remote sensing

MDPI

Article

Concrete Bridge Defects Identification and Localization Based on Classification Deep Convolutional Neural Networks and Transfer Learning

Hajar Zoubir [1], Mustapha Rguig [1], Mohamed El Aroussi [1], Abdellah Chehri [2], Rachid Saadane [1] and Gwanggil Jeon [3,*]

[1] Laboratory of Systems Engeneering (LaGes), Hassania School of Public Works, Casablanca 20000, Morocco
[2] Department of Mathematics and Computer Science, Royal Military College of Canada, Kingston, ON K7K 7B4, Canada
[3] Department of Embedded Systems Engineering, Incheon National University, Incheon 22012, Korea
* Correspondence: gjeon@inu.ac.kr

Abstract: Conventional practices of bridge visual inspection present several limitations, including a tedious process of analyzing images manually to identify potential damages. Vision-based techniques, particularly Deep Convolutional Neural Networks, have been widely investigated to automatically identify, localize, and quantify defects in bridge images. However, massive datasets with different annotation levels are required to train these deep models. This paper presents a dataset of more than 6900 images featuring three common defects of concrete bridges (i.e., cracks, efflorescence, and spalling). To overcome the challenge of limited training samples, three Transfer Learning approaches in fine-tuning the state-of-the-art Visual Geometry Group network were studied and compared to classify the three defects. The best-proposed approach achieved a high testing accuracy (97.13%), combined with high F1-scores of 97.38%, 95.01%, and 97.35% for cracks, efflorescence, and spalling, respectively. Furthermore, the effectiveness of interpretable networks was explored in the context of weakly supervised semantic segmentation using image-level annotations. Two gradient-based backpropagation interpretation techniques were used to generate pixel-level heatmaps and localize defects in test images. Qualitative results showcase the potential use of interpretation maps to provide relevant information on defect localization in a weak supervision framework.

Keywords: concrete bridge; visual inspection; defect; deep convolutional neural network; transfer learning; interpretation techniques; weakly supervised semantic segmentation

check for updates

Citation: Zoubir, H.; Rguig, M.; El Aroussi, M.; Chehri, A.; Saadane, R.; Jeon, G. Concrete Bridge Defects Identification and Localization Based on Classification Deep Convolutional Neural Networks and Transfer Learning. *Remote Sens.* **2022**, *14*, 4882. https://doi.org/10.3390/rs14194882

Academic Editor: Lefei Zhang

Received: 31 May 2022
Accepted: 26 September 2022
Published: 30 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Bridges are key elements of a road network and play a critical role in the functional operation of the transportation system. During their service life, they are subjected to multiple deterioration mechanisms induced by material aging, variable loading, aggressive environmental actions, and extreme weather conditions. As a result, various types of damage (e.g., crack and corrosion [1,2]) occur over time and alter the structural behavior of bridges. Therefore, it is essential to accurately and timely detect and evaluate the damage to prevent failure and maintain structural safety and serviceability.

Structural Health Monitoring (SHM) has attracted much attention and has been the subject of several works in recent decades. Numerous techniques based on sensors (e.g., accelerometers, velocimeters, and displacement sensors) [3], non-destructive testing (e.g., ground-penetrating radar, infrared and ultrasonic techniques) [4], and visual inspection [5,6] have been deployed to identify, localize, and quantify damage in bridges. However, visual inspection has been the predominant practice for bridge condition assessment [7–9]. Trained inspectors conduct an in situ examination of bridge elements based on

established guidelines and evaluate the condition of the entire bridge. However, the conventional framework of this practice is time-consuming, labor-intensive, and error-prone due to the subjective judgment of inspectors. Moreover, it requires access equipment and vehicles to reach areas of the bridge with low accessibility, which incurs additional costs to the monitoring operation [8].

In recent years, technological advances in civil engineering and related disciplines have promoted the emergence of innovative tools to manage civil infrastructures. Within this context, bridge owners and managers have shown increasing interest in Unmanned Aerial Vehicles (UAVs) as an assistive, efficient, and cost-effective means offering great potential for inspection automation [8,10]. However, one of the major challenges associated with this inspection scheme lies in deploying an efficient method to process the large amount of image data collected by the UAVs' sensors. To this end, several vision-based techniques have been extensively explored to automate defect detection in different civil engineering structures. These methods include traditional Image Processing Techniques (IPTs) [11], Machine Learning algorithms [12], and Deep Convolutional Neural Networks (DCNNs) [13].

In the particular context of concrete damage detection, cracks are the primary type of damage investigated by researchers. IPTs are used to extract representative properties of cracks from input images by applying various filters and morphological operations (e.g., Edge Detectors [14], Thresholding [15], Percolation [16,17], and Principal Component Analysis [18]). Then, the extracted features are fed to Machine Learning models, such as Support Vector Machines [19,20] and Nearest Neighbor Classifiers [20], to perform the classification task. However, IPTs provide hand-crafted features for training [21] and present limited learning capabilities that do not represent the complexity of the concrete texture and the challenging conditions of image acquisition, such as lighting, shading, and camera movements [21,22].

On the other hand, DCNNs extract features from a set of training images through the convolution operation and classify them within one learning framework. Owing to their robust feature extraction and learning capabilities, DCNNs have been widely examined in concrete damage classification studies.

For example, Dorafshan et al. [23] demonstrated the superiority of the AlexNet network [24] over six standard edge detectors in classifying concrete crack images of the SDNET dataset [25].

Kim et al. [26] trained and optimized the LeNet-5 network [27] to detect cracks in concrete surfaces using a dataset of 40,000 images. The proposed model achieved an accuracy of 99.8% and could be implemented using low-power computational devices.

Yu et al. [28] developed a method based on DCNNs to detect cracks in image patches of damaged concrete. The authors proposed an architecture consisting of six convolutional layers, two pooling layers, and three fully connected layers and employed the enhanced chicken swarm algorithm to optimize the meta-parameters of the DCNN model.

Mundt et al. [29] proposed the CODEBRIM dataset that features five non-exclusive damage classes in bridges (i.e., crack, spallation, exposed reinforcement bar, efflorescence, and corrosion). In addition, they investigated reinforcement learning approaches to build a DCNN model for the multi-target classification task, and their best meta-learned models yielded a testing accuracy of 72%.

Since training DCNNs requires a significant amount of image data and due to the limited size of concrete damage datasets, researchers have explored Transfer Learning techniques to train deep learning networks for concrete damage classification [30–33]. Pretrained DCNNs (e.g., AlexNet [24], VGG [34], ResNet [35], Inception [36]) on large benchmark datasets (e.g., ImageNet [37], MNIST [27], CIFAR100 [38]) are used to transfer knowledge from a source domain (e.g., ImageNet dataset) to a target domain (e.g., a small-scale concrete damage dataset) through different settings and learning approaches [39].

Yang et al. [21] developed a low-cost automated inspection approach based on UAVs and deep learning. They constructed the CSSC database and used a fine-tuned VGG16

model to classify cracks and spalling in concrete bridge elements and achieved a mean accuracy of 93.36% with the CSSC dataset.

Hüthwohl et al. [40] used a pre-trained inception-V3 network to define a hierarchical multi-classifier for reinforced concrete bridge defects (i.e., cracks, efflorescence, spalling, exposed reinforcement, and rust staining). Experimental results showed that the multi-classifier could assign class labels with an average F1-score of 83.5%.

Yang et al. [33] proposed an end-to-end-based Transfer Learning method for crack detection using three knowledge transfer approaches (i.e., sample, model, and parameter transfer knowledge), a fine-tuned VGG16 model, and three crack datasets. Their experiments showed that by training 13 convolutional and two fully connected layers of the pre-trained VGG16 model on the three datasets, crack detection was improved and achieved a testing accuracy of 97.07% on the SDNET dataset.

Bukhsh et al. [31] investigated cross-domain and in-domain Transfer Learning approaches. They compared the performance of the VGG16, InceptionV3, and the ResNet50 models in different Transfer Learning strategies to detect damages in six binary and multi-label concrete damage datasets. Their experiments demonstrated that combined representations of in-domain and cross-domain Transfer provide considerable performance gain, particularly with tiny datasets.

Zhu et al. [41] built a robust classifier to detect four defects, including cracks, pockmarks, spalling, and exposed rebar. They used the pre-trained inceptionV3 model to extract features from input images and a fully connected network to classify defects. The proposed model was trained on 1180 images with arbitrary sizes and resolutions for 374.1 s and recorded a testing accuracy of 97.8%.

On the other hand, Gao and Mosalam [32] proposed the concept of structural ImageNet and manually labeled 2000 images for four recognition tasks: component type identification (binary), spalling condition check (binary), damage level evaluation (three classes), and damage type determination (four classes). They applied two different strategies of Transfer Learning based on the pre-trained VGG16 model. For the damage type multi-classification task, a 68.8% accuracy with 23% overfitting was obtained by retraining the last two convolutional blocks of the network.

In the aforementioned works, the performance of the proposed methods has varied according to the size and complexity of the datasets and the adopted Transfer Learning approach. Most studies have re-trained more than two or all convolutional layers and update a high number of the network parameters to achieve a higher detection accuracy. However, this approach is computationally expensive, requires more training time, and is also subject to overfitting in the context of heavily parameterized networks and small datasets.

In a bridge condition assessment framework, defect localization is crucial to evaluate damage's impact on the bridge's structural integrity. For this purpose, deep learning based-semantic segmentation algorithms have been deployed to provide pixel-level classification results to improve damage detection accuracy.

Zhang et al. [42] designed a fully convolutional model to detect and group image pixels for three types of concrete surface defects (i.e., crack, spalling, and exposed rebar). The authors prepared a dataset with mask labeling of 1443 images to train and test the model. Their proposed method achieved a semantic segmentation accuracy of 75.5%.

Fu et al. [43] introduced a crack detection method based on an improved DeepLabv3+ semantic segmentation algorithm. They established a concrete bridge crack segmentation dataset to train and test the proposed model. The experimental results proved the effectiveness of the trained algorithm that reached an average intersection over union ratio of 82.37%.

Wang et al. [44] constructed a crack dataset of 2446 manually labeled images to train and evaluate the performance of five deep networks for semantic segmentation. The best model achieved an F1-score of 77.32% and an intersection over union ratio of 62.98%. The

authors also discussed the influence of dataset choice and image noise on the detection performance.

Dung and Anh [45] developed a fully convolutional network-based method and annotated 600 crack-labeled images for semantic segmentation. The proposed model reached approximately 90% for the average precision score. The authors demonstrated their method's effectiveness by accurately identifying and capturing crack path and density variation in a crack opening video.

The above studies have shown very promising results in detecting damages. However, the fully supervised semantic segmentation deep networks are complex and are faced with a common major challenge associated with data scarcity. These models require training labeled images with pixel-level annotations that are expensive and necessitate the empirical knowledge of field experts. Furthermore, most publicly available concrete damage datasets only provide image-level annotations.

To alleviate the heavy workload associated with data annotation in a fully supervised learning framework, weakly supervised segmentation methods consider different weak annotations (e.g., image-level and bounding box labels) as the supervision condition [46]. Within the context of damage detection, Dong et al. [47] designed a patch-based weakly supervised semantic segmentation network to detect cracks in construction materials. In their proposed method, an input image is cropped, and the resulting patches are annotated at an image level. Class activation maps of cracks are obtained for each patch. They are fed to a fully connected conditional random field to generate the corresponding synthetic labels, which are used to train a segmentation network.

König et al. [48] presented a weakly supervised segmentation approach leveraging classification labels to detect surface cracks. To obtain pixel-level segmentation pseudo labels, the authors utilized a patch-threshold segmentation combined with coarse localization maps generated by a Convolutional Neural Network trained on images with classification annotations. The generated pseudo labels were used to train a standard semantic segmentation network to perform crack segmentation.

Zhu and Song [49] developed a weakly supervised network for crack segmentation in asphalt concrete bridge decks. Based on an autoencoder, the original data generates a weakly supervised start point for convergence, and image feature extraction and segmentation are performed under weak supervision.

This paper investigates a weakly supervised framework based on interpretation techniques and leveraging image-level annotations to generate pixel-level maps. The goal is to provide a coarse localization of three distinct types of damage in concrete bridge images.

The main contributions of this work are the following:

1. A multi-class labeled dataset with more than 6900 images was constructed. The dataset features three common types of defects in concrete bridges (i.e., cracks, spalling, and efflorescence) and covers their diverse representation in the real world of bridge inspection.
2. Three classification schemes using the pretrained Visual Geometry Group (VGG) network with its 16 learning layers (i.e., VGG16 [34]), Transfer Learning, and the proposed dataset were compared. The experiments investigated the effect of the number of layers to be retrained on the model's performance in terms of classification measures (i.e., accuracy, precision, recall, and F1 score), computational time, and generalization ability.
3. Based on the best classification scheme, the effectiveness of interpretable neural networks was explored in the context of weakly supervised semantic segmentation (i.e., image-level supervision). Two gradient-based backpropagation interpretation techniques (i.e., Gradient-weighted Class Activation Mapping (Grad-CAM) [50] and Grad-CAM++ [51]) were used to generate pixel-level heatmaps and localize defects. Qualitative results of test images showcase the potential of interpretation heatmaps to provide localization information in a weak supervision framework.

The rest of the paper is organized as follows: Section 2 presents an overview of the methodology followed in this paper, the proposed dataset, the VGG16 model, and the interpretation techniques studied in this work. Section 3 details the experimental setup, and the experimental results are presented and discussed in Section 4. Conclusions are provided in the final section of the paper.

2. Methodology and Materials

An overview of the adopted methodology in this work is shown in the flowchart in Figure 1. The first module corresponds to dataset construction's image acquisition and preparation process. The second module represents the implementation of the pre-trained VGG16 network using Transfer Learning to classify three types of defects in concrete bridges. Finally, interpretation techniques were deployed in the third module to generate pixel-level heatmaps to localize concrete damage.

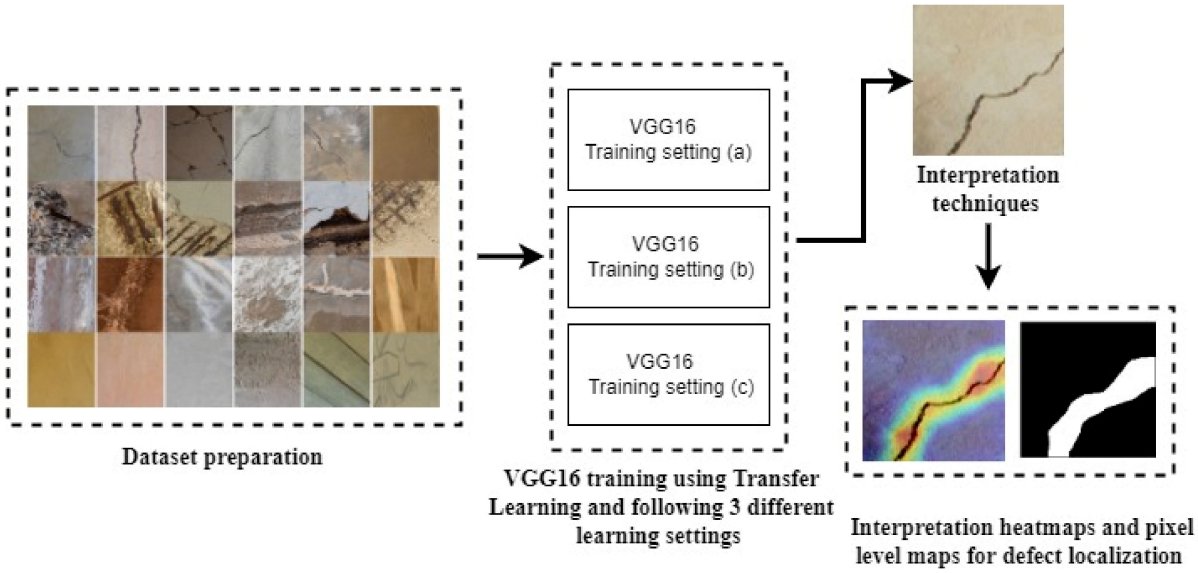

Figure 1. Overview of the proposed method.

2.1. Dataset

The dataset constructed in this paper consisted of 6952 RGB images with a 200 × 200 px resolution of concrete cracks (1304), concrete spalling (1100), concrete efflorescence (1029), and non-defective background (3519). Cracks and background images were extracted from the dataset established by the authors of [52].

More than 1200 images of Moroccan bridges representing decks and piers with concrete spalling and efflorescence were collected and processed according to the same experimental setup and procedure in [52]. The images were captured using two 20-MP consumer digital cameras with 5 mm of focal length, a sensitivity of 100 ISO, and a maximum resolution of 5152 × 3864. They were gathered at varying distances from bridges and a maximum 8× optical zoom was applied. Moreover, the images were taken under different weather and lighting conditions, and a flash was used to illuminate the dark bridge areas containing defects. It is noteworthy that the original images have not undergone any processing operations other than the manual cropping using the inbac tool [53].

The dataset in [52] was expanded with the concrete spalling and efflorescence classes, and the resulting dataset is publicly available at [54] for academic purposes.

Various colors, textures, surface conditions of concrete and defect representations were included in the constructed dataset to cover the variation of defect appearance, extent, and severity level in the real world of bridge inspection. Figure 2 presents sample images of the proposed dataset.

Figure 2. Sample images of the constructed dataset (row 1: concrete cracks, row 2: concrete spalling with exposed reinforcement, row 3: concrete efflorescence, row 4: different representations of the non-defective background class).

2.2. VGG16 and Transfer Learning

The Visual Geometry Group introduced VGG16 in 2014 [34]. The algorithm is very efficient and won first place in object localization and second place in image classification in the ImageNet Large Scale Visual Recognition Challenge. This model trained on the ImageNet dataset achieved a top-1 accuracy of 71.5% and a top-5 accuracy of 90.1% in image classification.

The network contains 13 convolutional layers with 3 × 3 filters (i.e., convolution kernels) to extract features. In addition, the network contains five max-pooling layers to reduce the number of learnable parameters (i.e., weights and biases) and three fully connected layers to map the flattened features to the Softmax layer where target class probabilities are calculated. In addition, the Rectified Linear Unit (ReLU) activation function is used to increase the non-linearity of the model. The network takes 224 × 224 RGB images as inputs and has more than 138 million learning parameters. Figure 3 presents the architecture of the VGG16 model.

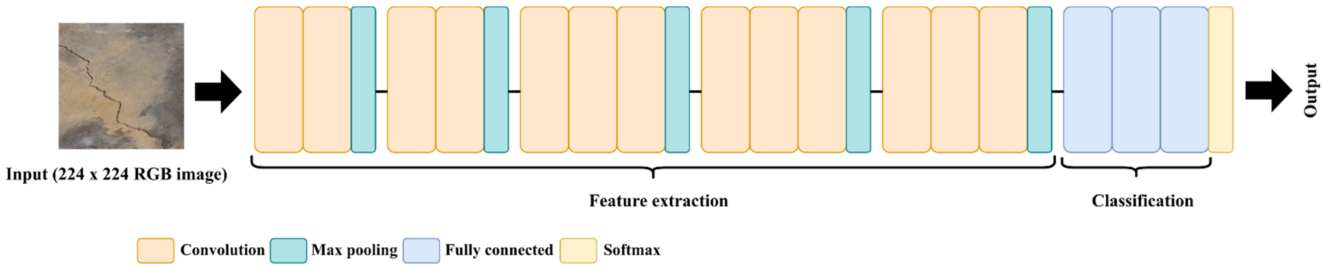

Figure 3. Architecture of the VGG16 model.

The learning layout of the VGG16 network, and DCNNs in general, is based on optimizing a loss function (e.g., Binary and Multi-Class Cross-Entropy loss) that measures the discrepancy between the predicted outputs and ground truth through back-propagation.

The optimization scheme generally uses gradient descent optimizers (e.g., Stochastic Gradient Descent and adaptive optimizers) to update the learning parameters of the network.

For image classification, VGG16 and other state-of-the-art DCNNs are usually trained on the ImageNet dataset that contains millions of images belonging to thousands of classes. However, since the size of domain-specific datasets (e.g., concrete defects datasets in the case of this study) is limited, Transfer Learning techniques are applied to overcome the scarcity of labeled data.

In a Transfer Learning approach, pre-trained models on large datasets (e.g., ImageNet) are fine-tuned and partially retrained on the small target dataset. In this learning framework, the weights of the lower-level layers are generally maintained since they represent generic features. In contrast, the high-level layers are more sensitive to the target dataset and must be retrained to update their learning parameters [23]. The Transfer Learning settings examined in this paper are detailed in the experimental setup section.

2.3. Interpretation Techniques

In the context of image classification, interpretation techniques are intended to explain the predictions of trained models by visualizing the regions of the inputs that contributed to the final classification result. Thus, they can provide a coarse localization of target objects using image-level annotations.

Hereafter, a simplified explanation of the intuition behind the two gradient-based back-propagation techniques used in this paper (i.e., Grad-CAM and Grad-CAM++) is presented.

2.3.1. Gradient-Weighted Class Activation Mapping (Grad-CAM)

The Grad-CAM approach is based on the gradient information for the last convolutional layer of a trained network [50].

The gradients of the score for class c (y^c) with respect to the feature maps A^k of the convolutional layer are computed via back-propagation and then global-average pooled to obtain the weights w_c^k [50]:

$$w_c^k = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{1}$$

where, Z is the number of pixels in the activation map.

The weight w_c^k expresses the importance of feature map k for the class c. The class discriminative localization map Grad-CAM $L_{Grad\text{-}CAM}^c$ is obtained by computing a weighted sum of the forward feature maps A^k of the last convolutional layer [50]:

$$L_{Grad\text{-}CAM}^c = ReLU\left(\sum_k w_k^c A^k\right) \tag{2}$$

where *ReLU* is the Rectified Linear Unit activation function. It is used to focus only on the features that have a positive influence on the target class [50].

2.3.2. Grad-CAM++

Grad-CAM++ is a generalization to Grad-CAM and provides better visualizations of the network decisions [48]. In Grad-CAM++, the weights w_c^k are computed as follows [51]:

$$w_c^k = \sum_i \sum_j \alpha_{ij}^{kc} . ReLU\left(\frac{\partial y^c}{\partial A_{ij}^k}\right) \tag{3}$$

where α_{ij}^{kc} are weighting coefficients for the pixel-wise gradients for class c and feature map A^k and are defined as follows:

$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 y^c}{\left(\partial A_{ij}^k\right)^2}}{2\frac{\partial^2 y^c}{\left(\partial A_{ij}^k\right)^2} + \sum_a \sum_b A_{ab}^k \left\{ \frac{\partial^3 y^c}{\left(\partial A_{ij}^k\right)^3} \right\}} \tag{4}$$

3. Experimental Setup

This work aims to define an efficient and automatable method to identify and localize damage in concrete bridge images using DCNNs and Transfer Learning. This section presents the Transfer Learning schemes followed to train the pre-trained VGG16 model on the proposed dataset. In addition, the weakly supervised semantic segmentation framework based on the above-explained interpretation techniques is also discussed.

3.1. VGG16 Fine-Tuning and Training

The VGG16 model can capture high-level features [21] and has the ability to generalize to other datasets [32]. Moreover, it has shown an excellent performance in many studies on damage classification in concrete surfaces [31–33]. Therefore, it was chosen as a base model for the learning approach proposed in this paper.

Training this deep network from scratch requires enormous computational resources, significantly labeled data, and excessive training time. Thus, three Transfer Learning settings were explored in this work and compared based on standard classification metrics, training time, and generalization ability.

First, the pre-trained VGG16 with the ImageNet weights was uploaded, and the last fully connected layers of the model were adjusted to the number of classes (i.e., four classes). Then, based on the assumption that the high-level layers of DCNNs are more sensitive to the target dataset, the last layers of the pre-trained model were retrained on the constructed dataset to update their learning parameters.

Gradient descent and back-propagation were used following three different approaches in this work:

- Retraining the classification layers (a)
- Retraining the classification layers and the last convolutional layer (b)
- Retraining the classification layers and the last two convolutional layers (c)

Figure 4 presents the three Transfer Learning-based training settings investigated in this paper.

The dataset presented in Section 2 was randomly split into three subsets: 70% of the images were used in the training set, 10% in the validation set, and 20% in the testing set. The number of images per subset and per class is shown in Table 1.

Table 1. Number of images per subset per class.

	Background	Cracks	Efflorescence	Spalling
Training set	2463	912	720	770
Validation set	351	130	102	110
Testing set	705	262	207	220

In addition, data augmentation techniques (i.e., random horizontal and vertical flips and random rotations) were applied to the training set to avoid overfitting.

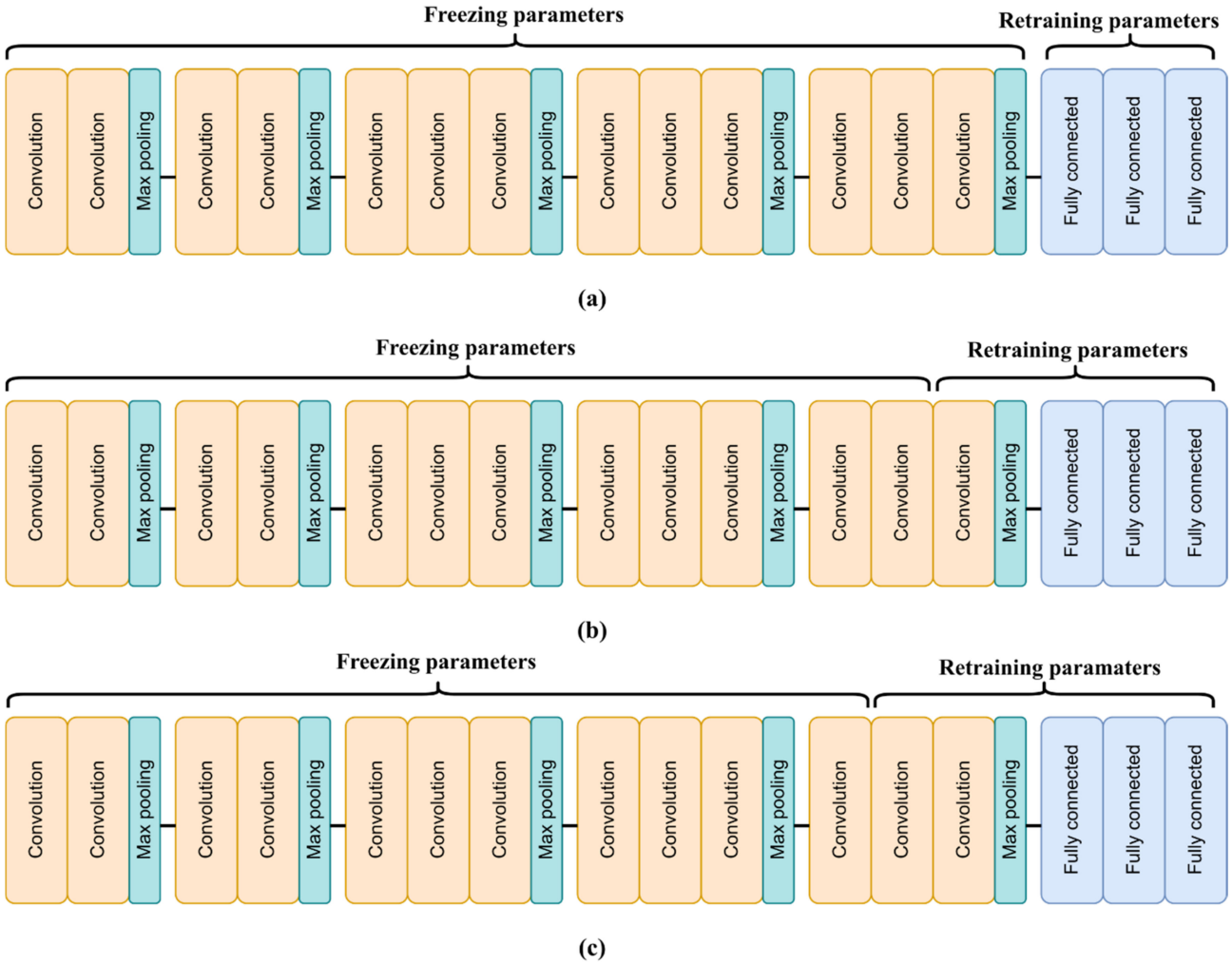

Figure 4. Transfer Learning configurations: (**a**) retraining only the classification layers, (**b**) retraining the classification layers and the last convolutional layers, (**c**) retraining the classification layers and the last two convolutional layers.

The optimization method recommended in [32] based on Stochastic Gradient Descent (SGD) with momentum and a small learning rate was used in this work's experiments. The SGD with momentum optimizer reduces the computational load and accelerates the training convergence. The training was conducted for 25 epochs as the results converged. In addition, low training and validation errors were achieved while mitigating overfitting. The cross-entropy loss function was optimized using the SGD with a learning rate of 0.001, a momentum of 0.9, and a mini-batch size of 32. All the experiments were carried out using Pytorch in Google Colaboratory (Colab) with the 12GB NVIDIA Tesla K80 GPU provided by the platform.

3.2. Evaluation Metrics

In each learning configuration, the performance of the model was evaluated using the following metrics:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

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

$$\mathrm{F1}_{\mathrm{Score}} = 2\left(\frac{1}{\mathrm{Recall}} + \frac{1}{\mathrm{Precision}}\right)^{-1} \quad (8)$$

TP (True Positives) refer to the number of correctly classified images as defects.

TN (True Negatives) refer to the number of background images that are correctly classified as background.

FP (False Positives) refer to the number of background images that are incorrectly identified with defects.

FN (False Negatives) refer to the number of images incorrectly identified as background images.

The Root Mean Squared Error (RMSE) was also used to assess the model's performance in the three different training schemes. It is defined by Equation (9):

$$RMSE = \sqrt{\sum_i (1 - y_i)^2 / n} \quad (9)$$

where y_i is the calculated probability of the image i (from the testing subset) belonging to the ground truth class.

3.3. Weakly Supervised Semantic Segmentation

Based on the best learning scheme, feature maps of the last convolutional layer of the trained model were used to provide visual explanations of classification results using Grad-CAM and Grad-CAM++. The implementation of these interpretation techniques was based on the publicly available repository in [55]. Pixel-level heatmaps were generated for test images, and a threshold of 0.5 was applied to each image to localize the regions with a target class probability above 50%.

4. Results and Discussions

This section presents and discusses the results obtained after training and testing the model following the three Transfer Learning schemes on the constructed dataset. In addition, some representative localization maps of cracks, spalling, and efflorescence generated by the two interpretation techniques are also presented.

4.1. Training and Testing Results

Figure 5 plots the model's learning and loss curves following the three Transfer Learning settings. It can be seen that in all three training approaches, the model converged quickly from the early epochs; this is mainly attributed to the reduced number of layers to retrain and the fixed parameters of the non-trainable layers.

The training and validation accuracies in the three learning settings generally increase over time, reaching a plateau around the last three epochs. The loss curves in (b) and (c) show a slight tendency to over-fitting due to the increasing number of parameters to update in their corresponding learning schemes. Therefore, it is believed that training more than two convolutional layers will lead to a higher possibility of overfitting and, consequently, will decrease the model's generalization ability.

Table 2 lists the three learning schemes' best training, validation, testing accuracies, and RMSE. The results show that the model in scheme (c) presents a better performance in training compared to settings (a) and (b). The achieved training accuracy was 98.34% in (c) over 94.62% and 91.10% in (b) and (a), respectively. The model in setting (c) also yielded a higher testing accuracy of 97.13% over 94.61% and 91.10% in (b) and (a), respectively. Furthermore, the results were obtained with only 1.21% overfitting (calculated as the difference between training and testing accuracies). These exciting results show that approach (c) enables the model to have better generalizability to extend the learning from the training subset to unseen test data. These results also demonstrate that more important features representing the target dataset were learned in the last two convolutional layers of the pre-trained model.

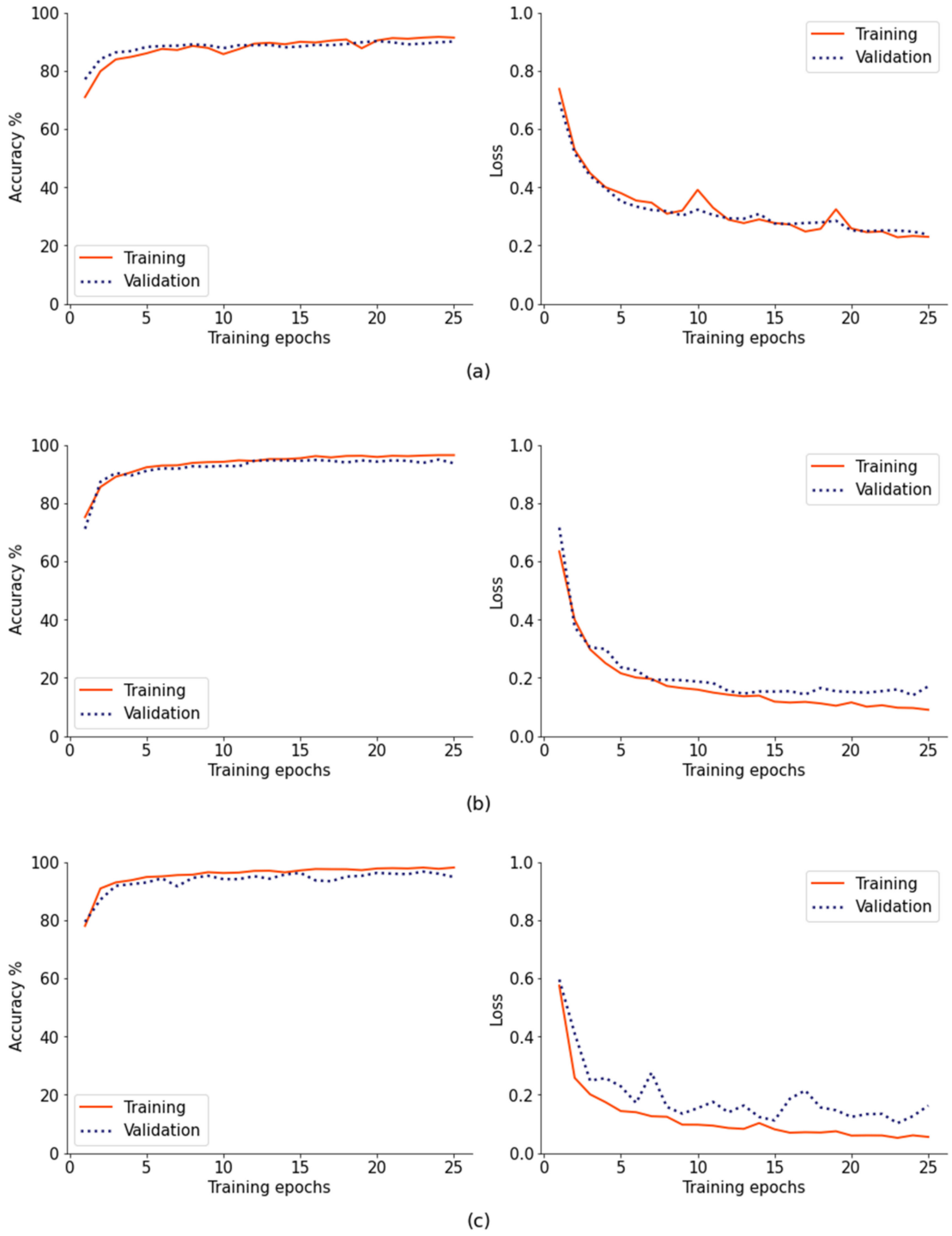

Figure 5. Learning and loss curves: (**a**) retraining only the classification layers, (**b**) retraining the classification layers and the last convolutional layers, (**c**) retraining the classification layers and the last two convolutional layers.

In addition, the Root Mean Squared Error decreased with more retrained layers (0.15 in (c) over 0.20 and 0.27 in (b) and (a), respectively), denoting that the predictive capacity of the model improves with updating more learning parameters.

It is also essential to mention that the training time corresponding to the three classification methods is reasonable. Moreover, the difference between the three approaches in terms of training time is significantly low (e.g., 8 s between (b) and (c)). At the same time, a considerable gain in performance was observed (e.g., a 1.61% gain in training accuracy

between (b) and (c)). Therefore, fine-tuning the classification layers and the last two convolution layers of the pre-trained VGG16 is efficient as it balances prediction performance, training time, and overfitting.

Table 2. Best training, validation, and testing accuracies and training times of the three training settings.

Transfer Learning Scheme	Best Training Accuracy	Best Validation Accuracy	Testing Accuracy	RMSE	Training Time
Retraining only the classification layers (a)	91.61%	90.62%	91.10%	0.27	6 min 35 s
Retraining the classification layers and the last convolutional layers (b)	96.73%	94.80%	94.62%	0.20	6 min 47 s
Retraining the classification layers and the last two convolutional layers (c)	98.34%	96.68%	97.13%	0.15	6 min 55 s

To further visualize the performance of the trained DCNN on the test subset, normalized confusion matrices for the three learning schemes are presented in Figure 6.

The results reflect confusion between background and efflorescence images and background and crack pictures. This confusion was particularly observed in schemes (a) and (b). For example, 7% and 6% of background images were classified as cracks or efflorescence in methods (a) and (b), respectively. However, this confusion was notably reduced in the scheme (c) as less than 3% of background images were predicted as cracks or efflorescence.

The observed confusion is mainly related to the complexity of the concrete surface in terms of colors and textures. In addition, some surface alterations in the training dataset (e.g., stains, markings, minor defects such as scaling and segregation) represent noisy defect-like features in concrete images and, as a result, make feature learning more challenging. For example, some background images contain concrete joints representing straight lines in the concrete surface and hence are likely to be misclassified as cracks. Generally, this confusion between classes can be further handled by adding more labeled samples to the training dataset and integrating an additional denoising process into image data.

Figure 7 illustrates some misclassification examples corresponding to the learning setting used in (c).

The precision, recall, and F1-scores of each defect class were computed using the confusion matrices. The results are summarized in Table 3.

The model achieved higher precision, recall, and F1-score results in learning scheme (c) compared to the other learning settings. For example, 86.64%, 94.03%, and 97.38% are the cracks F1-scores achieved in the learning settings (a), (b), and (c), respectively.

By comparing the F1-scores of the three classes in the learning scheme (c), the efflorescence class yielded a lower score than the other defects, which showed nearly similar performance (95.01% for efflorescence over 97.38% and 97.35% for cracks and spalling, respectively). This can be attributed to the wide representations of the efflorescence in concrete and the complexity of features required to describe this defect class. This issue can be solved by adding more training data that extensively covers the diverse representations of this type of damage.

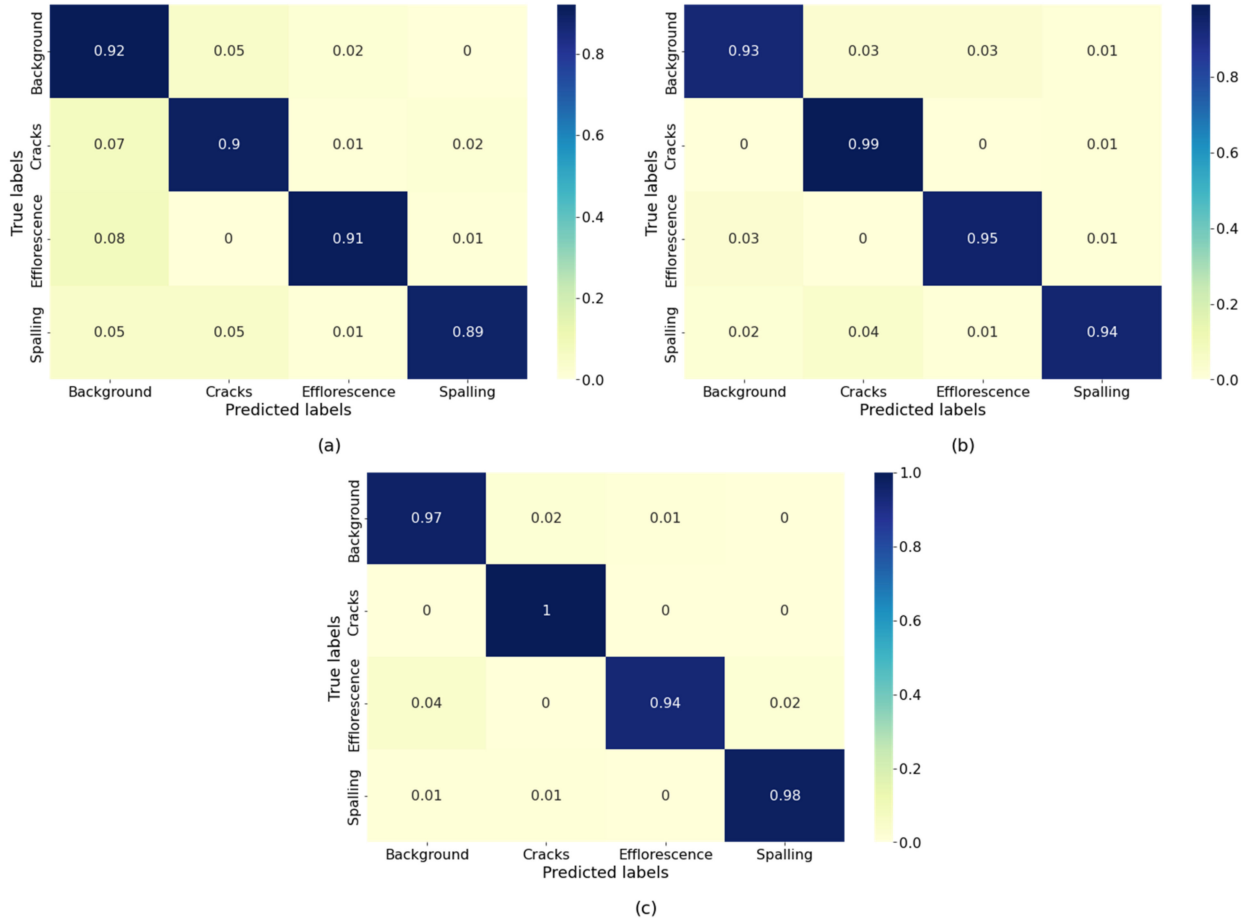

Figure 6. Confusion matrices of the three learning settings: (**a**) retraining only the classification layers, (**b**) retraining the classification layers and the last convolutional layers, (**c**) retraining the classification layers and the last two convolutional layers.

Figure 7. Misclassification examples (row 1: background images containing concrete joints misclassified as cracks, row 2: background images with surface alteration and different concrete colors misclassified as efflorescence).

Table 3. Precision, recall, and F1-scores of the three models.

	Damage Type	Learning Scheme (a)	Learning Scheme (b)	Learning Scheme (c)
Precision	Cracks	83.04%	89.66%	94.89%
	Efflorescence	89.62%	89.29%	95.69%
	Spalling	96.92%	95.89%	96.92%
Recall	Cracks	90.57%	98.86%	100%
	Efflorescence	90.91%	95.24%	94.34%
	Spalling	88.89%	93.75%	97.78%
F1-Score	Cracks	86.64%	94.03%	97.38%
	Efflorescence	90.26%	92.17%	95.01%
	Spalling	92.59%	94.81%	97.35%

4.2. Defect Localization Results

The trained model using learning scheme (c) was employed to implement the interpretation techniques presented in Section 2.

Images of the testing subset were used to visualize the implementation results. Figure 8 shows sample examples of the obtained results.

As intended, the resulting heatmaps highlight the discriminative image regions that contributed to image classification. These heatmaps show the probability of the target class at each pixel. By analyzing the qualitative results in Figure 8, the active regions are primarily consistent with the defect area. Grad-CAM++ provided better visualization results for cracks and efflorescence examples compared to Grad-CAM.

The pixel-level maps generated after applying a threshold of 0.5 provide a coarse localization of the concrete defects and offer semantically meaningful discrimination at the pixel level between defects and background. Therefore, it is believed that in the context of weakly supervised semantic segmentation, interpretation methods can provide relevant pixel-level maps using only image annotations as the supervision condition. The proposed method has reasonably captured a coarse localization of defects while avoiding the annotation workload of the fully supervised semantic segmentation-based frameworks.

However, since the visualization results using these interpretation techniques depend on the feature space learned by the classifier, some highlighted areas do not represent the target classes in the test images, and other regions representing damage were not captured.

As a result, it would be challenging to localize and quantify the damage precisely (e.g., crack path and density). This can be attributed to the underlying complexity of the training dataset, its limited size, and the limited learning capabilities of the pre-trained network due to the difference between the source domain (ImageNet dataset) and the target domain (the proposed concrete damage dataset). Thus, to further examine the potential of interpretation techniques in weakly supervised semantic segmentation, more customized networks tailored to the damage classification task and trained on more comprehensive datasets should be explored.

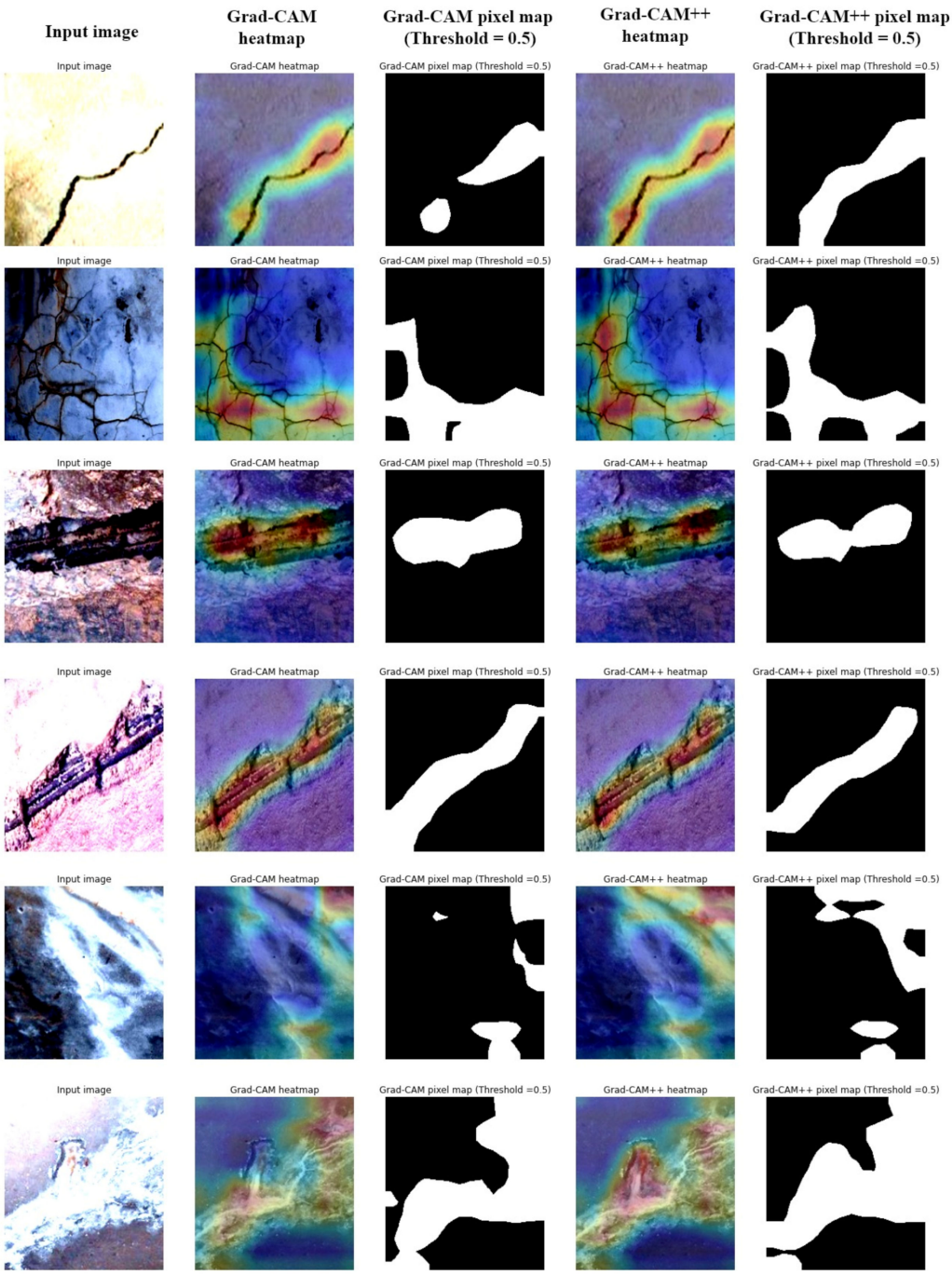

Figure 8. Sample results of the interpretation techniques implementation (rows 1–2: cracks, rows 3–4: concrete spalling, rows 5–6: efflorescence).

5. Conclusions and Perspectives

Structural Health Monitoring (SHM) is gaining increasing importance in assessing bridge conditions as it allows for identifying, localizing, and evaluating damage severity. Therefore, SHM is part of an economic strategy since it intervenes in the definition of maintenance actions and participates in the optimization of its allocated resources.

This paper presented a benchmark image dataset featuring three common defects (i.e., cracks, efflorescence, and spalling) of concrete bridges. The dataset covers different appearances of the three defects and the concrete surface in the real world of bridge inspection. A VGG16 network was trained on the proposed dataset following three Transfer Learning schemes with varying layers. In each learning configuration, the performance

of the model was evaluated based on classification metrics, computational time, and generalization ability. Experiments showed a significant gain in classification measures when retraining the classification layers and the last two convolutional layers of the VGG16 network. The trained model yielded a high testing accuracy of 97.13%, combined with high F1-scores of 97.38%, 95.01%, and 97.35% for cracks, efflorescence, and spalling, respectively. In addition, a slight tendency to overfitting was observed in the corresponding learning scheme, which means that increasing the number of layers to be retrained will lead to the degradation of the model's generalization performance. These experimental results show the robustness of the proposed learning setting as it ensures a balance between classification metrics, computational time, and generalization ability.

This work also explored the potential of interpretation techniques to localize the three defects in the context of weakly supervised semantic segmentation. To this end, two gradient-based back-propagation methods were used to generate pixel-level heatmaps of test images leveraging the above-discussed learning setting. The resulting maps highlight the regions contributing to the classification result and then provide relevant pixel-level maps to localize defects using a model trained on image-level annotations. However, since these techniques rely on the feature space learned by the model, their results are limited by the representativity of target classes in the training dataset, the challenging complexity of the concrete surface texture and condition in bridge inspection images, and the learning capability of the model.

Therefore, in another attempt to solve the damage localization task, more advanced object detection models and datasets with different annotation levels will be investigated in future works.

Author Contributions: Conceptualization, H.Z. and M.R.; methodology, H.Z., M.R. and M.E.A.; software, H.Z.; validation, H.Z., M.R., M.E.A., A.C. and R.S.; formal analysis, H.Z., M.R. and M.E.A.; investigation, H.Z. and M.R.; resources, H.Z.; data curation, H.Z.; writing—original draft preparation, H.Z., M.R., M.E.A., A.C. and R.S.; writing—review and editing, A.C. and G.J.; visualization, H.Z., M.R., M.E.A., A.C. and R.S.; supervision, M.R.; project administration, M.R.; funding acquisition, A.C. and G.J. All authors have read and agreed to the published version of the manuscript.

Funding: The financial support from the NSERC Discovery Grant program.

Data Availability Statement: Not applicable.

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

References

1. Ma, Y.; Guo, Z.; Wang, L.; Zhang, J. Probabilistic Life Prediction for Reinforced Concrete Structures Subjected to Seasonal Corrosion-Fatigue Damage. *J. Struct. Eng.* **2020**, *146*, 04020117. [CrossRef]
2. Wang, L.; Dai, L.; Bian, H.; Ma, Y.; Zhang, J. Concrete cracking prediction under combined prestress and strand corrosion. *Struct. Infrastruct. Eng.* **2019**, *15*, 285–295. [CrossRef]
3. Pourzeynali, S.; Zhu, X.; Ghari Zadeh, A.; Rashidi, M.; Samali, B. Comprehensive Study of Moving Load Identification on Bridge Structures Using the Explicit Form of Newmark-β Method: Numerical and Experimental Studies. *Remote Sens.* **2021**, *13*, 2291. [CrossRef]
4. Kot, P.; Muradov, M.; Gkantou, M.; Kamaris, G.S.; Hashim, K.; Yeboah, D. Recent Advancements in Non-Destructive Testing Techniques for Structural Health Monitoring. *Appl. Sci.* **2021**, *11*, 2750. [CrossRef]
5. Zollini, S.; Alicandro, M.; Dominici, D.; Quaresima, R.; Giallonardo, M. UAV Photogrammetry for Concrete Bridge Inspection Using Object-Based Image Analysis (OBIA). *Remote Sens.* **2020**, *12*, 3180. [CrossRef]
6. Galdelli, A.; D'Imperio, M.; Marchello, G.; Mancini, A.; Scaccia, M.; Sasso, M.; Frontoni, E.; Cannella, F. A Novel Remote Visual Inspection System for Bridge Predictive Maintenance. *Remote Sens.* **2022**, *14*, 2248. [CrossRef]
7. Omar, T.; Nehdi, M. Condition Assessment of Reinforced Concrete Bridges: Current Practice and Research Challenges. *Infrastructures* **2018**, *3*, 36. [CrossRef]
8. Dorafshan, S.; Maguire, M. Bridge inspection: Human performance, unmanned aerial systems and automation. *J. Civ. Struct. Health Monit.* **2018**, *8*, 443–476. [CrossRef]
9. Alsharqawi, M.; Zayed, T.; Abu Dabous, S. Integrated condition rating and forecasting method for bridge decks using Visual Inspection and Ground Penetrating Radar. *Autom. Constr.* **2018**, *89*, 135–145. [CrossRef]
10. Seo, J. Drone-enabled bridge inspection methodology and application. *Autom. Constr.* **2018**, *94*, 112–126. [CrossRef]

11. Mohan, A.; Poobal, S. Crack detection using image processing: A critical review and analysis. *Alex. Eng. J.* **2018**, *57*, 787–798. [CrossRef]
12. Hsieh, Y.-A.; Tsai, Y.J. Machine Learning for Crack Detection: Review and Model Performance Comparison. *J. Comput. Civ. Eng.* **2020**, *34*, 04020038. [CrossRef]
13. Sony, S.; Dunphy, K.; Sadhu, A.; Capretz, M. A systematic review of convolutional neural network-based structural condition assessment techniques. *Eng. Struct.* **2021**, *226*, 111347. [CrossRef]
14. 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]
15. 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]
16. Fast Crack Detection Method for Large-Size Concrete Surface Images Using Percolation-Based Image Processing | SpringerLink. Available online: https://link.springer.com/article/10.1007/s00138-009-0189-8 (accessed on 4 March 2022).
17. 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]
18. Abdel-Qader, I.; Pashaie-Rad, S.; Abudayyeh, O.; Yehia, S. PCA-Based algorithm for unsupervised bridge crack detection. *Adv. Eng. Softw.* **2006**, *37*, 771–778. [CrossRef]
19. 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. Automat. Sci. Eng.* **2016**, *13*, 591–599. [CrossRef]
20. Jahanshahi, M.R.; Masri, S.F. A Novel Crack Detection Approach for Condition Assessment of Structures. In Proceedings of the International Workshop on Computing in Civil Engineering 2011, Miami, FL, USA, 19–22 June 2011; pp. 388–395. [CrossRef]
21. Yang, L.; Li, B.; Li, W.; Liu, Z.; Yang, G.; Xiao, J. Deep Concrete Inspection Using Unmanned Aerial Vehicle Towards CSSC Database. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, BC, Canada, 24–28 September 2017; p. 9.
22. da Silva, W.R.L.; de Lucena, D.S. Concrete Cracks Detection Based on Deep Learning Image Classification. *Proceedings* **2018**, *2*, 489. [CrossRef]
23. Dorafshan, S.; Thomas, R.J.; Maguire, M. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Constr. Build. Mater.* **2018**, *186*, 1031–1045. [CrossRef]
24. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
25. Dorafshan, S.; Thomas, R.J.; Maguire, M. SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data Brief* **2018**, *21*, 1664–1668. [CrossRef] [PubMed]
26. Kim, B.; Yuvaraj, N.; Sri Preethaa, K.R.; Arun Pandian, R. Surface crack detection using deep learning with shallow CNN architecture for enhanced computation. *Neural Comput. Appl.* **2021**, *33*, 9289–9305. [CrossRef]
27. Lecun, Y. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 47. [CrossRef]
28. Yu, Y.; Rashidi, M.; Samali, B.; Mohammadi, M.; Nguyen, T.N.; Zhou, X. Crack detection of concrete structures using deep convolutional neural networks optimized by enhanced chicken swarm algorithm. *Struct. Health Monit.* **2022**, *21*, 14759217211053546. [CrossRef]
29. Mundt, M.; Majumder, S.; Murali, S.; Panetsos, P.; Ramesh, V. Meta-Learning Convolutional Neural Architectures for Multi-Target Concrete Defect Classification with the Concrete Defect Bridge Image Dataset. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 11188–11197. [CrossRef]
30. Su, C.; Wang, W. Concrete Cracks Detection Using Convolutional Neural Network Based on Transfer Learning. *Math. Probl. Eng.* **2020**, *2020*, 7240129. [CrossRef]
31. Bukhsh, Z.A.; Jansen, N.; Saeed, A. Damage detection using in-domain and cross-domain transfer learning. *Neural Comput. Appl.* **2021**, *33*, 16921–16936. [CrossRef]
32. Gao, Y.; Mosalam, K.M. Deep Transfer Learning for Image-Based Structural Damage Recognition: Deep transfer learning for image-based structural damage recognition. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 748–768. [CrossRef]
33. Yang, Q.; Shi, W.; Chen, J.; Lin, W. Deep convolution neural network-based transfer learning method for civil infrastructure crack detection. *Autom. Constr.* **2020**, *116*, 103199. [CrossRef]
34. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556. Available online: http://arxiv.org/abs/1409.1556 (accessed on 16 April 2021).
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
36. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. *arXiv* **2015**, arXiv:1512.00567.
37. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Li, F.-F. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
38. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. 2009. Available online: https://www.cs.toronto.edu/~{}kriz/learning-features-2009-TR.pdf (accessed on 30 May 2022).
39. Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A Comprehensive Survey on Transfer Learning. *Proc. IEEE* **2021**, *109*, 43–76. [CrossRef]

40. Hüthwohl, P.; Lu, R.; Brilakis, I. Multi-classifier for reinforced concrete bridge defects. *Autom. Constr.* **2019**, *105*, 102824. [CrossRef]
41. Zhu, J.; Zhang, C.; Qi, H.; Lu, Z. Vision-based defects detection for bridges using transfer learning and convolutional neural networks. *Struct. Infrastruct. Eng.* **2020**, *16*, 1037–1049. [CrossRef]
42. Zhang, C.; Chang, C.; Jamshidi, M. Simultaneous pixel-level concrete defect detection and grouping using a fully convolutional model. *Struct. Health Monit.* **2021**, *20*, 2199–2215. [CrossRef]
43. Fu, H.; Meng, D.; Li, W.; Wang, Y. Bridge Crack Semantic Segmentation Based on Improved Deeplabv3+. *J. Mar. Sci. Eng.* **2021**, *9*, 671. [CrossRef]
44. Wang, J.-J.; Liu, Y.-F.; Nie, X.; Mo, Y.L. Deep convolutional neural networks for semantic segmentation of cracks. *Struct. Control Health Monit.* **2022**, *29*, e2850. [CrossRef]
45. Dung, C.V.; Anh, L.D. Autonomous concrete crack detection using deep fully convolutional neural network. *Autom. Constr.* **2019**, *99*, 52–58. [CrossRef]
46. Zhang, M.; Zhou, Y.; Zhao, J.; Man, Y.; Liu, B.; Yao, R. A survey of semi- and weakly supervised semantic segmentation of images. *Artif. Intell. Rev.* **2020**, *53*, 4259–4288. [CrossRef]
47. Dong, Z.; Wang, J.; Cui, B.; Wang, D.; Wang, X. Patch-based weakly supervised semantic segmentation network for crack detection. *Constr. Build. Mater.* **2020**, *258*, 120291. [CrossRef]
48. König, J.; Jenkins, M.; Mannion, M.; Barrie, P.; Morison, G. Weakly-Supervised Surface Crack Segmentation by Generating Pseudo-Labels using Localization with a Classifier and Thresholding. *arXiv* **2021**, arXiv:2109.00456. [CrossRef]
49. Zhu, J.; Song, J. Weakly supervised network based intelligent identification of cracks in asphalt concrete bridge deck. *Alex. Eng. J.* **2020**, *59*, 1307–1317. [CrossRef]
50. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; p. 9.
51. Chattopadhyay, A.; Sarkar, A.; Howlader, P.; Balasubramanian, V.N. Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 839–847. [CrossRef]
52. Zoubir, H.; Rguig, M.; Elaroussi, M. Crack recognition automation in concrete bridges using Deep Convolutional Neural Networks. *MATEC Web Conf.* **2021**, *349*, 03014. [CrossRef]
53. Inbac. Available online: https://github.com/weclaw1/inbac (accessed on 10 March 2022).
54. MCBDD-ZRE/Concrete-Bridge-Defects-Dataset. GitHub. Available online: https://github.com/MCBDD-ZRE/Concrete-Bridge-Defects-Dataset (accessed on 30 May 2022).
55. Lin, V. Vickyliin/Gradcam_Plus_Plus-Pytorch. 2022. Available online: https://github.com/vickyliin/gradcam_plus_plus-pytorch (accessed on 21 July 2022).