applied sciences

MDPI

Article

Automated Multiple Concrete Damage Detection Using Instance Segmentation Deep Learning Model

Byunghyun Kim and Soojin Cho *

Department of Civil Engineering, University of Seoul, 163 Seoulsiripdae-ro, Dongdaemun-gu, Seoul 02504, Korea; shdik2002@uos.ac.kr

* Correspondence: soojin@uos.ac.kr; Tel.: +82-10-9401-6041

Received: 6 October 2020; Accepted: 10 November 2020; Published: 12 November 2020

Abstract: In many developed countries with a long history of urbanization, there is an increasing need for automated computer vision (CV)-based inspection to replace conventional labor-intensive visual inspection. This paper proposes a technique for the automated detection of multiple concrete damage based on a state-of-the-art deep learning framework, Mask R-CNN, developed for instance segmentation. The structure of Mask R-CNN, which consists of three stages (region proposal, classification, and segmentation) is optimized for multiple concrete damage detection. The optimized Mask R-CNN is trained with 765 concrete images including cracks, efflorescence, rebar exposure, and spalling. The performance of the trained Mask R-CNN is evaluated with 25 actual test images containing damage as well as environmental objects. Two types of metrics are proposed to measure localization and segmentation performance. On average, 90.41% precision and 90.81% recall are achieved for localization and 87.24% precision and 87.58% recall for segmentation, which indicates the excellent field applicability of the trained Mask R-CNN. This paper also qualitatively discusses the test results by explaining that the architecture of Mask R-CNN that is optimized for general object detection purposes, can be modified to detect long and slender shapes of cracks, rebar exposure, and efflorescence in further research.

Keywords: multiple damage; deep learning; mask r-cnn; concrete crack; efflorescence; rebar exposure; spalling

1. Introduction

Most of highly developed countries have difficulties in managing an increasing number of old civil structures. Currently, regular visual inspection is a widely used maintenance approach that considers the importance and scale of the structures. In South Korea, up to 270,000 structures require regular inspection annually, although the budget and number of inspectors are decreasing gradually. A 2017 report by American Society of Civil Engineers (ASCE) stated that 39% of all US bridges were over 50 years old, and the bridge maintenance costs were estimated at about $123 billion [1]. Thus, many industrialized countries are now seeking ways to inspect civil structures with reduced costs and labor without losing efficiency.

For concrete structures, important types of concrete damage are designated to be inspected on a regular basis according to the aging mechanism of concrete. For most civil engineers, cracks, efflorescence, spalling, and rebar exposure are usually considered serious risks to the structural safety of concrete structures. A crack, which is a line on the concrete surface along with the concrete splits, is the most common type of concrete damage. A crack causes not only apparent damage on the concrete surface but also moisture penetration that leads to rebar corrosion. Efflorescence is a crystalline deposit, usually brightly colored, that forms on or near the surface of concrete structures because of concrete hydration. As a sign of concrete carbonation, efflorescence is used as one of the important indicators to

evaluate the safety of the structure. Spalling is a phenomenon in which parts of the concrete surface detach, owing to rebar expansion caused by corrosion. The spalling of the concrete cover usually accompanies rebar exposure. Rebar exposure leads to corrosion by air moisture and can induce further corrosion of the rebar under the surface, which can reduce the loading capacity of the structure.

Although the abovementioned concrete damage is recognizable by the naked eye, several shortcomings of visual inspection indicate a necessity of CV-based inspection to replace visual inspection. First, visual inspection is highly dependent on expertise as well as the condition of the inspectors. Second, visual inspection consumes a considerable amount of time and cost as an inspector carefully looks though a large area of concrete surface. Third, visual inspection is hazardous when assessing dangerous or restricted areas of structures. Breakdowns of aerial work platforms or catwalks are not rare events and can involve many casualties [2,3]. To overcome the shortcomings of visual inspection, many research studies have sought to replace visual inspection with CV-based inspection using remote imaging devices such as drones and cameras. CV-based inspection can be more reliable and objective as it uses a designated computer algorithm. With a proper algorithm, the inspection can be automated by a computer, and the time and cost of inspection can be significantly reduced. Furthermore, remote imaging reduces the risks to inspectors in assessing dangerous areas.

Given these advantages over conventional visual inspection, many studies have reported on CV-based inspection, and most of them focused on concrete crack detection. A variety of studies reported on crack detection methods ranging from conventional approaches using algorithm-based image processing [4–10] to recent attempts that adopted deep learning techniques [11–15]. Abdel-Qader et al. [4] conducted a comparison of various image processing algorithm such as fast Haar transform for crack detection algorithm. Lecompte et al. [5] presented crack width method with an application of two different camera techniques. Yamaguchi et al. [6] proposed a crack detection method considering direction of crack and tested its performance on images captured from real structures. Rabah et al. [7] used terrestrial laser scanning method to measure cracks in 3D space. Torok et al. [8] demonstrated a new automated crack identification method using 3D model constructed by structural from motion techniques. Prasanna et al. [9] proposed a novel crack detection algorithm, spatially tuned robust multi-feature classifier and tested the algorithm with images captured by a pavement scanning robot. Kim et al. [10] compared various crack detection method and conducted parametric study for each image processing method.

In concrete crack assessment, it is also important to measure length and width of concrete cracks to evaluate structural integrity of concrete structures. Besides the developments of concrete crack detection methods using image processing, many researchers reported computer vision-based crack quantification methods. Liu et al. [16] proposed crack assessment procedure using 3D point cloud and measured crack width using image skeletonization techniques. Kim and Cho [17] proposed crack quantification methods that refines crack area detected by a deep learning model and measure the width using sequential image processing. Ni et al. [18] combined conventional Otsu thresholding and deep learninig technique for crack width measurement.

As mentioned above, it is important not only to evaluate cracks, but also to detect and quantify damages such as efflorescence, spalling, and rebar exposure for the integrity evaluation of concrete structures. Research that tried to detect concrete damages used various types of equipment and presented methodologies that detects the appearance or chemical properties of the damage.As efflorescence is recognizable by the naked eye, the number of studies on developing an efflorescence detection technique is limited. However, research studies on detecting efflorescence using special devices such as a hyperspectral camera or a laser scanner (i.e., LiDAR) have often been reported [19–22]. Damage types such as spalling or potholes are detected based on the dent features that appear when they are captured in images [23–27]. Rebar exposure is often accompanied by cracks and/or spalling. Therefore, there have been studies that detected rebar exposure with other concrete damage, rather than single-class detection of rebar exposure [28].

Despite many studies, there have been few examples that apply CV-based inspection techniques to real structures that have diverse types of objects in the vicinity such as accessories (e.g., electrical lines and drainage pipes) or surrounding objects (e.g., nearby passengers and vegetation). In particular, the separation of similar objects from the damage may be poorly performed, although the features are carefully crafted to distinguish damage from the objects [6,9,29,30]. To overcome the limitations of human-crafted feature-based inspection methods, several damage detection methods have used deep learning techniques, which train a computer to automatically define features of damage from a large number of images [11–15] . Cha et al. [11] detected multiple types of damage including concrete cracks, steel corrosion, bolt corrosion, and steel delamination using Faster R-CNN (Region-based Convolutional Neural Network) [31], and the result displayed the damage using rectangular bounding boxes without illustrating the damage shapes. Wang et al. [12] also used Faster R-CNN [31] to detect multiple types of concrete damage in masonry historic structures. Wu et al. [13] examined the possibility of unmanned-aerial-vehicle-based road damage inspection and discussed the current limitations owing to the lack of a reliable dataset. Xue and Li [14] successfully detected concrete damage in the form of bounding boxes in a tunnel lining in dark illumination conditions. Li et al. [15] implemented a fully convolutional network (FCN) [32] to detect multiple concrete damage such as cracks, spalling, efflorescence, and holes. Although studies have shown the successful application of deep learning techniques to detect damage from images, they have rarely shown the detectability against surrounding objects.

This study proposes a concrete damage detection method using an instance segmentation deep learning model called Mask R-CNN (Mask and Region-based Convolutional Neural Network) [33]. The scope of this study is on detecting multiple types of concrete damages and does not include the quantification of detected damages, which has already been proposed in the previous paper by the authors [17]. This method alters conventional visual inspection in the field environment with surrounding objects. Mask R-CNN is a combined deep learning model of Faster R-CNN [31] and a FCN [32]. Faster R-CNN is an object detection model that detects objects in an image by locating the bounding boxes of the objects, and FCN is a type of semantic segmentation model that detects objects in a form of segmentation by classifying every pixel in an image. As a combination of Faster R-CNN and FCN, Mask R-CNN achieved high accuracy in detecting 20 types of objects after training with more than 200,000 images in the COCO dataset [34]. In this study, a pretrained Mask R-CNN is fine-tuned using a concrete image dataset to detect four types of major concrete damage. Beyond the detection of concrete cracks [17], spalling, rebar exposure, and efflorescence are selected as types of damage to be detected by the Mask R-CNN model to alter the conventional visual inspection of concrete structures. This study briefly introduces the overall structure of the Mask R-CNN to present how to localize an object in an image and how to segment the localized object. The next section explains the training procedure of the Mask R-CNN in detail, including the labeling of the training dataset considering typical shapes of four types of concrete damage (cracks, spalling, rebar exposure, and efflorescence) and fine-tuning of the Mask R-CNN parameters. Then, the performance of the trained Mask R-CNN is verified for images taken in the field environment with surrounding objects.

Fully Convolutional Networks

2. Mask and Region-Based Convolutional Neural Network (Mask R-CNN)

Mask R-CNN [33] is an extension of Faster R-CNN [31] for instance segmentation. Instance segmentation detects individual objects of interest in an image and segments the shapes of the detected objects. The structure of Mask R-CNN is a combination of solvers for both object detection and segmentation, namely Faster R-CNN and a mask branch. Faster R-CNN first detects objects of interest in an image and mask branch and subsequently finds the shapes of the detected objects. Faster R-CNN is divided into two stages: region proposal (which determines the region in which an object is located) and classification (which classifies the region into predefined labels of objects). For each region determined by Faster R-CNN, the mask branch finds the shape of the object in the region. Thus, the structure of Mask R-CNN is composed of three stages: region proposal, classification,

and segmentation. This is depicted in Figure 1. The following paragraph explains each step of Mask R-CNN in more detail.

2.1. Feature Extraction by Convolutional Neural Network and Feature Pyramid Network

For feature extraction, Mask R-CNN takes an input image through convolutional neural network (CNN), which is depicted as five sequential gray boxes with varying width and height in Figure 1. CNN is an aggregation of filters with trainable weights and bias. The weights and bias decides features to be emphasized or dismissed in an input image. While training, the weights and bias are optimized to carry out a given task on a given dataset. In Mask R-CNN, ResNet-101 [35] is used for major feature extractor. The gray boxes from C1 to C5 in Figure 1 shows the 5 stages of ResNet-101. Their increasing width and decreasing height shows the increasing depth and the decreasing resolution of convolutional layers in ResNet-101. The feature maps created by CNN are fed to feature pyramid network (FPN) [36] in the next step. Since most of CNN downsizes an input image during feature extraction, FPN restore spatial information that are potentially lost while downsizing. The feature maps extracted by FPN is restored to five different sizes in order to detect objects of different sizes. To express this conceptually in Figure 1, the size of the blue feature map from P5 to P1 is represented with increasing height.

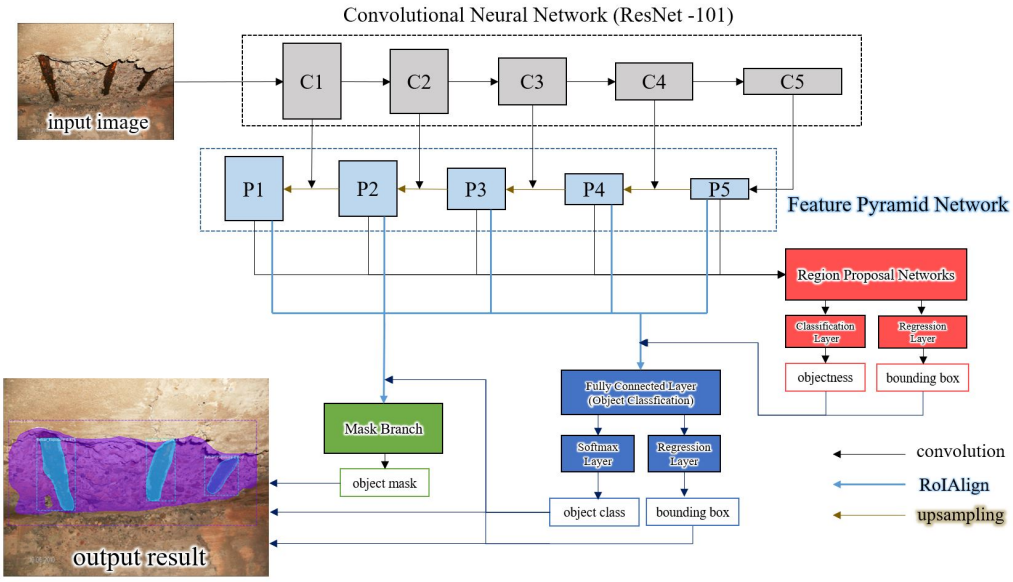


Figure 1. Overall architecture of Mask R-CNN.

2.2. Region Proposal Network

From the feature map extracted by CNN and FPN, Mask R-CNN detects an object and estimates shape of detected objects. Region Proposal Network (RPN), which is highlighted with red color in Figure 1, detects the existence and location of objects in an image. The core idea of RPN is to determine the existence and location of objects in the form of bounding boxes using a small FCN as an extension of the normal CNN [31]. In the step of the region proposal, an n $\times$ n sliding window scans multiple feature maps created by FPN and feed the scanning result in the intermediate layer. The intermediate layer is followed by two layers: the classification layer and regression layer. They are created by a 1 $\times$ 1 convolution from the intermediate layer. The classification layer and regression layer have 2k and 4k elements, respectively, for each sliding window that is scanned the feature map, where k denotes the number of anchors of a sliding window. The role of the anchor is to create various-sized candidate boxes from each sliding window on the feature map. The 2k elements of the classification layer represent object/non-object classes of the anchors, and the 4k elements of the regression layer

represent the horizontal and vertical (i.e., x and y) coordinates of the top left corners of the bounding boxes and the width and height of the anchors.

2.3. Object Classification and Bounding Box Refinement

Using the bounding box information and its objectness, Mask R-CNN determines object class and refine the given bounding box one more time. In this step, RoIAlign extracts features with high accuracy in the region of interest (RoI) proposed by RPN. In general, the feature map extracted by CNN tends to have a low resolution owing to the downscaling of the convolutional layers. Because of the low resolution, the spatial information of the objects in the input image is lost, resulting in decreased detection accuracy. RoIAlign is designed to extract features with more accurate spatial information using bilinear interpolation. The features extracted by RoIAlign are input to a fully connected layer first and then transfered to the box-classification layer, which classifies the detected object by the RPN, and the box-regression layer, which corrects the bounding box of the proposed object more accurately. The box classification layer uses softmax function to classify object in the given bounding box. In this study, the box-classification layer is designed to have four outputs of softmax for cracks, efflorescence, spalling, and rebar exposure for multiple concrete damage detection. The box-regression layer has four outputs indicating the position of the bounding box, similar to the regression layer in RPN. More details of RoIAlign are provided in the works of He et al. [33]. For more information on the box-classification layer and the box-regression layer, a paper regarding Faster R-CNN [31] can be referred.

2.4. Mask Branch

In the final step of Mask R-CNN, mask branch estimates the shape of the detected object from the previous procedures. The mask branch trains the shape of the object inside the bounding box regardless of its class. Therefore, the mask branch performs a pixel-to-pixel binary classification that distinguishes foreground objects from background objects in the bounding box. Please note that the mask branch achieves high accuracy without considering the class of the object. It was confirmed in the work of He et al. [33] that the accuracy was not significantly improved in the prediction when considering the class of the object. Originally, the mask branch performed four 3×3 convolutions to maintain the resolution, a single 2×2 deconvolution on the 14×14 feature map extracted by RoIAlign, and outputted a mask of 28×28. The 28×28 resolution of the output layer was sufficient to predict the shapes of the objects in the COCO dataset [34] for which Mask R-CNN was originally trained.

2.5. Loss Functions

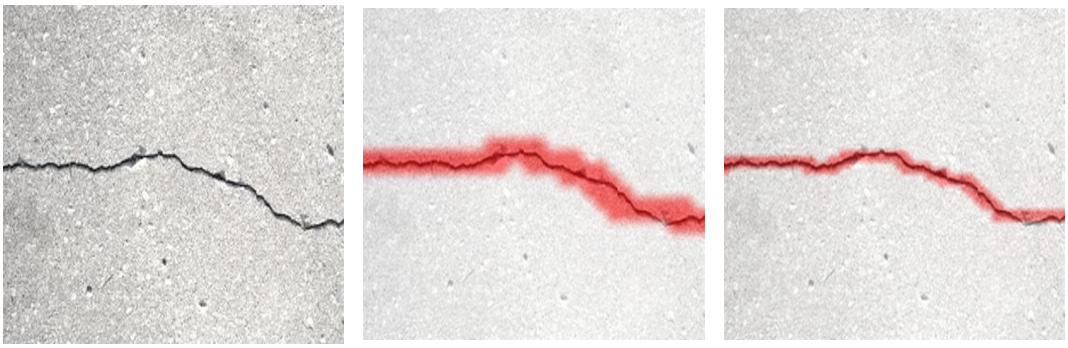
To train the layers of the Mask R-CNN described above, loss functions are designed considering the task of each output layer. The five layers in Figure 1 (classification layer and regression layer in the RPN, box-classification layer and box-regression layer in the classification stage, and the output layer of the mask branch) use loss functions for the training. In the RPN, the classification layer uses a logarithmic loss for binary classification that distinguishes objects from backgrounds, and the regression layer uses a smooth L1 loss to optimize for estimated four elements representing the bounding box locations. In the classification stage, box classification uses a log loss to classify all foreground objects and backgrounds, and the box-regression layer uses a smooth L1 loss to locate the bounding boxes as the regression layer of the RPN does. The mask branch is trained using a binary cross-entropy loss that distinguishes objects from backgrounds in the bounding box. In this study, the five loss functions are set to be simultaneously optimized during the training. Girshick et al. [37] and He et al. [33] can be referred for detailed description of each loss function.

2.6. Model Modification for Optimal Training of Mask R-CNN

a contribution

In this study, one additional 2×2 deconvolution is added to the original mask branch proposed by He et al. [33] to output a higher (i.e., 56×56) resolution, which outperforms in accurately

segmenting cracks that generally occupy a few pixels in the bounding boxes and efflorescence that has amorphous shapes. Figure 2 shows an example of finer mask representation by employing additional deconvolution in the original mask branch. Figure 2a is an example image with a crack. Figure 2b shows an exaggerated crack area owing to the low resolution of original mask representation, and Figure 2c shows a finer mask representation with a higher mask resolution of 56 × 56. As shown in Figure 2, it is worth increasing the resolution of the mask branch despite a trivial increase in the computational cost.



(**a**) (**b**) (**c**)

Figure 2. Resolution of mask representation comparison: (**a**) original image of concrete crack, (**b**) crack area representation by 28 × 28 mask (in red), and (**c**) crack area representation by 56 × 56 mask (in red).

3. Training Mask R-CNN for Multiple Concrete Damage Detection

The Mask R-CNN used in this study is a model pretrained using a COCO dataset [34] with ResNet-101 [38] as a backbone network. After pretraining, the Mask R-CNN model was fine-tuned using images of concrete damage. Because the resolution of the mask branch is readjusted from 28 × 28 to 56 × 56, the weights of the mask branch were randomly initialized and the other weights were initialized using the pretrained result.

To train the pretrained Mask R-CNN model for this study, 765 images in total were collected from the Internet or taken from actual structures including buildings and bridges. The number of training images was 319, 196, 249, and 237 for cracks, efflorescence, rebar exposure, and spalling, respectively. The resolutions of the images differed, ranging from 600 × 600 to 2000 × 2000. Photoshop, a commercial graphic software package, was used to annotate the damage on the images. Cracks, efflorescence, spalling, and rebar exposure were indicated with red, green, purple, and cyan for their respective regions, as shown in Figure 3. Please note that different from the other damage, cracks were indicated for regions including cracks and adjacent pixels, which resulted in high accuracy when quantifying the crack widths [17]. After annotation of all training images, the annotation format was converted to COCO dataset [34] using a format-converting module called Pycococreator [39] to be input to the Mask R-CNN model. The mask branch trained the annotated regions using different colors to segment the four types of damage, while the RPN, box-classification layer, and box-regression layer trained the bounding boxes marked with dotted lines. Please note that the cracks and rebar exposure with long and slender shapes were annotated using small overlapping bounding boxes to minimize the portion of intact regions in the bounding boxes, which can significantly deteriorate the performance of the model. Cha et al. [11] did not explicitly mention that small bounding boxes are required to annotate cracks, but their results infer the need for small bounding boxes. Griffiths and Boehm [40] mentioned this issue when detecting railways in an image.

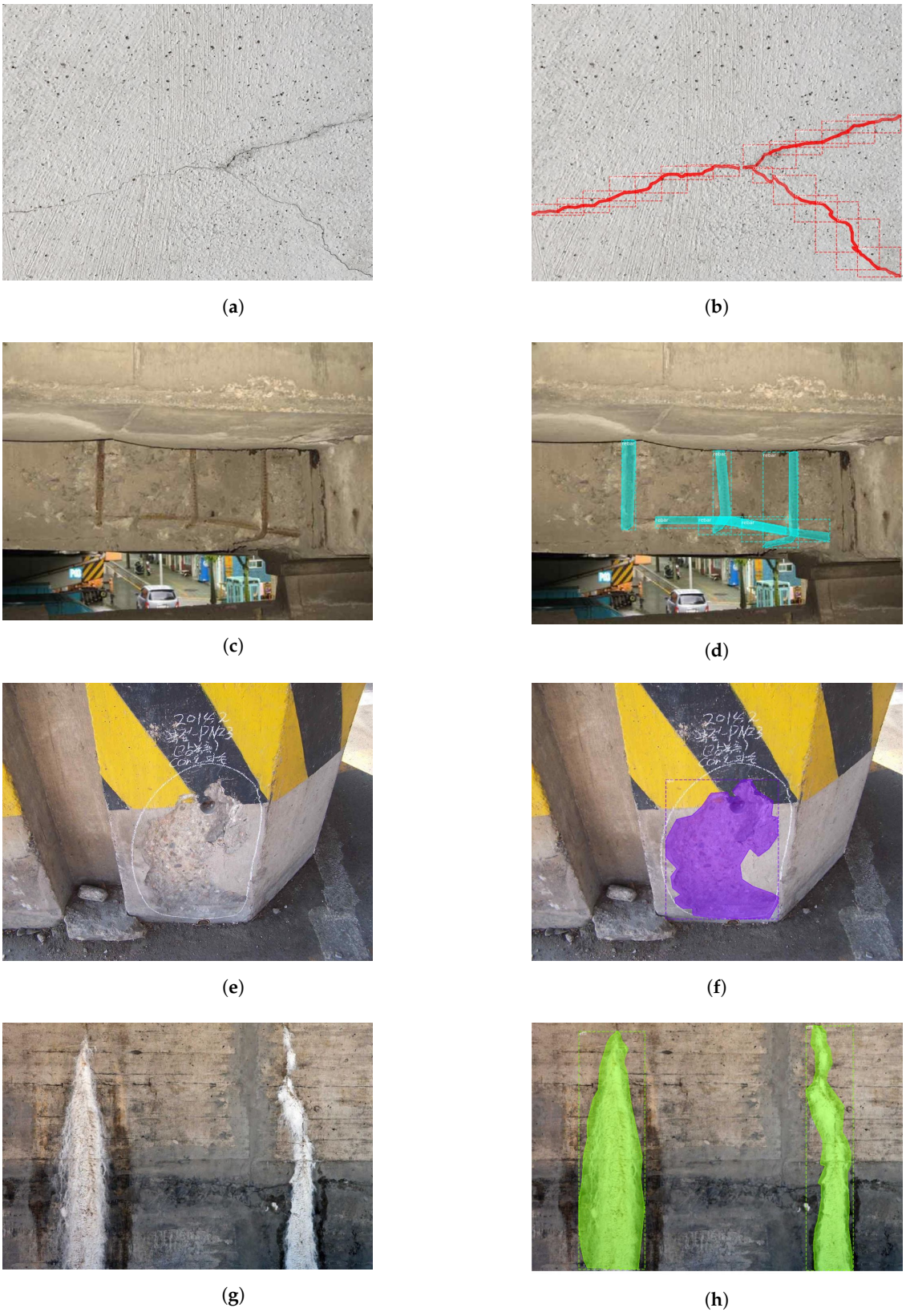


Figure 3. Examples of original training images (left column) and damage annotation (right column): (**a**,**b**) crack, (**c**,**d**) rebar exposure, (**e**,**f**) spalling, and (**g**,**h**) efflorescence.

The training was performed on a high-performance computer designed for deep learning. The computer had two Intel Xeon CPUs and four GeForce GTX 1080Ti processors with 11 GB of graphic

memory and 64 GB of RAM. The detailed parameters in the training were as follows. The minibatch size was set as 4 to allocate single images to a GPU. To prevent the trained model from overfitting the given training data, a total of four image augmentations were randomly selected and applied to the images in each iteration. The four types of image augmentation were flipping left-right, flipping up-down, rotating 90°, and zero padding of any number from 0 to 100 on the four corners of the image. Each example is shown in Figure 4. The augmentations varied the location of the bounding boxes to enhance the performance of the RPN and the box-regression layer. A stochastic gradient descent enhanced by gradient clipping was used as an optimizer for training; and its learning rate, weight decay, and learning momentum were set as 0.001, 0.0001, and 0.9, respectively. All images were resized to 1024 × 1024 considering the memory of the GPU, and the anchor sizes were set to 32, 64, 128, 256, and 512, which were the optimized sizes of the 1024 × 1024 images. The resize aims to fit input images into the dimension of the feature-extracting convolutional network of Mask R-CNN. Once the detection procedure of Mask R-CNN is completed, the images are resized to original size and the detected damages are transformed to corresponding pixels. The authors limited the size of training and test images to under 2000 pixels to avoid pixel information loss by resizing. The width-height ratios of the original images were maintained when resizing, while zero padding was used to unify the image size. The training was executed for 50 epochs at 191 iterations per epoch using the parameters described above.

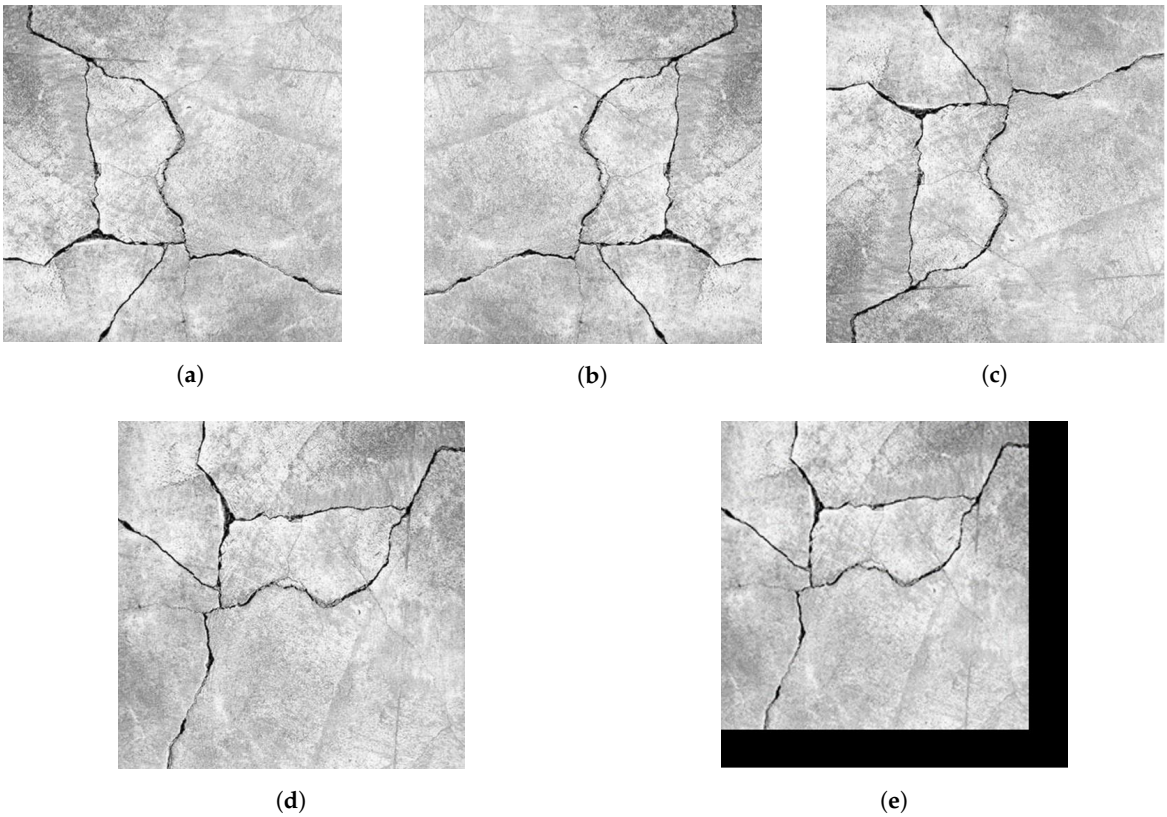


Figure 4. Examples of image augmentation: (**a**) original image of crack, (**b**) flipping left-right, (**c**) flipping up-down, (**d**) rotating 90°, and (**e**) zero padding at right and bottom corner.

4. Experimental Evaluation

4.1. Damage Detection Results

An evaluation of the trained model was carried out using 25 actual images containing damage as well as various environmental objects. These were obtained from the Internet or captured from real structures. Among the 25 images, 5, 5, 5, and 10 images contain cracks, efflorescence, rebar

exposure, and/or spalling and their collocations (e.g., spalling with rebar exposure, spalling with cracks, and efflorescence with cracks), respectively. Before applying the trained model, all test images were resized to 1024 × 1024, which is the input size of the Mask R-CNN model, regardless of their own resolutions. This was accomplished to infer the images by a single application of the trained model. The detection results only includes the bounding boxes with confidence scores higher than 0.5, which is the marginal value for the optimal object detection with Faster R-CNN-based models [41]. In Figures 5–8, some examples of damage detection are illustrated and compared with the ground truths. Figures 5–7 show examples for single types of damage (i.e., cracks and efflorescence), while Figure 8 shows collocated damage. In the figures, the ground truths of the damage are marked in the left columns, and they are visually compared with the detection results in the right columns. The ground truths and the detected bounding boxes and masks are colored as follows: red for cracks, green for efflorescence, cyan for rebar exposure, and purple for spalling. To consider the field environment with other surrounding objects, all test images were selected to include various objects as well as damage. This includes a stair railing in Figure 6, trash in Figure 7, and scaffolding in Figure 8. Please note that the results of the other tested images are listed in Table 1.

4.2. Evaluation Using Performance Measures

The performance of the trained Mask R-CNN model was measured using two well-known measures in the deep learning field: precision and recall. Precision and recall are calculated based on true positives (TPs), false positives (FPs), and false negatives (FNs). Generally, precision and recall are determined using the number of TPs, FPs, and FNs as follows:

$$\text{Precision} = \frac{\#TP}{\#TP + \#FP} \tag{1}$$

$$\text{Recall} \frac{\#TP}{\#TP + \#FN} \tag{2}$$

where # denotes the number of TPs, FPs, and FNs, respectively.

In this study, precision and recall are determined for two different purposes: localization and segmentation, which are the main tasks of an instance segmentation model. For localization, a resulting mask whose overlap with a ground truth is equal to or higher than 70% is categorized as a TP, a resulting mask whose overlap with a ground truth is lower than 70% is categorized as an FP, and a ground truth whose overlap is lower than 70% with a resulting mask is categorized as an FN. For segmentation, the number of pixels in the resulting masks overlapping with the ground truths are used to determine the TPs, FPs, and FNs. The overlapping pixels are categorized as TPs, the pixels of resulting masks that do not overlap with the ground truths are categorized as FPs, and the pixels of ground truths that do not overlap with the resulting masks are categorized as FNs. Based on the definitions of TPs, FPs, and FNs, the precision and recall can also be calculated using Equations (1) and (2), which indicate the performance of the trained model in localizing and segmenting the damage.

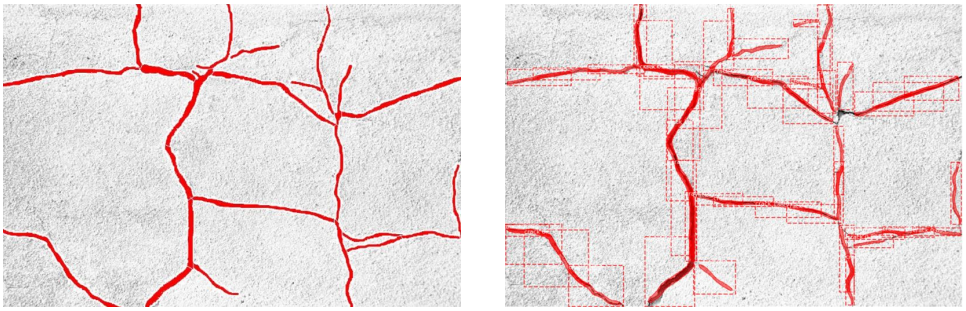
Table 1 lists two types of precision and recall calculated for the 25 test images. With regard to the localization metrics, the trained Mask R-CNN achieved an average of 90.43% precision and 90.81% recall for the entire test set. The results of damage types indicated that the trained Mask R-CNN achieved an average of 98.31% precision and 95.83% recall for cracks, 80% precision and 100% recall for efflorescence, 92.37% precision and 95.55% recall for spalling and rebar exposure, and 86.73% precision and 76.66% recall for multiple collocated damage. In segmentation metrics, the trained Mask R-CNN achieved an average of 87.24% precision and 87.58% recall for the entire test set. The results of damage types indicated that the trained Mask R-CNN achieved an average of 88.92% precision and 85.63% recall for cracks, 91.15% precision and 95.18% recall for efflorescence, 88.05% precision and 89.30% recall for spalling and rebar exposure, and 89.47% precision and 89.74% recall for multiple collocated damage.

Table 1. Evaluation results for trained Mask R-CNN for multiple concrete damage detection.

Image No.	Localization Purpose		Segmentation Purpose		Remarks
	Precision	Recall	Precision	Recall	
Crack-1	93.75	100	89.97	92.37	Figure 5a
Crack-2	100	100	91.34	87.88	
Crack-3	97.82	79.16	86.28	81.81	Figure 5b
Crack-4	100	100	87.97	90.94	
Crack-5	100	100	89.02	75.15	
Crack: average	98.31	95.83	88.92	85.63	
Efflorescence-1	100	100	95.20	93.65	
Efflorescence-2	100	100	93.91	88.73	
Efflorescence-3	50	100	78.28	94.8	Figure 6b
Efflorescence-4	50	100	89.33	98.83	
Efflorescence-5	100	100	99.02	99.89	Figure 6a
Efflorescence: average	80	100	91.15	95.18	
Spalling/Rebar-1	100	100	96.16	99.44	Figure 7a
Spalling/Rebar-2	95.23	77.78	87.51	89.36	
Spalling/Rebar-3	100	100	94.66	91.11	
Spalling/Rebar-4	100	100	96.38	82.82	
Spalling/Rebar-5	66.67	100	65.56	83.77	Figure 7b
Spalling/Rebar: average	92.37	95.55	88.05	89.30	
Collocated damages-1	100	83.33	87.59	86.01	
Collocated damages-2	100	66.67	88.44	68.92	Figure 8d
Collocated damages-3	87.5	80	88.65	84.80	
Collocated damages-4	85.71	100	67.67	88.93	
Collocated damages-5	100	100	90.86	91.72	Figure 8a
Collocated damages-6	75	85.71	74.84	79.48	
Collocated damages-7	79.31	59.25	83.14	78.85	Figure 8c
Collocated damages-8	100	83.33	92.80	77.70	
Collocated damages-9	83.33	75	85.58	84.51	
Collocated damages-10	96	80	89.47	89.74	Figure 8b
Collocated damages: average	86.73	76.66	85.16	82.06	
Overall average	90.41	90.81	87.24	87.58	

A visual investigation of the figures indicates the successful detection of multiple damages by the trained Mask R-CNN model. In Figure 5, the cracks were completely detected with minimal misses and no false alarms, although there are various nondamaged objects in the vicinity such as stains. In the multiple complex crack patterns in Figure 5b, a thin crack branch at the bottom right of the image was missed owing to the limitation of the resolution. In Figure 6, efflorescence was successfully detected, and the shapes were nearly identical to the ground truths. A false alarm was detected on a steel cantilevered support of the catwalk at the top right of Figure 6b owing to its bright gray color and complex shape. In Figure 7, severe spalling with rebar exposure was successfully detected except for one false alarm of cracks owing to the wavy pattern of the paint covering the concrete. In Figure 8, the trained model showed robust performance in detecting collocated multiple damage, while Figure 8c,d showed certain conditions that imped the accurate inference though. In Figure 8c, relatively narrow cracks around the spalling were missed, while wide cracks were successfully detected. In addition, some of the cracks in the middle of the right side of Figure 8c were missed because they had visual features similar to those of overlaid cement paste, which was trained as a noncrack object. In Figure 8d, the spalling could not be completely detected owing to the blocking of a scaffold in the line of sight. The visual investigation inferred that despite the robust performance of the trained model, the performance can worsen owing to some unexpected objects with visual features similar to the damage and interruptions when capturing the entire damage shape.

(**a**)



(**b**)

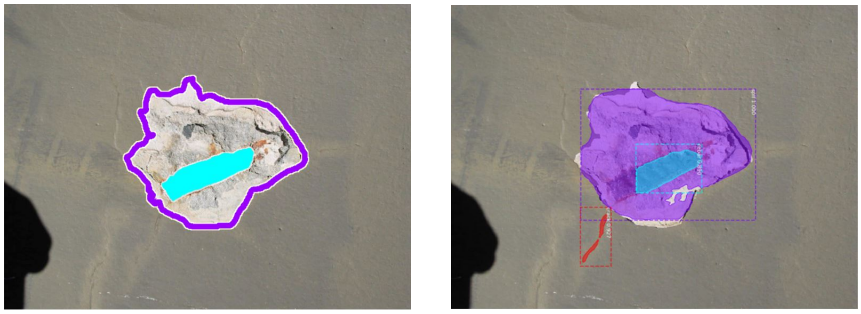
Figure 5. Examples of crack detection results (left column: original images with ground truth, right column: detection result of trained Mask R-CNN): (**a**) detection result of Crack-1 in Table 1, and (**b**) detection result of Crack-3 in Table 1.

(**a**)

(**b**)

Figure 6. Examples of efflorescence detection results (left column: original images with ground truth, right column: detection result of trained Mask R-CNN): (**a**) detection result of Efflorescence-5 in Table 1, and (**b**)detection result of Efflorescence-3 in Table 1.

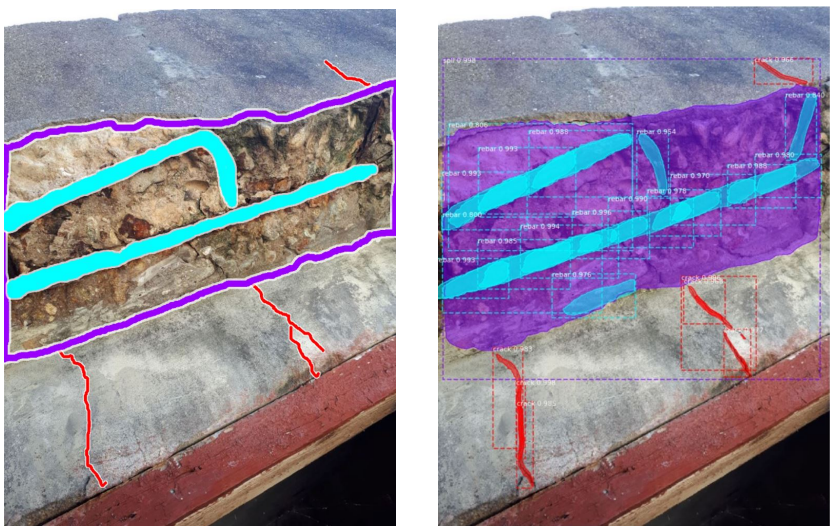
(**a**)



(**b**)

Figure 7. Examples of spalling and rebar exposure detection results (left column: original images with ground truth, right column: detection result of trained Mask R-CNN): (**a**) detection result of Spalling/Rebar-1 in Table 1, and (**b**) detection result of Spalling/Rebar-5 in Table 1.

(**a**)



(**b**)

Figure 8. *Cont.*

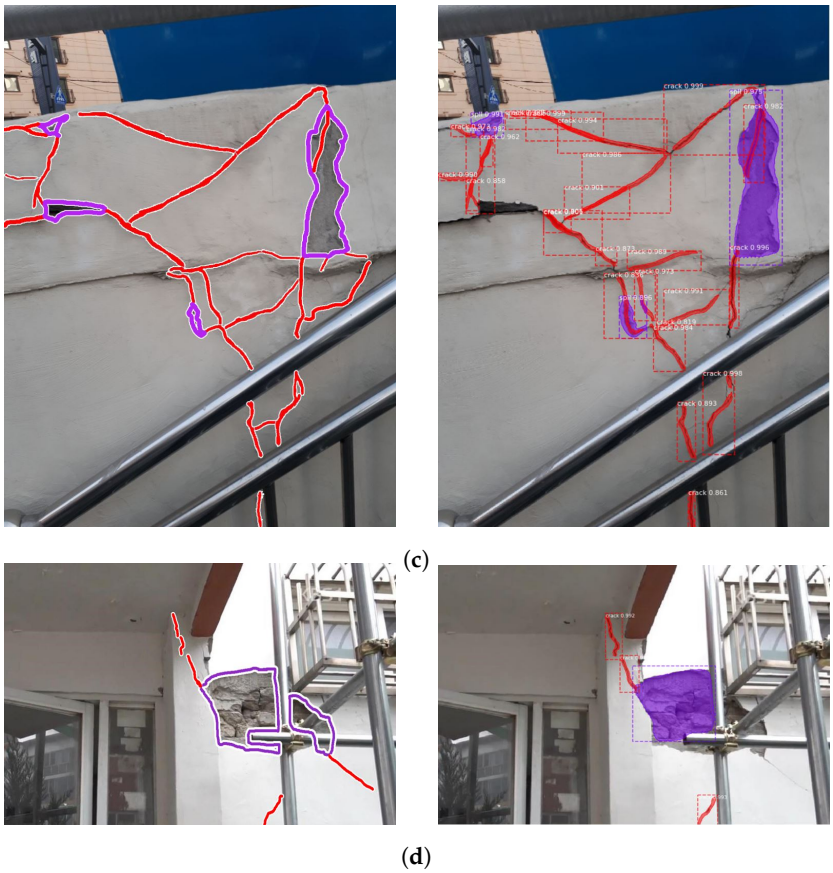


Figure 8. Examples of collocated damage detection results (left column: original images with ground truth, right column: detection result of trained Mask R-CNN): (**a**) detection result of Collocated damages-5 in Table 1, (**b**) detection result of Collocated damages-10 in Table 1, (**c**) detection result of Collocated damages-7 in Table 1 and (**d**)detection result of Collocated damages-2 in Table 1.

4.3. Two Possible Methods for Improved Accuracy

The structure of Mask R-CNN used in this study is a modified version of the work of He et al. [33], especially with regard to the mask branch. The COCO dataset for which the Mask R-CNN was developed contains 20 types of objects whose shapes are large, regular, and patterned, while the concrete damages studied in this paper are small, slender, irregular, and weakly patterned. For example, cracks are generally very thin and are a couple of pixels wide. Efflorescence can have either a radiated or downflow shape with unclear boundaries. Spalling may or may not contain exposed rebars, and their aggregate patterns may be strong or weak. Thus, the sizes of the network parts can be modified considering the characteristics of the damage for accurate localization and segmentation purposes.

For accurate localization, the size of the RPN can be increased. RPN, which serves a practical detection role in Mask R-CNN, was originally designed to speed up the slow region proposal process of a selective search. To improve the speed of the region proposal, the network size of the RPN is designed to be relatively small. However, the inference of the trained Mask R-CNN should not only be sufficiently fast but also accurate. To achieve better performance in detection accuracy, a good alternative is to increase the resolution or depth of the RPN or add additional networks to Mask R-CNN. The feasibility of this approach was investigated in the work of Cai and Vasconcelos [42] and can be applied to further research.

For accurate segmentation, the size of the mask branch can be further increased. The size of the mask branch is also relatively small compared to the backbone network ResNet-101 [38]. The mask branch originally had a low resolution because the performance of the mask branch with a 28×28 resolution is accurate enough to estimate object shapes in the COCO dataset [34]. However, in the case

of cracks, spalling, and efflorescence, their shapes are too irregular to be estimated by a small network with a 28 × 28 resolution. In this study, the resolution of the mask branch is set to 56 × 56 in advance to avoid poor detection results. Although the resolution of the mask branch is enhanced, there are still a few detection results with inaccurate mask estimation. For instance, the efflorescence detection result in Figure 6b missed the slender parts of the efflorescence, and the crack detection result in Figure 8b also missed small parts of cracks where the crack width was relatively narrow. To improve the quality of mask estimation, the resolution of the mask branch can be increased even higher, but this approach may have limitations when it comes to the computational cost. Therefore, a good solution is to apply a statistical image-processing-based approach in addition to simply increasing the resolution of the mask branch to improve the accuracy of damage detection.

5. Conclusions

This study presented an automated multiple concrete damage detection method called Mask R-CNN that is based on a state-of-the-art instance segmentation deep learning model. Mask R-CNN consists of three stages: region proposal, classification, and segmentation. This method is implemented with a backbone network ResNet-101 followed by an RPN and mask branch. The mask branch in this study had an enhanced resolution of 56 × 56 to improve the performance of the segmentation of amorphous concrete damage. Using a total of 765 images with annotations of damage (319, 196, 249, and 237 images for cracks, efflorescence, rebar exposure, and spalling, respectively), the training of Mask R-CNN occurred for 50 epochs on a workstation with four GPUs with image augmentation implemented.

The validation of the trained Mask R-CNN was carried out on 25 actual images that contained cracks, efflorescence, rebar exposure, and spalling with their collocation, along with various environmental objects. Two types of metric were proposed to validate the performance of localization and segmentation. For test images, the average precision was 90.41%, and the average recall was 90.81% for localization purposes. For segmentation, the trained Mask R-CNN achieved an average of 87.24% for precision and 87.58% for recall, which shows a good accuracy in the field environment. It was also suggested that the performance of the network can be improved by increasing the depth and resolution of the network architecture or by optimizing the network for slender-shaped objects such as cracks and rebar exposure. The experiment results showed the potential of the Mask R-CNN model to replace the existing visual inspection of concrete structures with high accuracy.

In future studies, the classes of the structural damage can be extended to not only the four concrete damages, cracks, efflorescence, rebar exposures, and spalling in this study, but also damages such as separated aggregate and corrosion of steel structures. In addition, since it requires heavy computational costs to find small concrete cracks on massive civil structures, it is needed to investigate for a deep learning model that has a less parameters or a capability to process larger area at a time. Also, localization and quantification of multiple damages in higher accuracy needs to be performed in the future.

Author Contributions: B.K. and S.C. conceived and designed the methods; B.K. developed the software; B.K. performed the validation and visualization of results; B.K. and S.C. wrote the paper; S.C. supervised the research. S.C. acquired the research funding. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Research Foundation of Korea (NRF).

Acknowledgments: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 018R1C1B6006600).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. ASCE's 2017 Infrastructure Report Card | GPA: D+. Available online : https://www.infrastructurereportcard.org/tag/2020/. (accessed on 12 November 2020).
2. Bridge Inspector Dies after Cherry Picker Accident—CBS Pittsburgh. Available online : https://pittsburgh.cbslocal.com/2013/06/19/bridge-inspector-in-critical-condition-after-cherry-picker-accident/ (accessed on 12 November 2020).
3. Worker Killed in Bridge Inspection Accident—NBC Connecticut. Available online: https://www.nbcconnecticut.com/on-air/as-seen-on/worker-killed-in-accident-with-snooper-truck_hartford/51812/ (accessed on 12 November 2020).
4. Abdel-Qader, I.; Abudayyeh, O.; Kelly, M.E. Analysis of Edge-Detection Techniques for Crack Identification in Bridges. *J. Comput. Civ. Eng.* **2003**, *17*, 255–263. [CrossRef]
5. Lecompte, D.; Vantomme, J.; Sol, H. Crack Detection in a Concrete Beam using Two Different Camera Techniques. *Struct. Health Monit. Int. J.* **2006**, *5*, 59–68. [CrossRef]
6. Yamaguchi, T.; Nakamura, S.; Saegusa, R.; Hashimoto, S. Image-based crack detection for real concrete surfaces. *IEEJ Trans. Electr. Electron. Eng.* **2008**, *3*, 128–135. [CrossRef]
7. Rabah, M.; Elhattab, A.; Fayad, A. Automatic concrete cracks detection and mapping of terrestrial laser scan data. *NRIAG J. Astron. Geophys.* **2013**, *2*, 250–255. [CrossRef]
8. Torok, M.M.; Golparvar-Fard, M.; Kochersberger, K.B. Image-Based Automated 3D Crack Detection for Post-disaster Building Assessment. *J. Comput. Civ. Eng.* **2014**, *28*, A4014004. [CrossRef]
9. Prasanna, P.; Dana, K.J.; Gucunski, N.; Basily, B.B.; La, H.M.; Lim, R.S.; Parvardeh, H. Automated Crack Detection on Concrete Bridges. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 591–599. [CrossRef]
10. Kim, H.; Ahn, E.; Cho, S.; Shin, M.; Sim, S.H. Comparative analysis of image binarization methods for crack identification in concrete structures. *Cem. Concr. Res.* **2017**, *99*, 53–61. [CrossRef]
11. Cha, Y.J.; Choi, W.; Suh, G.; Mahmoudkhani, S.; Büyüköztürk, O. Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 731–747. [CrossRef]
12. Wang, N.; Zhao, Q.; Li, S.; Zhao, X.; Zhao, P. Damage Classification for Masonry Historic Structures Using Convolutional Neural Networks Based on Still Images. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 1073–1089. [CrossRef]
13. Wu, W.; Qurishee, M.A.; Owino, J.; Fomunung, I.; Onyango, M.; Atolagbe, B. Coupling Deep Learning and UAV for Infrastructure Condition Assessment Automation. In Proceedings of the 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA, 16–19 September 2018; pp. 1–7. [CrossRef]
14. Xue, Y.; Li, Y. A Fast Detection Method via Region-Based Fully Convolutional Neural Networks for Shield Tunnel Lining Defects. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 638–654. [CrossRef]
15. Li, S.; Zhao, X.; Zhou, G. Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. *Comput.-Aided Civ. Infrastruct. Eng.* **2019**, *34*, 616–634. [CrossRef]
16. Liu, Y.F.; Cho, S.; Fan, J.S. Concrete Crack Assessment Using Digital Image Processing and 3D Scene Reconstruction. *J. Comput. Civ. Eng.* **2016**, *30*, 1–19. [CrossRef]
17. Kim, B.; Cho, S. Image-based Concrete Crack Assessment using Mask and Region-based Convolutional Neural Network. *Struct. Control Health Monit.* **2019**, *26*, e2381. [CrossRef]
18. Ni, F.; Zhang, J.; Chen, Z. Zernike-moment measurement of thin-crack width in images enabled by dual-scale deep learning. *Comput.-Aided Civ. Infrastruct. Eng.* **2019**, *34*, 367–384. [CrossRef]
19. Kim, B.; Cho, S. Efflorescence assessment using hyperspectral imaging for concrete structures. *Smart Struct. Syst.* **2018**, *22*, 209. [CrossRef]
20. Kim, B.; Kim, D.; Cho, S. A Study on Concrete Efflorescence Assessment using Hyperspectral Camera. *J. Korean Soc. Saf.* **2017**, *32*, 98–103. [CrossRef]
21. Kramar, S.; Urosevic, M.; Pristacz, H.; Mirtič, B. Assessment of limestone deterioration due to salt formation by micro-Raman spectroscopy: Application to architectural heritage. *J. Raman Spectrosc.* **2010**, *41*, 1441–1448. [CrossRef]

22. González-Jorge, H.; Puente, I.; Riveiro, B.; Martínez-Sánchez, J.; Arias, P. Automatic segmentation of road overpasses and detection of mortar efflorescence using mobile LiDAR data. *Opt. Laser Technol.* **2013**, *54*, 353–361. [CrossRef]
23. Dawood, T.; Zhu, Z.; Zayed, T. Machine vision-based model for spalling detection and quantification in subway networks. *Autom. Constr.* **2017**, *81*, 149–160. [CrossRef]
24. German, S.; Brilakis, I.; DesRoches, R. Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety assessments. *Adv. Eng. Inform.* **2012**, *26*, 846–858. [CrossRef]
25. Tanaka, H.; Tottori, S.; Nihei, T. Detection of Concrete Spalling Using Active Infrared Thermography. *Q. Rep. RTRI* **2006**, *47*, 138–144. [CrossRef]
26. Wang, H.W.; Chen, C.H.; Cheng, D.Y.; Lin, C.H.; Lo, C.C. A Real-Time Pothole Detection Approach for Intelligent Transportation System. *Math. Probl. Eng.* **2015**, *2015*, 1–7. [CrossRef]
27. Mednis, A.; Strazdins, G.; Zviedris, R.; Kanonirs, G.; Selavo, L. Real time pothole detection using Android smartphones with accelerometers. In Proceedings of the 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), Barcelona, Spain, 27–29 June 2011; pp. 1–6. [CrossRef]
28. Lee, Y.I.; Kim, B.; Cho, S. Image-based spalling detection of concrete structures using deep learning. *J. Korea Concr. Inst.* **2018**, *30*, 91–99. [CrossRef]
29. Talab, A.M.A.; Huang, Z.; Xi, F.; HaiMing, L. Detection crack in image using Otsu method and multiple filtering in image processing techniques. *Optik* **2016**, *127*, 1030–1033. [CrossRef]
30. Yokoyama, S.; Matsumoto, T. Development of an Automatic Detector of Cracks in Concrete Using Machine Learning. *Procedia Eng.* **2017**, *171*, 1250–1255. [CrossRef]
31. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
32. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440. [CrossRef]
33. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988. [CrossRef]
34. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755. [CrossRef]
35. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995. [CrossRef]
36. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]
37. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
39. GitHub-waspinator/pycococreator: Helper functions to create COCO datasets Available online: https://github.com/waspinator/pycococreator (accessed on 12 November 2020). .
40. Griffiths, D.; Boehm, J. Rapid Object Detection Systems, Utilising Deep Learning and Unmanned Aerial Systems (UAS) for Civil Engineering Applications. *Int. Soc. Photogramm. Remote Sens. (ISPRS)* **2018**, *42*, 391–398. [CrossRef]
41. Oksuz, K.; Can Cam, B.; Akbas, E.; Kalkan, S. Localization Recall Precision (LRP): A New Performance Metric for Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 504–519.

42. Cai, Z.; Vasconcelos, N. Cascade R-CNN: High Quality Object Detection and Instance Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**. [CrossRef]