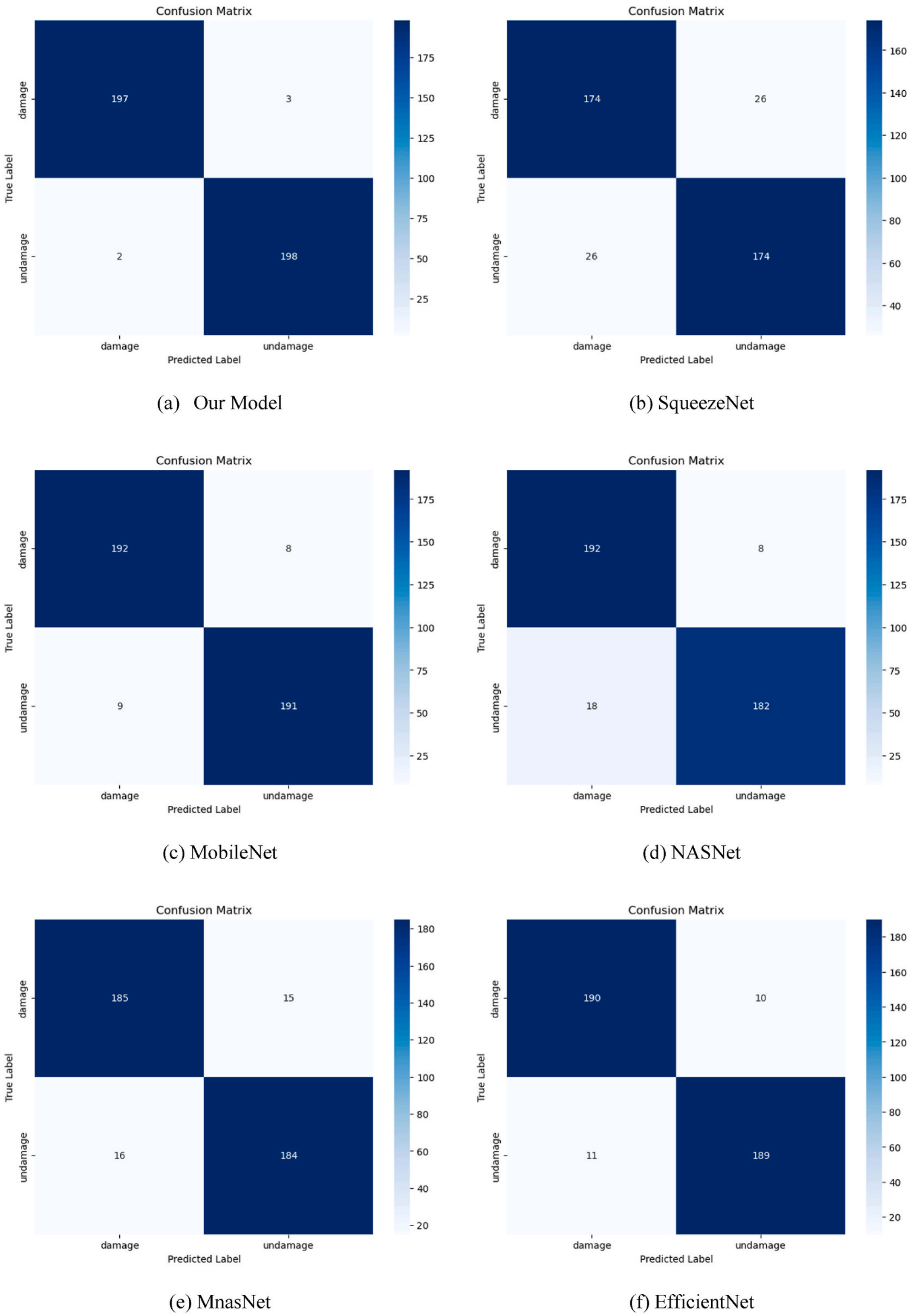
Table 1
Performance evaluation of six lightweight models for RC damage classification.

Model	Accuracy	Precision	Recall	F1 score
Our Model	98.75	98.50	99.0	98.74
SqueezeNet [69]	87.0	87.0	87.0	87.0
MobileNet [70]	95.75	96.0	95.52	95.76
NASNet [71]	93.50	96.0	91.43	93.66
MnasNet [72]	92.25	92.50	92.04	92.26
EfficientNet [73]	94.75	95.0	94.53	94.76

unbiased evaluation of the CNN model's performance when faced with different image features.

Selecting a deep learning classifier for a specific task is challenging due to the vast array of available deep learning methods. Due to this abundance, the relative advantage of a single model is difficult to determine. Comparative studies of different models are often necessary to identify the best classification approach for a specific research task. To demonstrate the effectiveness of our approach in RC damage classification, additional experiments were conducted using several widely recognized and lightweight models, namely SqueezeNet [69], MobileNet [70], NASNet [71], MnasNet [72], and EfficientNet [73]. The relationship between the loss and batches processed in the six models is presented in Fig. 3. The training loss shows a clear decreasing trend, indicating the effectiveness of the six models. Specifically, our model begins to converge around the fifth epoch, with a training loss of 0.082. By the 30th epoch, the loss reaches its optimal value of 0.004 for training. The accuracy of the six models during the training process is depicted in Fig. 4. Accuracy increases with the number of batches processed. Our model starts to converge around the fifth epoch and achieves a training accuracy of over 99 %. On the test dataset, the model achieves an accuracy of 98.75 %.

The results of our performance analysis are shown in Table 1. Our model achieved accuracy, precision, recall, and F1 score results of 98.75 %, 98.5 %, 99.0 %, and 98.74 %, respectively. Our model outperformed the other models. We also visualized the performance of the six models by using confusion matrix diagrams. On a test set of 400 images, our model correctly predicted 197 damaged and 198 undamaged images (Fig. 5a) and performed better than all other models. MobileNet came in at a close second and correctly predicted



(a) Our Model

(b) SqueezeNet

(c) MobileNet

(d) NASNet

(e) MnasNet

(f) EfficientNet

Fig. 5. Confusion matrices of six lightweight models for RC damage and non-damage classification.

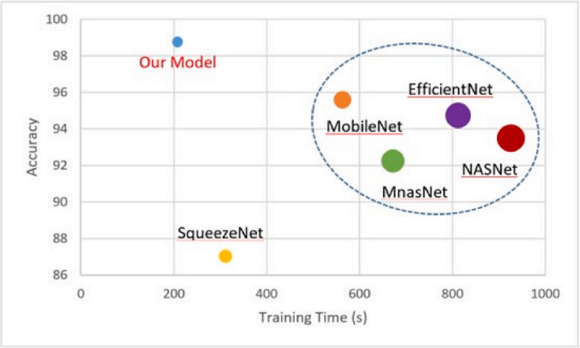


Fig. 6. Accuracy and training time of six lightweight models.

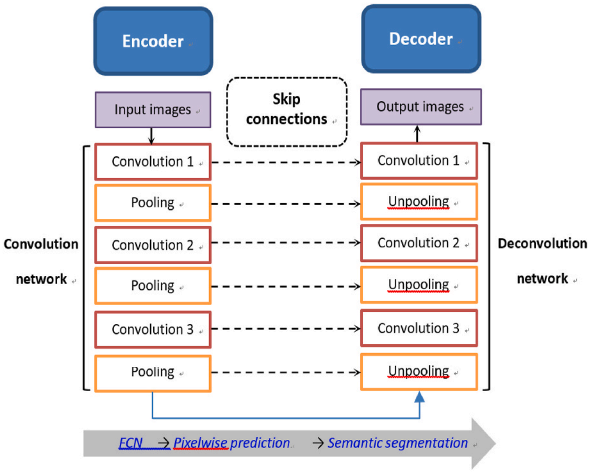


Fig. 7. Fully convolutional network architecture.

192 damaged and 191 undamaged images (Fig. 5c). NASNet, MnasNet, and EfficientNet exhibited similar performance (Fig. 5d–f). SqueezeNet performed the worst (Fig. 5b). These results not only illustrate each model's ability to classify damaged and undamaged images but also highlight the superiority of our model in terms of accuracy and performance.

The accuracy and training time of each model is shown in Fig. 6. In the figure, the size of each dot represents the complexity of the corresponding model's network structure, with more complex structures requiring the updating of more parameters, thereby increasing training time. Our model (blue dot), featuring a network structure with three convolutional layers, was the simplest of the six models and had an average training time of 207 s. The assessment was conducted on an NVIDIA Quadro RTX 6000 24 GB GPU. Furthermore, our model achieved the highest accuracy of 98.75 % among the six models despite its simplicity. NASNet had a training time of 927 s, the longest among the six models. MobileNet, NASNet, MnasNet, and EfficientNet had similar performance, with training times ranging from 10 to 15 min and accuracies ranging from 92 % to 96 %. Additionally, although SqueezeNet's training time was approximately 5 min, its accuracy was only 87 %. Due to the large number of parameters in most deep learning models, extended training times are a common challenge. Our model, being a lightweight model, not only had a short training time but was also exceptionally accurate.

This study primarily focuses on the automation of damage classification, with the choice of computationally efficient architectures driven by the needs of practical application scenarios. In certain cases, especially those requiring rapid on-site decision-making, such as post-disaster assessments or real-time structural health monitoring, damage classification needs to be performed in near real-time. Furthermore, when deploying algorithms on resource-constrained devices such as drones or portable equipment, lightweight model architectures can significantly reduce hardware requirements, thereby lowering costs and enhancing usability.

4.2. Architecture and hyperparameters of FCN

The FCN in this study is based on an encoder–decoder architecture with an encoder path that uses convolutions and pooling to extract damage features. High-level extracted features are restored through the decoder path and input to the pixel-wise classification layer. Skip connections are used to establish connections between high-level and low-level features, enabling the fusion of multiscale

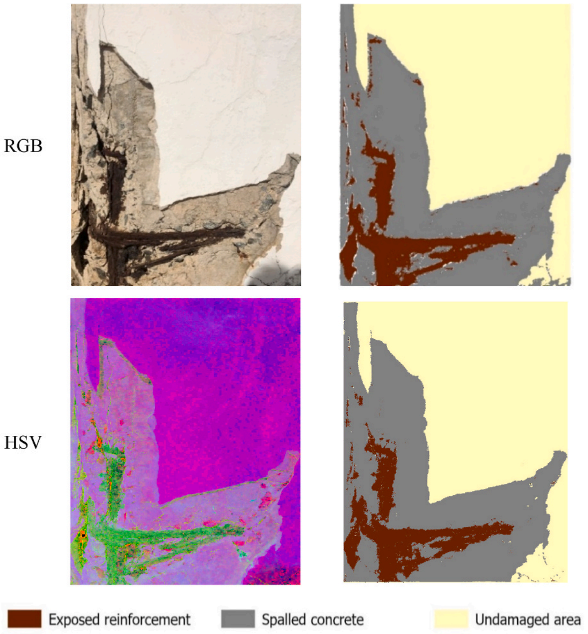


Fig. 8. FCN segmentation results for three objects in RGB and HSV images.

Table 2
FCN performance for RGB image.

Metrics (%)	Exposed reinforcement	Spalled concrete	Undamaged area
Precision	88.32	74.98	98.61
Recall	86.15	97.57	96.97
F1 score	87.22	84.80	97.78
Accuracy	89.95		

Table 3
FCN performance for HSV image.

Metrics (%)	Exposed reinforcement	Spalled concrete	Undamaged area
Precision	92.11	91.10	99.58
Recall	96.79	91.96	99.28
F1 score	94.39	91.53	99.43
Accuracy	95.98		

features to enhance the accuracy of segmentation. Fig. 7 illustrates the FCN model. The model has three convolutional layers and three deconvolutional layers. (1) First, an image is input into the encoder path, which consists of three convolutional layers. Max pooling is then employed to downsample the image, reducing the size of the input tensor. The kernel size for all pooling layers is 2 × 2, and the stride is 1. (2) High-level features are passed to the decoder path, which consists of three deconvolutional layers. In the decoder path, each deconvolutional layer performs upsampling operations. Additionally, the features extracted using downsampling are directly passed to the upsampling process through concatenation. The upsampling operation is mainly responsible for accurately restoring and localizing the features. (3) Finally, a 2 × 2 kernel is used for convolution with three outputs to predict scores for three categories of damage, namely, exposed reinforcement, spalled concrete, and undamaged regions, at the corresponding positions in the previous output. Subsequently, the previous outputs are upsampled to produce pixel-dense output by using the deconvolutional layers. The probability feature representation from the output of the decoder is fed into the Softmax function, enabling independent pixel-wise classification for each category. In each deconvolutional layer, upsampling is performed with a stride of 1 to double the previous output size. The image output by the FCN is the same size as the input image.

The images classified by the CNN are input to the FCN for the purpose of damage segmentation. Through hyperparameter fine-tuning and training processes, the FCN is able to extract features related to exposed reinforcement, spalled concrete, and undamaged areas, thereby effectively performing the task of image segmentation. The results of RGB and HSV image segmentation are presented in Fig. 8. The model demonstrated good segmentation capability for the three types of objects. Table 2 presents the performance metrics of the FCN for exposed reinforcement, spalled concrete, and undamaged areas in RGB images. The accuracy of our

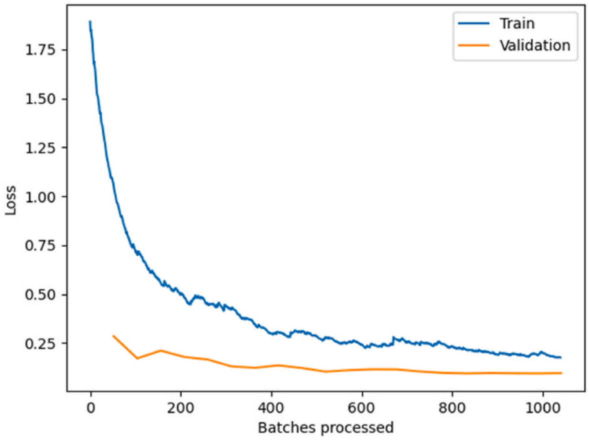


Fig. 9. FCN training and validation losses for RGB images.

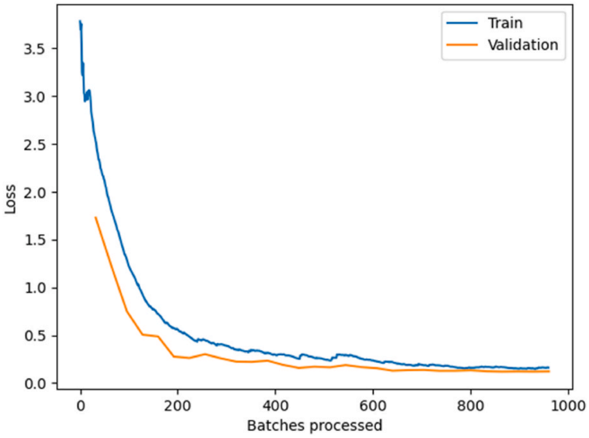


Fig. 10. FCN training and validation losses for HSV images.

model was 89.95 %. Additionally, Table 3 shows the performance metrics of the FCN model for exposed reinforcement, spalled concrete, and undamaged areas in HSV images, with our model achieving an accuracy of 95.98 %. For the exposed reinforcement class, the values for precision, recall, and F1 score were 92.11 %, 96.79 %, and 94.39 %, respectively. For the spalled concrete class, the values for precision, recall, and F1 score were 91.10 %, 91.96 %, and 91.53 %, respectively. These results indicate that the FCN is capable of accurately segmenting the shapes and extent of exposed reinforcement and spalled concrete in HSV images.

The loss function is a metric for evaluating the deviation between the predicted results and the ground truth. A larger deviation between the predicted results and the ground truth leads to a higher loss value. The objective of model training is to minimize the loss value. Dice loss is a function commonly used in semantic segmentation processes. In the RGB images, the validation dice loss was close to the training dice loss, indicating that FCN did not exhibit overfitting during the training process (Fig. 9). Both the validation and the training dice loss were low, at 0.072 and 0.28, respectively. Additionally, in HSV images, the training and validation curves of FCN were nearly identical, with training and validation final losses being 0.16 and 0.12, respectively, indicating closer performance alignment (Fig. 10).

FCNs segment objects of interest and can handle input images of arbitrary sizes, providing class predictions for each pixel. However, pixel-level classification is computationally expensive. The FCN developed in the present study was fine-tuned to be efficient. The model accurately segmented exposed reinforcement and spalled concrete in RGB images (Fig. 11). Nevertheless, RGB values are influenced by lighting conditions, meaning that changes in the color or intensity of the light source can alter the RGB channels, leading to color misjudgment. In color recognition tasks, especially in unstable outdoor lighting environments, the HSV color model is highly useful. Among the parameters constituting the HSV color space, hue remains more stable under various light intensities compared with saturation and value. Therefore, using HSV images for hue-based recognition can aid in segmenting damaged and undamaged areas in images (Fig. 12).

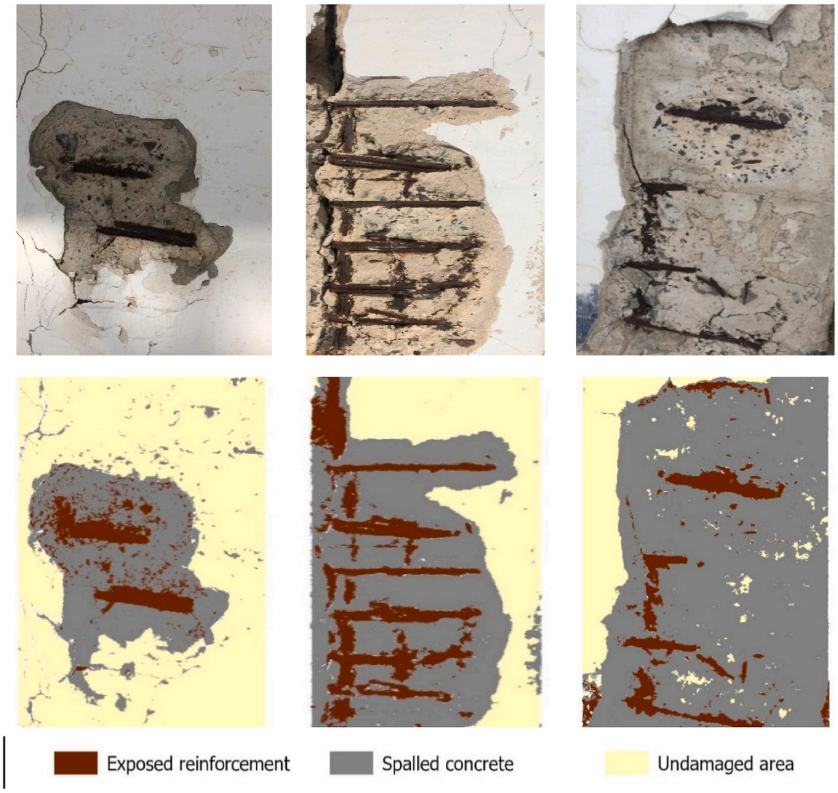


Fig. 11. The segmentation results of FCN for exposed reinforcement and spalled concrete in RGB images.

4.3. Discuss

Various challenges, including dataset biases, overfitting, and computational efficiency problems, were encountered during the conduct of this study. These challenges were addressed using a variety of strategies as detailed in the following paragraphs.

First, dataset bias is a common challenge in machine learning research. Initially, our dataset consisted of only 500 images. Small samples limit the model's ability to classify different damage types and may not adequately represent the diversity of damage characteristics in the real world, particularly coupled damage or previously unseen damage types. To address these problems, we expanded the dataset to include 2000 images to cover a broader range of damage scenarios and conditions. We also ensured that the expanded dataset included images taken in different locations and under different environmental conditions (e.g., varying lighting conditions and shooting angles), thereby reducing biases toward specific types of damage or environments. Such expansion enhanced the proposed model's applicability and accuracy. Additionally, by collecting images under these conditions, we evaluated the model's performance in real-world scenarios. The experimental results demonstrate that the model maintains strong recognition capability even under these conditions.

Second, overfitting is a notable problem in deep learning, especially on small datasets. Overfitting can result in the model performing well on training data but poorly on unseen test data, thereby reducing its practical utility. To mitigate overfitting, we employed data augmentation and regularization strategies. Various techniques such as rotation, flipping, and scaling were applied to increase the diversity of the training dataset to increase model robustness. Techniques such as dropout and batch normalization were also applied to keep the model lean and to minimize fitting noise in the training data. These techniques effectively reduced the risk of overfitting.

Finally, the computational efficiency of a model dictates its range of practical applicability, and we thus chose a lightweight model architecture and used a GPU to accelerate computation. We ran the model on an NVIDIA Quadro RTX 6000 24 GB GPU. Our model ran approximately 11 times faster on this system than it did on a traditional CPU. GPUs facilitate model training with large-scale data within a reasonable timeframe, accelerating model development and subsequent deployment.

In conclusion, through strategies such as dataset expansion, data augmentation, regularization techniques, and GPU acceleration, we successfully addressed challenges related to dataset biases, overfitting, and computational efficiency. These strategies not only improved the accuracy and generalization ability of our model but also significantly enhanced computational efficiency, enabling our model to be better applied to practical damage detection tasks.

In this study, our primary focus was to propose and validate a CNN-based image classification and semantic segmentation model for specific damage detection tasks. The emphasis was on developing and optimizing the model to achieve the best classification and

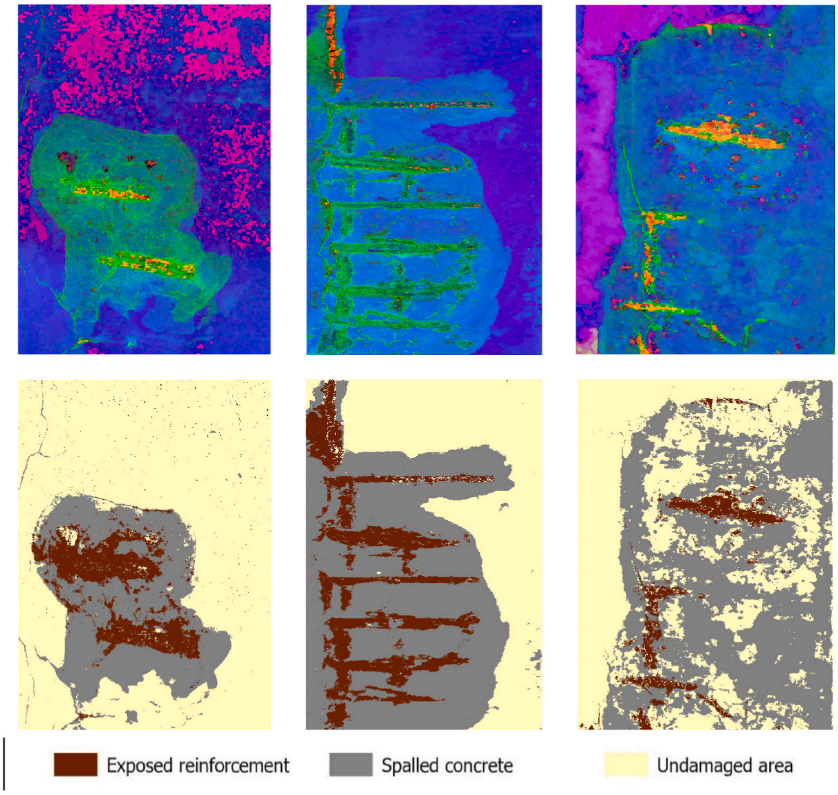


Fig. 12. The segmentation results of FCN for exposed reinforcement and spalled concrete in HSV images.

segmentation accuracy, which involved extensive hyperparameter tuning. However, to further quantify the contribution of each component to the model's performance, future work will include a systematic ablation study. This study will evaluate the importance of each component by systematically removing or modifying model components, guiding model simplification, and future design. An ablation study will help validate design assumptions and enhance the model's interpretability and scientific rigor. In the next phase of work, a comprehensive ablation study will be conducted to systematically assess the contributions of different model components, including network architecture analysis, optimizer and learning rate evaluation, and loss function selection, providing additional empirical support for future research.

5. Conclusion

Advancements in deep learning algorithms have enabled image-based damage detection in civil engineering. CNNs and FCNs are two popular methods for using convolutional layers to identify key features in images, and they are effective at classifying and segmenting damaged areas. The present study proposed a two-step deep learning approach that integrated a CNN and an FCN for damage detection in RC structures. A network architecture was designed, model hyperparameters were fine-tuned, and a model was trained and tested. The model demonstrated high accuracy in damage detection and effectively segmented damaged areas, effectively identifying the size and extent of damage.

Damage assessment depends not only on the use of effective detection techniques but also on the collection of images and available information about damaged areas. This study used a data set comprising 2000 smartphone-captured images with a resolution of 3264 × 2448 pixels. The data set included images containing exposed reinforcement, spalled concrete, and undamaged concrete, with corresponding labels. Classical networks with a large number of trainable parameters are prone to overfitting and may struggle to achieve high recognition accuracy when small data sets are used. However, by constructing an image data set and fine-tuning hyperparameters, the present study achieved effective automatic damage detection. In the first step of the study, training, validation, and testing resulted in an accuracy exceeding 99.0 % for damage classification. In the second step, an overall accuracy of 95.89 % for damage segmentation was achieved.

The impact of original image resolution on segmentation performance and damage classification is a topic worth further investigation. In future research, we plan to explore the feasibility of designing deep learning systems with varying image resolutions and analyze the effects of image resolution on model performance. This will help us improve classification accuracy in a broader range of application scenarios and optimize the model's adaptability to different resolutions. Additionally, models that can detect other forms of damage under various conditions will be developed. We will consider collaborations with other researchers and institutions to access larger datasets and explore the possibility of crowdsourcing efforts to efficiently collect annotated images.

Data availability statement

Request on corresponding authors.

CRediT authorship contribution statement

Ching-Lung Fan: Writing – original draft, Software, Methodology, 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.

Acknowledgment

The authors would like to thank associate professor Yu-Jen Chung, Department of Marine Science, ROC Naval Academy, for providing a GPU computer, which enabled this study to perform image detection tasks quickly.

Appendix. The code and hyperparameters of the CNN algorithm in this study

Algorithm

```
model = models.Sequential
model.add (layers.Conv2D (32, (3, 3), activation = ’relu’, input_shape=(150, 150, 3)))
model.add (layers.MaxPooling2D ((2, 2)))
model.add (layers.Conv2D (64, (3, 3), activation = ’relu’))
model.add (layers.MaxPooling2D ((2, 2)))
model.add (layers.Conv2D (128, (3, 3), activation = ’relu’))
model.add (layers.MaxPooling2D ((2, 2)))
model.add (layers.BatchNormalization)
model.add (layers.Flatten)
model.add (layers.Dropout (0.5))
model.add (layers.Dense (500, activation = ’relu’))
model.add (layers.Dense (1000, activation = ’relu’))
model.add (layers.Dense (2, activation = ’softmax’))
from keras import optimizers
model.compile (loss = ’categorical_crossentropy’, optimizer = optimizers.adam (lr = 1e-4),
    metrics = [’accuracy’])
from keras.preprocessing.image import ImageDataGenerator train_datagen = ImageDataGenerator
    (rescale = 1./255, rotation_range = 40, width_shift_range = 0.2, height_shift_range = 0.2,
    shear_range = 0.2, zoom_range = 0.2, horizontal_flip = True)
history = model.fit_generator (train_generator, steps_per_epoch = 20, epochs = 30, validation_data
    = validation_generator, validation_steps = 20)
```

References

[1] G. Gui, H. Pan, Z. Lin, Y. Li, Z. Yuan, Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection, KSCE J. Civ. Eng. 21 (2) (2017) 523–534, https://doi.org/10.1007/s12205-017-1518-5.

[2] M. Kurata, J. Kim, J.P. Lynch, G.W. van der Linden, H. Sedarat, P. Hipley, L.H. Sheng, Internet-enabled wireless structural monitoring systems: development and permanent deployment at the New Carquinez Suspension Bridge, J. Struct. Eng. 139 (10) (2013) 1688–1702, https://doi.org/10.1061/(ASCE)ST.1943-541X.0000609.

[3] H. Bae, K. Jang, Y.K. An, Deep super resolution crack network (SrcNet) for improving computer vision–based automated crack detectability in in situ bridges, Struct. Health Monit. 20 (4) (2021) 1428–1442, https://doi.org/10.1177/1475921720917227.

[4] Q. Fang, H. Li, X. Luo, L. Ding, H. Luo, T.M. Rose, W. An, Detecting non-hardhat-use by a deep learning method from far-field surveillance videos, Autom. Constr. 85 (2018) 1–9, https://doi.org/10.1016/j.autcon.2017.09.018.

[5] Q. Mei, M. Gül, Multi-level feature fusion in densely connected deep-learning architecture and depth-first search for crack segmentation on images collected with smartphones, Struct. Health Monit. 19 (6) (2020) 1726–1744, https://doi.org/10.1177/1475921719896813.

[6] Y. Xu, Y. Bao, J. Chen, W. 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 (3) (2019) 653–674, https://doi.org/10.1177/1475921718764873.

[7] E. Garilli, N. Bruno, F. Autelitano, R. Roncella, F. Giuliani, Automatic detection of stone pavement’s pattern based on UAV photogrammetry, Autom. Constr. 122 (2021) 103477, https://doi.org/10.1016/j.autcon.2020.103477.

[8] D. Reagan, A. Sabato, C. Niezrecki, Feasibility of using digital image correlation for unmanned aerial vehicle structural health monitoring of bridges, Struct. Health Monit. 17 (5) (2018) 1056–1072, https://doi.org/10.1177/1475921717735326.

[9] C.M. Yeum, J. Choi, S.J. Dyke, Automated region-of-interest localization and classification for vision-based visual assessment of civil infrastructure, Struct. Health Monit. 18 (3) (2019) 675–689, https://doi.org/10.1177/1475921718765419.

[10] Y. Zhang, K.V. Yuen, Review of artificial intelligence-based bridge damage detection, Adv. Mech. Eng. 14 (9) (2022) 16878132221122770, https://doi.org/10.1177/16878132221122770.

[11] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444, https://doi.org/10.1038/nature14539.

[12] C.Z. Dong, F.N. Catbas, A review of computer vision–based structural health monitoring at local and global levels, Struct. Health Monit. 20 (2) (2021) 692–743, https://doi.org/10.1177/1475921720935585.

[13] C. Koch, S. Paal, A. Rashidi, Z. Zhu, M. König, I. Brilakis, Achievements and challenges in machine vision-based inspection of large concrete structures, Adv. Struct. Eng. 17 (3) (2014) 303–318, https://doi.org/10.1260/1369-4332.17.3.303.

[14] B. Lei, Y. Ren, N. Wang, L. Huo, G. Song, Design of a new low-cost unmanned aerial vehicle and vision-based concrete crack inspection method, Struct. Health Monit. 19 (6) (2020) 1871–1883, https://doi.org/10.1177/1475921719898862.

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

[16] D. Marcos, M. Volpi, B. Kellenberger, D. Tuia, Land cover mapping at very high resolution with rotation equivariant CNNs: towards small yet accurate models, ISPRS J. Photogramm. Remote Sens. 145 (2018) 96–107, https://doi.org/10.1016/j.isprsjprs.2018.01.021.

[17] C.L. Fan, Y.J. Chung, Design and optimization of CNN architecture to identify the types of damage imagery, Math 10 (19) (2022) 3483, https://doi.org/10.3390/math10193483.

[18] C. Xiong, Q. Li, X. Lu, Automated regional seismic damage assessment of buildings using an unmanned aerial vehicle and a convolutional neural network, Autom. Constr. 109 (2020) 102994, https://doi.org/10.1016/j.autcon.2019.102994.

[19] D. Zou, M. Zhang, Z. Bai, T. Liu, A. Zhou, X. Wang, W. Cui, S. Zhang, Multicategory damage detection and safety assessment of post-earthquake reinforced concrete structures using deep learning, Comput. Aided Civ. Infrastruct. Eng. 37 (9) (2022) 1188–1204, https://doi.org/10.1111/mice.12815.

[20] T. Settou, M.K. Kholladi, A. Ben Ali, Improving damage classification via hybrid deep learning feature representations derived from post-earthquake aerial images, Int. J. Image Data Fusion 13 (1) (2022) 1–20, https://doi.org/10.1080/19479832.2020.1864787.

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

[22] H. Zoubir, M. Rguig, M. El Aroussi, A. Chehri, R. Saadane, G. Jeon, Concrete bridge defects identification and localization based on classification deep convolutional neural networks and transfer learning, Remote Sens 14 (19) (2022) 4882, https://doi.org/10.3390/rs14194882.

[23] M.K. Kim, Q. Wang, H. Li, Non-contact sensing based geometric quality assessment of buildings and civil structures: a review, Autom. Constr. 100 (2019) 163–179, https://doi.org/10.1016/j.autcon.2019.01.002.

[24] S. Yoon, G.H. Gwon, J.H. Lee, H.J. Jung, Three-dimensional image coordinate-based missing region of interest area detection and damage localization for bridge visual inspection using unmanned aerial vehicles, Struct. Health Monit. 20 (4) (2021) 1462–1475, https://doi.org/10.1177/1475921720918675.

[25] S. Zhao, F. Kang, J. Li, Concrete dam damage detection and localization based on YOLOv5s-HSC and photogrammetric 3D reconstruction, Autom. Constr. 143 (2022) 104555, https://doi.org/10.1016/j.autcon.2022.104555.

[26] X.U. Yang, F.A.N. Yunlei, B.A.O. Yuequan, L.I. Hui, Few-shot learning for structural health diagnosis of civil infrastructure, Adv. Eng. Inf. 62 (2024) 102650, https://doi.org/10.1016/j.aei.2024.102650.

[27] Y. Zhang, K.V. Yuen, Crack detection using fusion features-based broad learning system and image processing, Comput. Aided Civ. Infrastruct. Eng. 36 (2021) 1568–1584.

[28] G.P. Bu, S. Chanda, H. Guan, J. Jo, M. Blumenstein, Y.C. Loo, Crack detection using a texture analysis-based technique for visual bridge inspection, Electron. J. Struct. Eng. 14 (1) (2015) 41–48, https://doi.org/10.56748/ejse.141881.

[29] C.L. Fan, Detection of multidamage to reinforced concrete using support vector machine-based clustering from digital images, Struct. Control Health Monit. 28 (12) (2021) e2841, https://doi.org/10.1002/stc.2841.

[30] Y. Shi, L. Cui, Z. Qi, F. Meng, Z. Chen, Automatic road crack detection using random structured forests, IEEE Trans. Intell. Transport. Syst. 17 (12) (2016) 3434–3445, https://doi.org/10.1109/TITS.2016.2552248.

[31] M.R. Jahanshahi, S.F. Masri, C.W. Padgett, G.S. Sukhatme, An innovative methodology for detection and quantification of cracks through incorporation of depth perception, Mach, Vis. Appl. 24 (2) (2013) 227–241, https://doi.org/10.1007/s00138-011-0394-0.

[32] D. Lattanzi, G.R. Miller, Robust automated concrete damage detection algorithms for field applications, J. Comput. Civ. Eng. 28 (2) (2014) 253–262, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000257.

[33] O. Moselhi, T. Shehab-Eldeen, Classification of defects in sewer pipes using neural networks, J. Infrastruct. Syst. 6 (3) (2000) 97–104, https://doi.org/10.1061/(ASCE)1076-0342(2000)6:3(97.

[34] H.S. Yoo, Y.S. Kim, Development of a crack recognition algorithm from non-routed pavement images using artificial neural network and binary logistic regression, KSCE J. Civ. Eng. 20 (4) (2016) 1151–1162, https://doi.org/10.1007/s12205-015-1645-9.

[35] X. Liang, Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization, Comput. Aided Civ. Infrastruct. Eng. 34 (5) (2019) 415–430, https://doi.org/10.1111/mice.12425.

[36] Y. Xu, S. Li, D. Zhang, Y. Jin, F. Zhang, N. Li, H. Li, Identification framework for cracks on a steel structure surface by a restricted Boltzmann machine algorithm based on consumer-grade camera images, Struct. Control Health Monit. 25 (2) (2018) e2075, https://doi.org/10.1002/stc.2075.

[37] C.M. Yeum, S.J. Dyke, J. Ramirez, Visual data classification in post-event building reconnaissance, Eng. Struct. 155 (2018) 16–24, https://doi.org/10.1016/j.engstruct.2017.10.057.

[38] Z. Tong, J. Gao, Z. Han, Z. Wang, Recognition of asphalt pavement crack length using deep convolutional neural networks, Road Mater. Pavement Des. 19 (6) (2018) 1334–1349, https://doi.org/10.1080/14680629.2017.1308265.

[39] Y. Gao, K.M. Mosalam, Deep transfer learning for image-based structural damage recognition, Comput. Aided Civ. Infrastruct. Eng. 33 (9) (2018) 748–768, https://doi.org/10.1111/mice.12363.

[40] R. Santos, D. Ribeiro, P. Lopes, R. Cabral, R. Calçada, Detection of exposed steel rebars based on deep-learning techniques and unmanned aerial vehicles, Autom. Constr. 139 (2022) 104324, https://doi.org/10.1016/j.autcon.2022.104324.

[41] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, Proc. IEEE Conf. Comput. Vis. Pattern Recognit (2015) 3431–3440.

[42] S. Li, X. Zhao, G. Zhou, Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network, Comput. Aided Civ. Infrastruct. Eng. 34 (7) (2019) 616–634, https://doi.org/10.1111/mice.12433.

[43] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: 18th International Conference, Med. Image Comput. Comput. Assist. Interv. MICCAI, 2015, pp. 234–241. October 5–9, Munich, Germany.

[44] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: a deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495, https://doi.org/10.1109/TPAMI.2016.2644615.

[45] L.C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587, https://doi.org/10.48550/arXiv.1706.05587.

[46] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, X. Yang, Automatic pixel-level crack detection and measurement using fully convolutional network, Comput. Aided Civ. Infrastruct. Eng. 33 (12) (2018) 1090–1109, https://doi.org/10.1111/mice.12412.

[47] 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.

[48] J.J. Rubio, T. Kashiwa, T. Laiteerapong, W. Deng, K. Nagai, S. Escalera, K. Nakayama, Y. Matsuo, H. Prendinger, Multi-class structural damage segmentation using fully convolutional networks, Comput. Ind. 112 (2019) 103121, https://doi.org/10.1016/j.compind.2019.08.002.

[49] M. Wang, J.C. Cheng, A unified convolutional neural network integrated with conditional random field for pipe defect segmentation, Comput. Aided Civ. Infrastruct. Eng. 35 (2) (2020) 162–177, https://doi.org/10.1111/mice.12481.

[50] C. Zhang, C. Chang, M. Jamshidi, Simultaneous pixel-level concrete defect detection and grouping using a fully convolutional model, Struct. Health Monit. 20 (2021) 2199–2215, https://doi.org/10.1177/1475921720985437.

[51] H. Fu, D. Meng, W. Li, Y. Wang, Bridge crack semantic segmentation based on improved Deeplabv3+, J. Mar. Sci. Eng. 9 (6) (2021) 671, https://doi.org/10.3390/jmse9060671.

[52] J.J. Wang, Y.F. Liu, X. Nie, Y.L. Mo, Deep convolutional neural networks for semantic segmentation of cracks, Struct. Control Health Monit. 29 (1) (2022) e2850, https://doi.org/10.1002/stc.2850.

[53] Y. Xu, Y. Fan, H. Li, H, Lightweight semantic segmentation of complex structural damage recognition for actual bridges, Struct. Health Monit. 22 (5) (2023) 3250–3269, https://doi.org/10.1177/14759217221147015.

[54] C. Feng, H. Zhang, S. Wang, Y. Li, H. Wang, F. Yan, Structural damage detection using deep convolutional neural network and transfer learning, KSCE J. Civ. Eng. 23 (10) (2019) 4493–4502, https://doi.org/10.1007/s12205-019-0437-z.

[55] F. Ni, J. Zhang, Z. Chen, Pixel-level crack delineation in images with convolutional feature fusion, Struct. Control Health Monit. 26 (1) (2019) e2286, https://doi.org/10.1002/stc.2286.

[56] S. Li, X. Zhao, Image-based concrete crack detection using convolutional neural network and exhaustive search technique, Adv. Civ. Eng. 2019 (1) (2019) 6520620, https://doi.org/10.1155/2019/6520620.

[57] Y.J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, Comput. Aided Civ. Infrastruct. Eng. 32 (5) (2017) 361–378, https://doi.org/10.1111/mice.12263.

[58] Y. Xue, Y. Li, A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects, Comput. Aided Civ. Infrastruct. Eng. 33 (8) (2018) 638–654, https://doi.org/10.1111/mice.12367.

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

[60] L. Zhang, J. Shen, B. Zhu, A research on an improved Unet-based concrete crack detection algorithm, Struct. Health Monit. 20 (4) (2021) 1864–1879, https://doi.org/10.1177/1475921720940068.

[61] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, Z.S. Lin, B. Guo, Swin transformer: hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022.

[62] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J.M. Alvarez, P. Luo, P. SegFormer, Simple and efficient design for semantic segmentation with transformers, Adv. Neural Inf. Process. Syst. 34 (2021) 12077–12090.

[63] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European Conference on Computer Vision *(ECCV)*, 2018, pp. 801–818.

[64] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, J. Sun, Unified perceptual parsing for scene understanding, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 418–434.

[65] Q. Zhou, H. Shi, W. Xiang, B. Kang, L.J. Latecki, DPNet: Dual-path network for real-time object detection with lightweight attention, IEEE Trans. Neural Netw. Learn. Syst. (2024), https://doi.org/10.1109/TNNLS.2024.3376563.

[66] C. Liu, G. Li, WegFormer: transformers for weakly supervised semantic segmentation, Expet Syst. 41 (3) (2024) e13495, https://doi.org/10.1111/exsy.13495.

[67] M.R. Jahanshahi, S.F. Masri, Parametric performance evaluation of wavelet-based corrosion detection algorithms for condition assessment of civil infrastructure systems, J. Comput. Civ. Eng. 27 (4) (2013) 345–357, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000225.

[68] 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.

[69] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and $<$ 0.5 MB model size, arXiv preprint arXiv:1602.07360, https://doi.org/10.48550/arXiv.1602.07360.

[70] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861, https://doi.org/10.48550/arXiv.1704.04861.

[71] B. Zoph, V. Vasudevan, J. Shlens, Q.V. Le, Learning transferable architectures for scalable image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8697–8710. Salt Lake City, Utah.

[72] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, Q.V. Le, Mnasnet: platform-aware neural architecture search for mobile, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 16–20, Long Beach, California, 2019, pp. 2820–2828.

[73] M. Tan, Q.V. Le, Efficientnet: rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, May 2019, pp. 6105–6114.