POLITECNICO DI TORINO
Repository ISTITUZIONALE

Civil infrastructure defect assessment using pixel-wise segmentation based on deep learning

Original
Civil infrastructure defect assessment using pixel-wise segmentation based on deep learning / Savino, Pierclaudio; Tondolo, Francesco. - In: JOURNAL OF CIVIL STRUCTURAL HEALTH MONITORING. - ISSN 2190-5452. - ELETTRONICO. - 13:(2023), pp. 35-48. [10.1007/s13349-022-00618-9]

Availability:
This version is available at: 11583/2970859 since: 2022-09-01T11:17:43Z

Publisher:
Springer

Published
DOI:10.1007/s13349-022-00618-9

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

08 May 2026

Journal of Civil Structural Health Monitoring (2023) 13:35–48
https://doi.org/10.1007/s13349-022-00618-9

ORIGINAL PAPER

Civil infrastructure defect assessment using pixel-wise segmentation based on deep learning

Pierclaudio Savino[1] · **Francesco Tondolo**[1]

Received: 13 June 2022 / Revised: 17 July 2022 / Accepted: 2 August 2022 / Published online: 25 August 2022
© The Author(s) 2022

Abstract
Nowadays, the number of aging civil infrastructures is growing world-wide and when concrete is involved, cracking and delamination can occur. Therefore, ensuring the safety and serviceability of existing civil infrastructure and preventing an inadequate level of damage have become some of the major issues in civil engineering field. Routine inspections and maintenance are then required to avoid leaving these defects unexplored and untreated. However, due to the limitations of on-field inspection resources and budget management efficiency, automation technology is needed to develop more effective and pervasive inspection processes. This paper presents a pixel-wise classification method to automatically detect and quantify concrete defects from images through semantic segmentation network. The proposed model uses Deeplabv3+ network with weights initialized from pre-trained neural networks. The comparison study among the performance of different deep neural network models resulted in ResNet-50 as the most suitable network for applications of civil infrastructure defects segmentation. A total of 1250 images have been collected from the Internet, on-field bridge inspections and Google Street View in order to build an invariant network for different resolutions, image qualities and backgrounds. A randomized data augmentation allowed to double the database and assign 2000 images for training and 500 images for validation. The experimental results show global accuracies for training and validation of 93.42% and 91.04%, respectively. The promising results highlighted the suitability of the model to be integrated in digitalized management system to increase the productivity of management agencies involved in civil infrastructure inspections and digital transformation.

Keywords Civil infrastructure · Automated inspection · Damages · Semantic segmentation · Deep learning · Computer vision

1 Introduction

The active management of aging civil infrastructure has become a twenty-first century challenge for transportation agencies around the world, committed to maintaining healthy infrastructure and preventing unexpected structural failures. However, there is clear evidence of reduced efficiency, slow recovery operations and aged infrastructure assets. In the 1990s, the UK and Switzerland construction maintenance markets, already accounted for 50% of the total value of the construction markets [1]. The Report Card published by the American Society of Civil Engineers in 2021 stated that the United States has more than 617,000 bridges, 42% of which are at least 50 years old and with the current investment rate it will take until 2071 to cover the $125 billion of rehabilitation cost [2]. In South Korea, regular annual inspections of 270,000 structures are required, although the budget and number of inspectors are gradually decreasing [3]. Moving to the European continent, much of the bridge infrastructures were built after World War II and are now beyond their useful lives. All these numbers highlighted the need for reliable and integrated systems of inspection, assessment and maintenance to ensure safety and efficient allocation of resources. Quickly and frequently surveys are required to plan essential maintenance and repairs in a proactive way before it becomes too dangerous and expensive. Currently, the assessment of structural condition involves the engagement of qualified inspector which perform on-site inspections with the use of photographs, annotations, drawings and the collection of historical information. However, as these are carried out at pre-fixed time intervals, there is a

✉ Pierclaudio Savino
pierclaudio.savino@polito.it

[1] Department of Structural, Geotechnical and Building Engineering, Politecnico di Torino, 10129 Turin, Italy

risk of performance below an established threshold between an inspection and another. Furthermore, such inspection can be time-consuming, costly, laborious and dangerous especially for inaccessible structures. Finally, depending on the expertise of the inspector, there are subjectivity and human error that can lead to a different classification for similar defects. To address all these issues, improved inspection with less human intervention, lower costs and higher spatial resolution needs to be developed to enable automated assessment of civil infrastructures condition. In recent years, with the development of low-cost and high-quality imaging devices, computer vision technique has been gathering increasing attention in the research of the civil engineering community. Indices for local condition assessment such as crack, spalling, corrosion, and delamination can be extracted from visual images containing structure surface. The advantages of this method are related to the possibility to enable long-distance, non-contact, low-cost, objective and automatic condition assessment [4, 5]. Moreover, vision sensors used in conjunction with vehicle or unmanned aerial vehicles (UAV) is proposed as one of the promising strategies for fast scanning of higher spatial resolution without the need of traffic closure.

Computer vision-based inspection varies from conventional approaches using image processing algorithms to recent attempts based on deep learning techniques. Traditional detection algorithms rely on the manual features extraction which transform the available data into valuable information, ranging from statistical-based method on grey-scale distribution [6], colour and texture descriptors [7, 8], binarization methods [9] and machine learning-based model [10]. However, the application of image processing in an automated structural inspection environment is limited, as these techniques do not consider the contextual information provided by the regions around the defects. These techniques need to be tuned manually, depending on the type of target structures to be monitored [5]. Furthermore, varying the lighting and shadowing controlled during image capturing or acquiring skewed long-range images, could yield false and erroneous results. Real-world situations are very varied and building a general algorithm that can be successful in these general cases is quite complex.

The development of deep learning techniques has greatly extended the capability and robustness of traditional vision-based damage detection by automatically extracting features, without requiring time-consuming and complex processes. As the features are defined by the machine, human bias/error is avoided and replaced by the error of the system, moving from a knowledge-driven approach to a data-driven approach. Different applications for damage detection have been studied for a wide variety of structures and type of defects, ranging from cracks and spalling to corrosion. Convolutional neural network (CNN) architectures have been developed to build a classifier for detecting cracks of steel box girders [11], road pavement [12] and concrete surface [13]. All these methods, to locate the crack, first consider scanning the original images with sub-patches and then activating only those with defects. To overcome the rough localization based on sliding window detection, Quqa et al. [14] proposed a two-step approach that first identifies the "cracked" regions and then applies image processing techniques only to locate the crack pixels. In order to avoid the need for a wide dataset to obtain high level of accuracy for CNNs trained from scratch, the transfer learning technique was adopted on pre-trained networks. The well-known AlexNet architecture has been fine tuned to classify cracking [15, 16] and spalling [17] on concrete surface. Savino and Tondolo [18] compared eight pre-trained networks to classify images containing undamaged, cracked and delaminated structural elements, reaching the maximum accuracy of 94% with GoogLeNet architecture. Kruachottikul et al. [19] developed a defect-inspection system for reinforced concrete bridge substructure, able to classify cracking, erosion, honeycomb, scaling, and spalling defects with an accuracy of 81%. Since the image classification approach can only distinguish between images based on the expected class, object detection methods have recently been applied to recognize and locate multiple damages within bounding box. Cha et al. [20] proposed a Faster Region-based Convolutional Neural Network (Faster R-CNN) to detect in the same image concrete crack, steel corrosion, bolt corrosion and steel delamination with an average precision of 87.8%. Faster R-CNNs were also used to identify and locate damage of masonry historic structures [21], urban shield tunnel lining [22] and large crane structures [23]. However, object-detection-based methods, providing only class labels and the bounding box around the region of interest, cannot precisely define the shape of the damage but only identify and locate it. Moreover, suffering in the case of overlapping regions, they are unsuitable to provide morphological information and the extent of defects.

An effective method to delineate the precise location and shape of object is named semantic segmentation. More specifically, a semantic segmentation network classifies each pixel of an image with a certain label, providing an image that is segmented by class. To the knowledge of the authors, relatively few works used at the date of this paper, semantic segmentation neural networks for civil infrastructures defect assessment. Zhang et al. [24] proposed a CNN architecture for the pixel-level pavement crack detection on 3D asphalt surfaces with a precision of 90.13%. Zhu and Song [25] presented a weakly supervised segmentation and detection network based on autoencoder to identify cracks on asphalt concrete bridge deck. Most of the studies concerned pixel-level surface crack detection using transfer learning [26] or combining the advantages of pre-trained networks [27–30].

Pozzar et al. [31] investigated the performance of different pre-trained models to detect multiclass concrete damages using thermographic images, identifying the VGG16 model as the most promising with average IoU values of 59.5% for delamination and 39.4% for crack.

In most of the previous research, neural networks were trained with images collected under near-ideal laboratory conditions, such as camera positions and angles, depending on the appearance and location of the defects. Furthermore, as they were considered datasets with hundreds of images, much smaller images were cropped from the original images to increase their number. However, this approach cannot cover the diversity of the on-field environment because it is difficult to reproduce ideal conditions and continuously control lighting direction, positions and angles of cameras installed on UAV. This condition makes most existing image-based methods highly dependent on the data used, generalizing poorly to other datasets. Most of the research efforts, as mentioned previously, only focused on the semantic segmentation of specific defect at a time. To the knowledge of the authors, never have been investigated the performance of pre-trained semantic segmentation models to detect multiclass concrete defects. Based on the mentioned gaps, this research proposed a CNN able to perform the semantic segmentation of images containing "Crack", "Delamination" and "Background" in several civil infrastructures. The first objective was to train a robust neural network that is not affected by quality of images and that is effective for a wide range of on-field inspection. Therefore, the neural network has been developed considering a dataset of 1250 images, collected from Internet, on-field bridge inspections and Google Street View. The images are affected by a broad range of noise linked by the sources and represent real environmental conditions with several background. The second objective was to find among the existing pre-trained neural networks the most suitable for civil engineering defects detection task. It will allow further research to detect additional types of structural damage, such as corrosion, efflorescence, stain moisture and voids. Furthermore, morphological information were extracted to prove the superiority of the semantic segmentation approach over the existing object detection methods in providing quantitative information about civil infrastructure defects.

2 Semantic segmentation

The general architecture for the semantic image segmentation task is a CNN which associates each pixel of an image with a corresponding class label. Generally, the architecture of CNN consists of shallow layers to learn low-level features and deep layers specialized on high-level details. For image classification task, aimed to learn what the image contains, the expensive computation of deep neural networks is relieved by down-sampling of feature maps with pooling or strided convolutions. However, for image segmentation task, the full-resolution semantic prediction must be preserved by adopting encoder/decoder structures (Fig. 1). The encoder part down-samples the input into low-resolution feature maps and learns to discriminate between classes, the decoder part up-samples from a low-resolution map to a full-resolution segmentation map.

The down-sampling part is actually a very deep CNN which is built adopting multiple layers, such as convolution, pooling and activation layers. Since the excessive downsizing of encoder part due to consecutive pooling operations results in a loss of information which cannot be recovered in the decoder part, Chen et al. [32] proposed DeepLabv3+ decoder to refine the segmentation results. The proposed model employs Atrous Spatial Pyramid Pooling (ASPP) to capture multi-scale contextual information without losing spatial resolution. Finally, the softmax layer predicts the class of each pixel after a series of transpose convolutions which upsample the resolution of the feature maps.

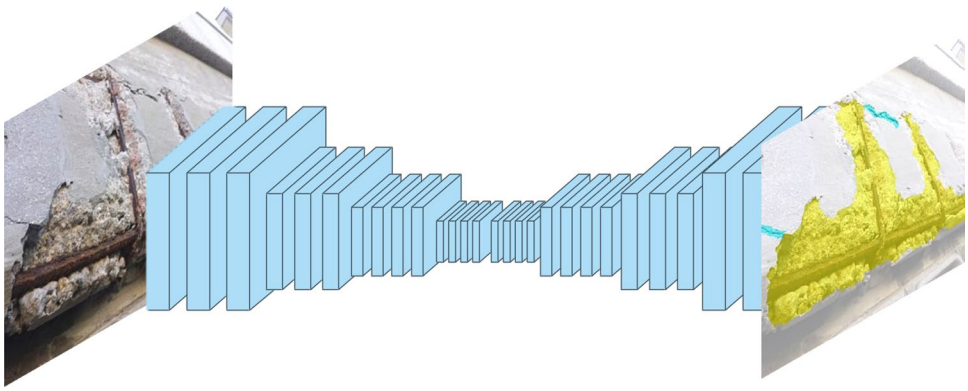

Fig. 1 Semantic segmentation neural network

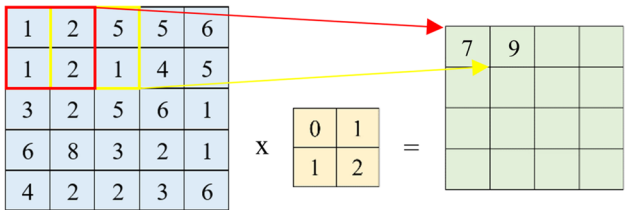

Fig. 2 Convolutional operation example

2.1 Convolutional layer

Convolutional layers are the main building blocks used in CNNs, responsible for capturing different level of features. The first element involved in a Convolutional layer to perform convolution operation is called kernel or filter. A convolution is a linear operation that involve an element-wise multiplication between the input and the weights contained in the filter (Fig. 2). The sliding step size of the kernel on the input is defined as a stride and affects, together with the padding, the size of the convolved feature.

Assuming the case of one-dimensional convolution, the output of the convolution process is

$$y(i) = \sum_{k=1}^{K} x(i+k) \cdot w(k) + b \tag{1}$$

where $x\,(i)$ is the input, $w\,(k)$ is the filter of length K and b is the bias. Systematic application of the filter across an image, allows to extract a feature anywhere in the image and create a feature map. It is important to note that the local dependencies in the original image depend on the weights that are automatically tuned during the training process. Since the convolution is a linear operation, the Convolutional layer ends with an activation function to introduce nonlinear transformation components. The most used activation function is the Rectified Linear Unit (ReLU) which returns the value provided as input directly, or zero for negative input. Because ReLU is linear for positive values, it facilitates much faster computation during the training process of a neural network with backpropagation.

2.2 Pooling layer

Similar to the Convolutional layer, the Pooling layer decreases the size of the convolved feature to reduce the probability of overfitting and the computational power. The key features are commonly extracted by two types of Pooling: Max Pooling and Average Pooling (Fig. 3).

The Max Pooling returns the maximum value for a portion of the feature map covered by the kernel, whereas the Average Pooling computes the mean value. The Pooling layer is frequently used after the Convolutional layer in order to intensify the important features kept in the Convolutional layers and discard all the information irrelevant for the output.

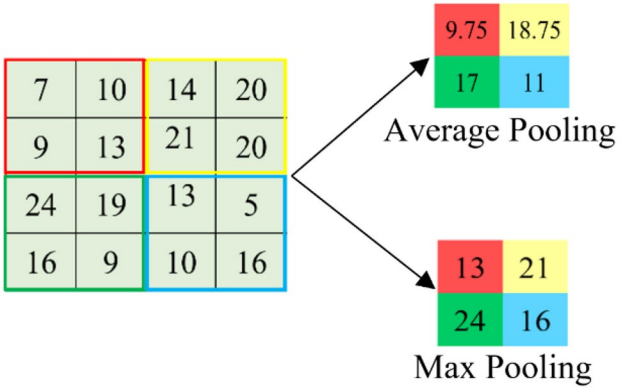

Fig. 3 Pooling layer example

2.3 Atrous spatial pyramid pooling

DeepLabv3+ applies several parallel Atrous convolution, also called hole convolution or dilated convolution, to capture the features computed by CNNs at different scale. Atrous convolution is a type of convolution that increases the filter size using the same number of parameters (Fig. 4). The dilation rate, l, indicates how much the filter is widened, l-1 is the number of hole or zeros filled between consecutive filter parameters.

When the rate is 1, it corresponds to a standard convolution; when the rate is equal to 2, the receptive field goes to 5×5 while having 3×3 convolution parameters. Similarly, the Atrous convolution with dilation rate 3 is able to get the information of 7×7 convolution with the same number of parameters. In case of one-dimensional convolution, the Atrous convolution for each location i on the output feature map results:

$$y(i) = \sum_{k=1}^{K} x(i+lk) \cdot w(k) \tag{2}$$

where $x\,(i)$ is the input of a pixel, and $w\,(k)$ is the filter of length K. As pointed out above, the standard convolution is a special case in which the dilation rate is $l=1$.

In the ASPP model, four parallel Atrous convolutions with different rate are applied to ensure detailed spatial information and capturing features at each scale. After the parallel operations, the results are concatenated by converting to a 1-dimensional vector.

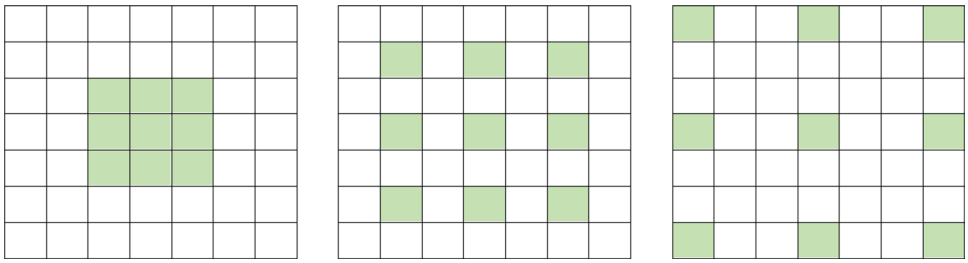

Fig. 4 Atrous convolution (3 × 3) with different dilation rates of 1, 2, and 3, respectively

2.4 Transposed convolutional layer

To produce the pixel-to-pixel prediction results, an upsampling operation is introduced to increase the spatial resolution of a coarse feature map to the dimension of the original image. This operation is called transposed convolution and, unlike the Convolutional layer, the output becomes larger than the input. As presented in Fig. 5, each element in the input is multiplied by the kernel (i.e., the element containing the weights) and then these middle matrices are combined with strides in both width and height directions. Finally, the assembled values in overlapping regions are added together to extract the extended input.

In contrast to the regular convolution where strides are specified for the input, in the transposed convolution, they are specified for intermediate matrices increasing the size of the output.

2.5 Softmax layer

At the last layer of the CNN, it is necessary to have a layer that assigns a score with each class to each pixel within the original image. To convert the vector of output number into a vector of multiclass categorical probability distribution by a normalized exponential, it is used the softmax function, which is expressed as

$$\sigma(y)_i = \frac{e^{y_i}}{\sum_{j=1}^{K} e^{y_j}} \tag{3}$$

where y_i are the elements of the input vector to the softmax function and K is the number of classes in the multiclass classifier. To quantify how far the network prediction are from the actual classes, it is calculated the Cross-Entropy loss function, defined as

$$L_{\mathrm{CE}} = -\sum_{i=1}^{n} t_i \log\left(p_i\right) \tag{4}$$

where t_i is ground truth and p_i is the Softmax predicted score at the specific pixel. The smaller the loss function, the closer the predicted values are to the right classes.

3 Training process

In this study, to minimizes the Cross-Entropy loss during the training process, it has been selected the stochastic gradient descent algorithm with momentum. This is an iterative method for optimizing the loss by adjusting the weights of the network and increasing accuracy. To define the most suitable architecture for the damage detection task, it has been used the transfer learning of pre-trained networks on ImageNet dataset. The transfer of learned generic features helps to achieve better performance with less training time without considering randomly initialized weights from scratch. On the other hand, the fine-tuning of deep layers and the new classification layer refines the representations of the high-level features of the new dataset in the base model. Furthermore, for the best performing network the hyperparameters configuration has been optimized to define the optimal architecture. In this work Deeplabv3+ networks have been created, with weights initialized from pre-trained MobileNet-v2 [33], Xception [34], ResNet-18 and ResNet-50 [35] networks. The MATLAB Deep Learning Toolbox allows easy implementation of the "deeplabv3plusLayers" to create a DeepLabv3+ layer with the specified base network, number

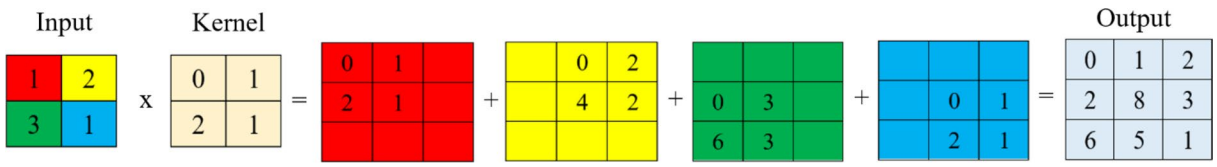

Fig. 5 Transposed convolution with a 2 × 2 kernel and stride of 1 for a 2 × 2 input

of classes, and image size. In addition, the "pixelClassificationLayer" function creates a pixel classification output layer to provide the categorical label for each image pixel. The training has been performed using MATLAB on a NVIDIA GeForce GTX 1650 Ti with 4 GB of GPU memory.

3.1 Pretrained networks

Pretrained networks are layered architectures shared by their respective teams, which allows to replace the final layer and retrain some of the previous layer to reach a stable state on a new task. Deep learning models have different accuracy, speed and size which should be considered as a starting point in the choice for a new classification system. The present experimental study involved balanced fast/accurate pre-trained network to be deployed with high performance on embedded system, such as MobileNet-v2, Xception, ResNet-18 and ResNet-50.

MobileNet-v2 is a neural network architecture announced by Google researcher to run efficiently on devices with low computational power. The main idea behind MobileNet architecture is the split of the convolution layer into a depthwise convolution layer and 1×1 convolution layer to form a "depthwise separable" convolution block. The depthwise convolution applies a single convolutional filter for each channel image, the pointwise convolution builds new features through computing linear combinations in depth dimension. In v2, 1×1 expansion and 1×1 projection layers were added at the beginning and end of the depthwise convolution to form the Bottleneck Residual block which allows the use of low-dimension tensors and reduces the number of computations. The full MobileNet-v2 architecture, then consists of 17 of these building blocks followed by a 1×1 convolution, a global average pooling layer and a classification layer (Table 1).

Xception Model was developed by Google researchers as extension of inception architecture, involving "depthwise separable" convolutions and Max Pooling, all linked with shortcuts connections. Adding connections which skip one or more layers, avoid degradation problem related to learning of identity mappings for deeper networks. The specificity of Xception is that the depthwise convolution is not followed by a pointwise convolution, but the order is inverted. The feature extraction base is formed by a linear stack of 36 Convolutional layers structured into 14 modules. The diagram in Fig. 6 shows in detail the number of filters, the filter size and the strides.

The shortcuts connections were introduced for the first time within the deep Residual Network which made it possible to train hundreds or thousands of layers without running into the vanishing gradient problem. There are several variants of ResNet architecture that are based on the same concept but with different number of layers, such as ResNet-18 and ResNet-50. ResNet networks consist mainly of five types of convolution blocks called conv1, conv2, conv3 and conv5 followed by a fully connected layer and a softmax layer. Each convolution block uses 2 convolution layers of size 3×3 for ResNet-18 or 3 convolution layers of size 1×1, 3×3 and 1×1 for ResNet-50. In Table 2 is reported a summary of the output size at every layer and the dimension of the convolutional filters at every point in the architectures.

Table 3 provides further details about the network architectures adopted in this study. The depth is defined as the number of sequential convolutional or fully connected layers from the input layer to the output layer.

Table 1 MobileNet-v2 architecture

Input	Operator	*n*
$224\times224\times3$	conv2d	1
$112\times112\times32$	bottleneck	1
$112\times112\times16$	bottleneck	2
$56\times56\times24$	bottleneck	3
$28\times28\times32$	bottleneck	4
$14\times14\times64$	bottleneck	3
$14\times14\times96$	bottleneck	3
$7\times7\times160$	bottleneck	1
$7\times7\times320$	conv2d 1×1	1
$7\times7\times1280$	avgpool 7×7	1
$1\times1\times1280$	conv2d 1×1	–

3.2 Building database

The performance of a neural network is related to the variety of the training images. Considering a training dataset with images characterized by constrained conditions may lead CNNs to poorly perform in case of classification task outside the assumptions. In order to obtain training images under a wide variety of possible situations, the training dataset was established collecting raw images from Internet, on-field bridge inspection, and Google Street View. The use of three different sources allowed to gather images with different quality, resolution and background, increasing the usefulness of the research also for on-field applications with low-cost sensors. A total of 1250 images have been manually labeled, using the MATLAB Image Labeler app, where the pixels are labeled as "Delamination", "Crack" and "Background", respectively. Figure 7 shows examples of collected raw images used to build the datastore and their ground truth annotation. Withe pixels correspond to "Background", yellow color is used to annotate "Delamination" and cyan is used for "Crack" annotation.

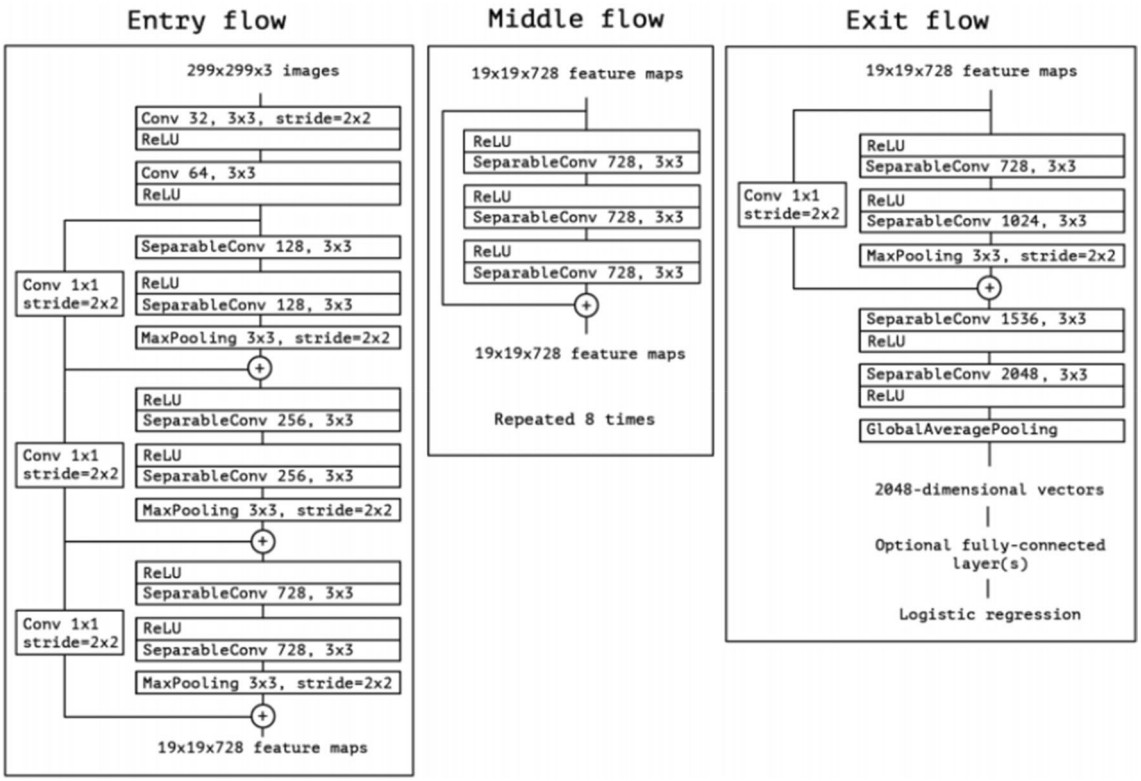

Fig. 6 Xception architecture

Table 2 ResNet-18 and ResNet-50 architectures

Layer	Output	ResNet-18	ResNet-50
conv1	112×112	7×7×64, stride 2	
conv2	56×56	3×3 maxpool, stride 2	
		$\begin{bmatrix} 3\times3\times64 \\ 3\times3\times64 \end{bmatrix}\times2$	$\begin{bmatrix} 1\times1\times64 \\ 3\times3\times64 \\ 1\times1\times64 \end{bmatrix}\times3$
conv3	28×28	$\begin{bmatrix} 3\times3\times128 \\ 3\times3\times128 \end{bmatrix}\times2$	$\begin{bmatrix} 1\times1\times128 \\ 3\times3\times128 \\ 1\times1\times128 \end{bmatrix}\times4$
conv4	14×14	$\begin{bmatrix} 3\times3\times256 \\ 3\times3\times256 \end{bmatrix}\times2$	$\begin{bmatrix} 1\times1\times256 \\ 3\times3\times256 \\ 1\times1\times256 \end{bmatrix}\times6$
conv5	7×7	$\begin{bmatrix} 3\times3\times512 \\ 3\times3\times512 \end{bmatrix}\times2$	$\begin{bmatrix} 1\times1\times512 \\ 3\times3\times512 \\ 1\times1\times512 \end{bmatrix}\times3$
–	1×1	avgpool, 1000-fc, softmax	

To decrease the computational cost and training time, all the images have been cropped into 300×300 pixels resolution after the labeling operation. Furthermore, to make neural networks invariant to distortions in image data and decrease the probability of overfitting, the amount of training data is increased by applying randomized augmentation with a combination of rotation, reflection and shear. After data augmentation, the doubled database is randomly divided by 80% for the training set and 20% to validate the model. Specifically, 2000 images are randomly selected to generate the training set and another 500 are used to create the validation set.

Table 3 Pretrained networks properties

Network	Depth	Size (MB)	Parameters (Millions)
MobileNet-v2	53	12	3.5
Xception	71	85	22.9
ResNet-18	18	44	11.7
ResNet-50	50	96	25.6

A common issue in concrete damage datasets is the unbalance of class distribution between pixels containing cracks and the others because in general they cover less area in images. When a dataset is unbalanced, the error of the overrepresented classes contributes much more than the

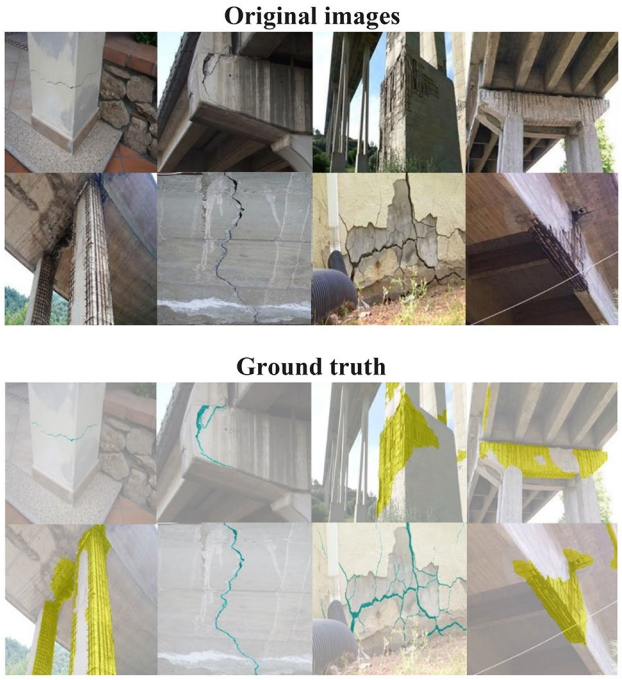

Fig. 7 Examples of images used to build the datastore and their ground truth

Table 4 Pixel numbers and median frequency class weights for each class

	Delamination	Crack	Background
Pixel count	1.4801×10^7	4.3806×10^6	1.3743×10^8
Frequency	0.1620	0.0273	0.7635
Class weight	1	5.9286	0.2122

error contribution of the underrepresented classes, making poor performance for the underrepresented classes. To avoid a semantic segmentation biased toward the dominant classes, a class weighting has been adopted during the training to increase or decrease the importance of a pixel. The weight of each class, w^c, has been defined computing the median frequency weighting according to

$$w^c = \frac{\text{median}(f)}{f^c} \tag{5}$$

where the frequency f represents the number of pixels of the class divided by the total number of pixels in the images that contain an instance of the class c. The number of pixels for each class within the training set, denoted by "Pixel count", its frequency and the class weights, can be seen in Table 2.

From Table 4, it can be noticed for the "Crack" class, the lower number of pixels and frequency corresponding to a heavier weighting.

4 Comparative analysis and evaluation

Defined the model architectures and the dataset, the model hyperparameters need to be configured to start with the training process. Being external to the networks, these values cannot be directly estimated from data but can be set using heuristics. Thus, the optimal network architecture has been explored considering a fixed number of 10 epochs, a mini-batch size of 16 images, a momentum of 0.9 and a L_2 regularization of 0.0001. To identify a suitable initial learning rate, have been examined with the training process of each network the values 10^{-3}, 10^{-4} and 10^{-5}. A comparative study has been first established according to the percentage of correctly classified pixels, defined as

$$\text{GA} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{6}$$

where GA is the global accuracy, TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives. Figure 7 depicts the validation accuracy recorded during the training process of each network for different learning rates.

As reported in Fig. 8, the best performing networks have been ResNet-18 and ResNet-50 with a learning rate of 0.001, achieving a global accuracy of 88.76% and 88.57% respectively. However, this metric can present misleading results in case of class imbalance, resulting biased towards the classes that dominate the image. For this reason, to define the network with superior segmentation ability, we considered accuracy and intersection-over-union (IoU) for individual classes. The IoU metric is the ratio between the amount of overlap and the union between the predicted segmentation and the ground truth:

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{7}$$

The lower the IoU, the worse is the prediction result. Tables 3 and 4 show the accuracy and IoU-related measures for ResNet-18 and ResNet-50, respectively.

From Tables 5 and 6, it can be noticed that accuracies for "Delamination" and "Crack" classes improve from

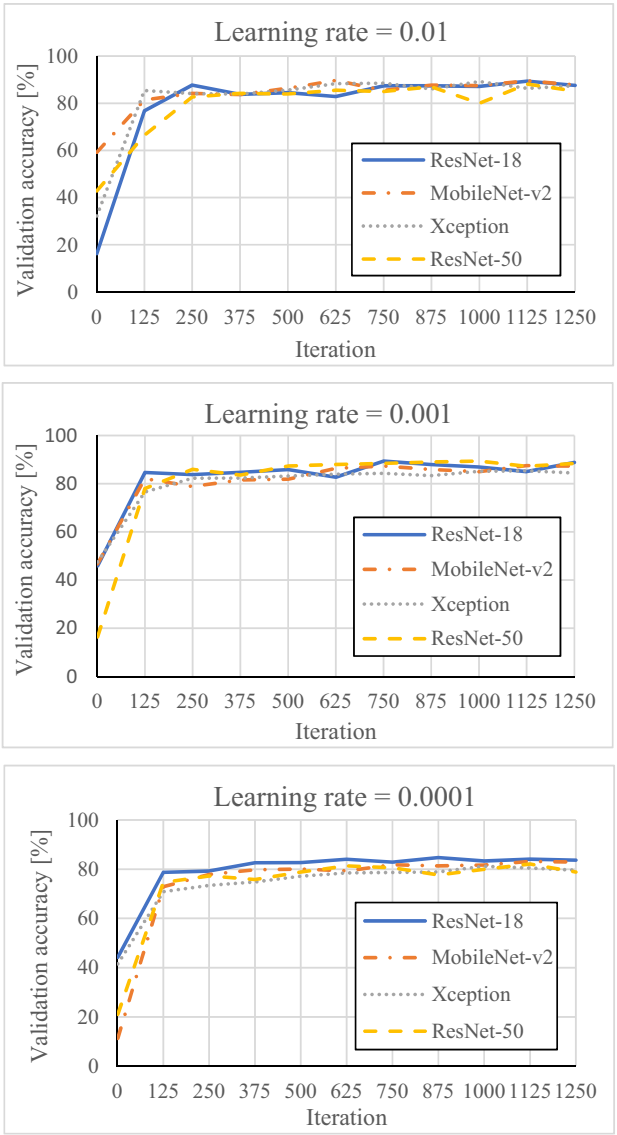

Fig. 8 Validation accuracy during training processes with different learning rates

Table 5 Accuracy and IoU for ResNet-18 network

	Accuracy	IoU
Delamination	0.89331	0.65435
Crack	0.85088	0.243
Background	0.88799	0.87536

ResNet-18 to ResNet-50 network. It can be also noticed a corresponding slight decrease in both the accuracy and the IoU metric for the "Background" class. Overall, since the mean accuracy improves from 0.87739 to 0.89427, by approximately 1.7%, and the mean IoU from 0.5909 to 0.59213, in this work as a reference architecture it has been chosen the one of the ResNet-50 network.

Table 6 Accuracy and IoU for ResNet-50 network

	Accuracy	IoU
Delamination	0.91743	0.65907
Crack	0.883	0.2438
Background	0.88227	0.87353

5 Results and discussion

As shown in the previous section, ResNet-50 achieved better results than ResNet-18. Therefore, an empirical evaluation of optimal hyperparameters has been performed for a further improvement of performance. An extensive search has been conducted to define the best contribution of the previous parameter update to the current learning iteration given by momentum. Furthermore, the regularization term L_2 for the weights to the loss function has been assessed to reduce overfitting. To observe the complete convergence behavior and avoid overfitting, the loss function has been minimized considering a training process with 20 Epochs and training iterations on mini-batch with size 8. The final configuration of tuned hyperparameters is: learning rate 0.001, momentum 0.9, regularization 0.0001, epochs 20 and mini-batch size 8. Figure 9 depicts the final training and validation results for the network considered in this work.

The latest training and validation accuracies achieved after a training time of about 3 h are 93.42% and 91.04%, respectively. Table 7 summarizes semantic segmentation quality metrics for each class.

The result proves that among the three classes, "Crack" has the lowest accuracy whereas "Delamination" has the highest. This finding about the difficulty to recognize the pixel of this class make sense from a visual point of view, since "Delamination" and "Background" have more easily distinguishable spatial features. To examine the performance of the trained and validated network, it has been presented in Fig. 10 an example of images used for the validation processes. The first column contains the original images, the second column consists of ground truth and the last column represents the predictions.

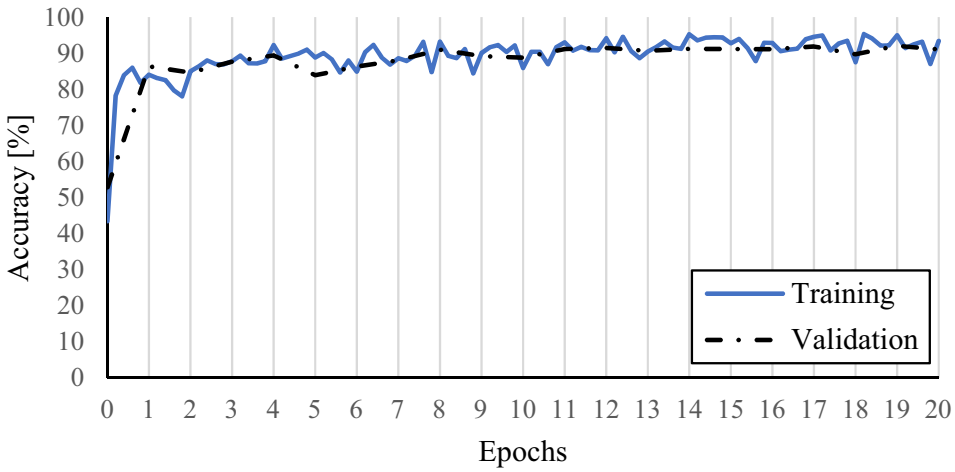

Fig. 9 Accuracies for each epoch

Despite the good performance of the proposed network, it still has some inaccuracies in detecting mainly cracks. The typical incorrect prediction refers to the thickness of the cracks being larger than the ground truth. Although minor errors, the results demonstrate the reliability of this model for the automatic assessment of existing concrete structures. Therefore, a larger training database could improve model capacity and generalization in future applications. On the other hand, performance and results could be improved considering high-resolution images. Figure 11 shows some examples on test images with high-resolution, that are never used for both training and validation phase.

Test images showed that considering high-resolution images could add significant capability to classify civil infrastructure damages, even those related to the "Crack" class. Looking into the details, the proposed method is not susceptible to various background patterns, concrete texture, exposure and environment, resulting useful for on-field civil infrastructure inspection.

5.1 Damages' measurement

Once the damages have been detected, it is possible to extract morphological information to determine durability, conditions of exposure, and to define economic and safety impact. Currently, the actual need and urgency are defined with an approximate and qualitative way, according to quick survey on the infrastructural heritage. For each defect on the structure, extension and intensity are indicated through constant coefficients without referring to quantitative analyses. Complexity, level of detail and the cumbersome of investigations are conversely related to the number of infrastructures on which they are applied and to the uncertainty of the results. The proposed deep learning-based inspection approach not only makes the process automatic but provides useful data to reconstruct damage evolution without operator dependent errors.

Once predicted the class for each pixel, properties of image regions can by quantified by using the MATLAB function "regionprops (Image, 'properties')". Table 8 lists some measurements on the test images of Fig. 11 related to the actual number of pixels in the region classified as "Delamination", "Crack" and "Background" ('Area').

Furthermore, the amount of damage can be defined in terms of percentage of the total area, or other units, given a proper spatial calibration factor.

Table 7 Accuracy and IoU for the optimal tuned hyperparameters

	Accuracy	IoU
Delamination	0.92489	0.71144
Crack	0.86449	0.28453
Background	0.90934	0.90035

6 Conclusions

This paper proposed an automated civil infrastructure inspection based on deep learning to detect and quantify "Delamination", "Crack" and "Background" regions on real structures, at pixel level. To ensure a wide range of adaptability, the training and validation dataset were built

Fig. 10 Examples of detection result by the proposed network

collecting images from the Internet, on-field bridge inspection and Google Street View under uncontrolled situations. Multiple environments, concrete texture and photo properties have been considered in order to obtain a very robust model suitable for real on-field inspections. For the collected images, each pixel has been manually labeled to create ground truth data for training semantic segmentation algorithms. Data augmentation was implemented to enhance diversity and expand the dataset. After augmentation, the number of images used for training and validation was 2000 and 500, respectively.

A comparison study has been performed between pre-trained networks to define the most suitable for the semantic segmentation of civil infrastructure defects. To find the best training model, the best learning rate has been selected with empirical method. The most performing ResNet-50 network has been fine-tuned to set hyperparameters configuration, achieving the highest validation accuracy of 91.04%. With the validation datasets, it is observed that the highest accuracies correspond to "Delamination" and "Background" classes whereas "Crack" class is found the most challenging to detect accurately. In addition, the performance of the trained network was tested considering test images with high-resolution, not used for training and validation. This analysis demonstrated that the proposed method could provide very accurate detection results with reference to all classes. Furthermore, the proposed method has been used not only for detection task but also to quantify defects by extracting morphological information. This research confirmed a high degree of applicability and advantage for computer vision-based inspection in civil infrastructures, which may significantly improve the productivity in the future.

To improve the performance of the semantic segmentation networks and allow engineers to apply this technique for their specific tasks, the datastore can be downloaded as open source from the website (https://drive.google.com/drive/folders/1sdzPAai6d6fVgM-qEFCnCl0MQkIG7NTN?usp=sharing).

Future research will concern the improvement of the semantic segmentation metrics considering a larger dataset and multispectral images that provide further information about each pixel. Furthermore, LiDAR sensor data and digital models will be integrated to develop a fully automated inspection procedure. Thus, computer vision-based method is expected to replace traditional visual inspection in the near future because of the objective assessment and the saving in resources.

Fig. 11 Test images with high-resolution: **a** underside of stairs; **b** piers; **c** girders; **d** piers; **e** pier cap; **f** abutment; **g** pier cap; **h** concrete surface

Table 8 Area measurements on test images (Fig. 11)

Figure	"Delamination" [px]	"Crack" [px]	"Background" [px]
10a	0	58,317	6,504,083
10b	241,068	8768	2,065,204
10c	164,318	4323	986,214
10d	218,733	12,148	5,227,039
10e	765,182	50,047	3,939,521
10f	190,094	2222	657,644
10g	251,914	4280	1,292,598
10h	0	27,957	3,711,667

Funding Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Fam CL (2021) Detection of multidamage to reinforced concrete using support vector machine-based clustering from digital images. Struct Contr Heal Monit. https://doi.org/10.1002/stc.2841
2. ASCE (2021) Infrastructure report card. American Society of Civil Engineers
3. Kim B, Cho S (2020) Automated multiple concrete damage detection using instance segmentation deep learning model. Appl Sci. https://doi.org/10.3390/app10228008
4. Dong CZ, Catbas FN (2020) A review of computer vision-based structural health monitoring at local and global levels. Struct Heal Monit 20(2):692–743. https://doi.org/10.1177/1475921720935585
5. Spencer BF Jr, Hoskere V, Narazaki Y (2019) Advances in computer vision-based civil infrastructure inspection and monitoring. Engineering 5(2):199–222. https://doi.org/10.1016/j.eng.2018.11.030
6. Hutchinson TC, Chen Z (2006) Improved image analysis for evaluating concrete damage. J Comput Civ Eng 20(3):210–216. https://doi.org/10.1061/(ASCE)0887-3801(2006)20:3(210)
7. Medeiros FN, Ramalho GL, Bento MP, Medeiros LC (2010) On the evaluation of texture and color features for nondestructive corrosion detection. EURASIP J Adv Signal Process. https://doi.org/10.1155/2010/817473
8. Hoang ND, Nguyen QL, Tran XL (2019) Automatic detection of concrete spalling using piecewise linear stochastic gradient descent logistic regression and image texture analysis. Complexity 20(5):536–572. https://doi.org/10.1155/2019/5910625
9. Kim H, Ahn E, Cho S, Shin M, Sim SH (2017) Comparative analysis of image binarization methods for crack identification in concrete structures. Cem Concr Res 99:53–61. https://doi.org/10.1016/j.cemconres.2017.04.018
10. Dawood T, Zhu Z, Zayed T (2017) Machine vision-based model for spalling detection and quantification in subway networks. Autom Const 81:149–160. https://doi.org/10.1016/j.autcon.2017.06.008
11. Xu Y, Bao Y, Chen J, Zuo W, Li H (2019) Surface fatigue crack identification in steel box girder or bridges by a deep fusion convolutional neural network based on consumer-grade camera images. Struct Heal Monit 18(3):653–674. https://doi.org/10.1177/1475921718764873
12. Zhang L, Yang F, Zhang YD, Zhu YJ (2016) Road crack detection using deep convolutional neural network. In: Proceedings of 2016 IEEE International Conference on Image Processing (ICIP), 2016 Sep 25–28; Phoenix, AZ, USA. https://doi.org/10.1109/ICIP.2016.7533052
13. Cha YJ, Choi W, Buyukozturk O (2017) Deep learning-based crack damage detection using convolutional neural networks. Compt Aid Civ Infr Eng 32(5):361–378. https://doi.org/10.1111/mice.12263
14. Quqa S, Martakis P, Movsessian A et al (2022) Two-step approach for fatigue crack detection in steel bridges using convolutional neural networks. J Civil Struct Health Monit 12:127–140. https://doi.org/10.1007/s13349-021-00537-1
15. Kim B, Cho S (2018) Automated vision-based detection of cracks on concrete surfaces using a deep learning technique. Sens 18(19):3452. https://doi.org/10.3390/s18103452
16. Rajadurani RS, Kang ST (2021) Automated vision-based crack detection on concrete surfaces using deep learning. Appl Sci. https://doi.org/10.3390/app11115229
17. Yein LP, Kim B, Cho S (2018) Image-based spalling detection of concrete structures using deep learning. J Korea Conc Inst 30:91–99. https://doi.org/10.4334/jkci.2018.30.1.091
18. Savino P, Tondolo F (2021) Automated classification of civil structure defects based on convolutional neural network. Front Struct Civ Eng 15:305–317. https://doi.org/10.1007/s11709-021-0725-9
19. Kruachottikul P, Cooharojananone N, Phanomchoeng G et al (2021) Deep learning-based visual defect-inspection system for reinforced concrete bridge substructure: a case of Thailand's department of highways. J Civil Struct Health Monit 11:949–965. https://doi.org/10.1007/s13349-021-00490-z
20. Cha YJ, Choi W, Suh G, Mahmoudkhani S, Buyukozturk O (2018) Autonomous structural visual inspection using region based deep learning for detecting multiple damage types. Compt Aid Civ Infr Eng 33:731–747. https://doi.org/10.1111/mice.12334
21. Wang N, Zhao Q, Li S, Zhao X, Zhao P (2018) Damage classification for masonry historic structures using convolutional neural networks based on still images. Compt Aid Civ Infr Eng 33:1073–1089. https://doi.org/10.1111/mice.12411
22. Xue Y, Li Y (2018) A fast detection method via region based fully convolutional neural networks for shield tunnel lining defects. Compt Aid Civ Infr Eng 33:638–654. https://doi.org/10.1111/mice.12367
23. Zhou Q, Ding S, Qing G, Hu J (2022) UAV vision detection method for crane surface cracks based on faster R-CNN and image segmentation. J Civ Str Health Mon. https://doi.org/10.1007/s13349-022-00577-1
24. Zhang A, Wang KCP, Li B, Yang E, Dai X, Peng Y, Fei Y, Liu Y, Li JQ, Chen C (2017) Automated pixel-level pavement crack detection on 3D asphalt surfaces using deep-learning network. Compt Aid Civ Infr Eng 32:805–819. https://doi.org/10.1111/mice.12297

25. Zhu J, Song J (2020) Weakly supervised network based intelligent identification of cracks in asphalt concrete bridge deck. Alex Eng J. https://doi.org/10.1016/j.aej.2020.02.027
26. Ji J, Wu L, Chen Z, Yu J, Lin P, Cheng S (2018) Automated pixel-level surface crack detection using U-Net. In: Kaenampornpan M, Malaka R, Nguyen D, Schwind N (eds) Multi-disciplinary trends in artificial intelligence. MIWAI 2018. Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-030-03014-8_6
27. Zhou Q, Qu Z, Ju F (2022) A multi-scale learning method with dilated convolutional network for concrete surface cracks detection. IET Image Process 16:1389–1402. https://doi.org/10.1049/ipr2.12417
28. Ni FT, Zhang J, Chen ZQ (2019) Pixel level crack delineation in images with convolutional feature fusion. Struct Control Health Monit. https://doi.org/10.1002/stc.2286
29. Feng C, Zhang H, Wang H, Wang S, Li Y (2020) Automatic pixel-level crack detection on dam surface using deep convolutional network. Sensors. https://doi.org/10.3390/s20072069
30. Yang X, Li H, Yu Y, Luo X, Huang T (2018) Automatic pixel level crack detection and measurement using fully convolutional network. Compt Aid Civ Infr Eng 33(12):1090–1109. https://doi.org/10.1111/mice.12412
31. Pozzar S, Azar E, Chamberlain Pravia Z, Dalla Rosa F (2021) Semantic segmentation of defects in infrared thermographic images of highly damaged concrete structures. J Perform Constr Facil. https://doi.org/10.1061/(ASCE)CF.1943-5509.0001541
32. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018 Sep 8–14; Munich, Germany. https://doi.org/10.48550/arXiv.1802.02611
33. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L (2018) MobileNetV2: inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. https://doi.org/10.48550/arXiv.1801.04381
34. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.48550/arXiv.1610.02357
35. Kaiming H, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. https://doi.org/10.48550/arXiv.1512.03385

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.