



Damage detection using in-domain and cross-domain transfer learning

Zaharah A. Bukhsh¹ · Nils Jansen² · Aaqib Saeed¹

Received: 8 December 2020 / Accepted: 26 June 2021 / Published online: 13 July 2021
© The Author(s) 2021

Abstract

We investigate the capabilities of transfer learning in the area of structural health monitoring. In particular, we are interested in damage detection for concrete structures. Typical image datasets for such problems are relatively small, calling for the transfer of learned representation from a related large-scale dataset. Past efforts of damage detection using images have mainly considered cross-domain transfer learning approaches using pre-trained IMAGENET models that are subsequently fine-tuned for the target task. However, there are rising concerns about the generalizability of IMAGENET representations for specific target domains, such as for visual inspection and medical imaging. We, therefore, evaluate a combination of in-domain and cross-domain transfer learning strategies for damage detection in bridges. We perform comprehensive comparisons to study the impact of cross-domain and in-domain transfer, with various initialization strategies, using six publicly available visual inspection datasets. The pre-trained models are also evaluated for their ability to cope with the extremely low-data regime. We show that the combination of cross-domain and in-domain transfer persistently shows superior performance specially with tiny datasets. Likewise, we also provide visual explanations of predictive models to enable algorithmic transparency and provide insights to experts about the intrinsic decision logic of typically black-box deep models.

Keywords Damage detection · Transfer learning · Pre-trained models · In-domain learning · Cross-domain learning · Visual inspection

1 Introduction

Civil structures such as bridges are reaching their end of service life due to aging, increased usage, and adverse climate impact [14]. Asset owners have to employ human experts to conduct periodic visual inspections to ensure structural safety and usability. In particular, so-called condition scorecards are used to rate the condition of bridges, whereas damage details are captured in images for further analysis [25]. Given that countries have thousands

of bridges, for instance, 60,000 bridges across the USA and 39,000 bridges in Germany, such expert-driven visual inspections are very costly and labor-intensive.

Toward a more automated process, unmanned aerial vehicles (UAVs) equipped with cameras, thermal infrared sensors, GPS, as well as light detection and ranging (LiDAR) sensors have proved useful to characterize cracks, spalls, concrete degradation, and corrosion [13, 36, 44]. Although UAVs are cost-effective and safe for harder-to-reach areas, they capture both damaged and structurally sound parts of bridges. As a consequence, an enormous amount of high-dimensional sensory data is created [9]. Manual damage identification from such an enormous data source demands tremendous efforts and is prone to discrepancies due to human errors, fatigue, and poor judgments of bridge inspectors [2, 46]. For image data, the involved subjectivity in a visual inspection process results in inaccurate outcomes and poses serious concerns for public safety [40], as shown by the collapsing incidents like the Malahide viaduct [50] or the I-35 Minneapolis bridge [63].

✉ Zaharah A. Bukhsh
z.bukhsh@tue.nl

Nils Jansen
n.jansen@science.ru.nl

Aaqib Saeed
a.saeed@tue.nl

¹ Eindhoven University of Technology, Eindhoven, The Netherlands

² Radboud University, Nijmegen, The Netherlands

The large unlabeled datasets from visual inspections call for state-of-the-art machine learning methods. In particular, we focus on damage detection of concrete surfaces. The field of computer vision has witnessed unprecedented developments since the advent of deep learning and the availability of large-scale annotated datasets, such as IMAGENET. Deep neural networks, specifically convolutional neural networks (CNNs), have shown strong performance in tasks such as object detection [20, 21, 53], image segmentation [29], image synthesis [47], and reconstruction [22], with promising applications in multiple domains.

While the use of deep learning techniques for structural health monitoring of civil structures is gaining momentum [18, 70],

to the best of our knowledge only six bridge inspection datasets [26, 28, 37, 11, 42, 67] are publicly available that can be utilized to develop automated damage detection models. These datasets are relatively small scale because of the tedious process of image acquisition and data labeling by experts.

Due to the limitation of small-scale datasets, recent studies for damage detection of bridges, tunnels, and roads have adopted transfer learning as the de facto standard for crack detection [7, 31, 52], pothole identification [39, 68], and related defect classification tasks [67]. These studies exclusively rely on cross-domain transfer learning, where IMAGENET is used as the source dataset (or upstream task), which is then fine-tuned for specific downstream tasks. However, there is skepticism about employing transfer learning for disparate target domains [19, 23, 27, 32]. For instance, [24] showed that reusing of IMAGENET features offers little benefit to performance when evaluated with medical imaging. Similar to medical images, bridge inspection datasets are fundamentally different from the IMAGENET dataset in terms of the number of classes, the quality of images, the size of the areas of interest, and the task specification.

The problem of choosing the relevant source dataset to learn general-purpose representations for the visual inspection target dataset is overlooked in the literature. Recent studies on damage detection adopt IMAGENET dataset as a default choice of source dataset, which falls under the realm of cross-domain transfer learning. This work seeks to eliminate this literature gap and performs a fine-grained investigation to study the potential advantages and downsides of *cross-domain transfer learning for structural damage detection*. We also learn and transfer *in-domain representation* and *its combination with cross-domain transfer strategies* for automated damage detection tasks. Specifically, in-domain transfer learning refers to a strategy in which the source and target dataset belong to a similar domain. Furthermore, we evaluate the transferability of learned representations under a low-data regime.

Besides transfer learning strategies, we also compared three state-of-the-art pre-trained models (InceptionV3, VGG16, and ResNet50) to small vanilla CNN models for the six publicly available bridge inspection datasets. The models are compared and evaluated using different initialization techniques, namely random, in-domain, and cross-domain initialization.

This work offers following contributions and findings:

- We conduct rigorous evaluations of transfer strategies with different initialization settings and network architectures for damage detection using the six publicly available visual inspection datasets. We also establish benchmarks for publicly available damage detection datasets.
- We develop multiple CNN models using cross-domain and in-domain transfer learning strategies for damage detection (classification) tasks.
- We find that the cross-domain transfer of pre-trained models proves helpful only when they are fine-tuned on the target task.
- We show that the smaller and simpler CNN models provide comparable performance to standard architectures when used as a fixed-feature extractor.
- We note that in-domain representations provide performance improvements compared to the random initialization.
- We show that the in-domain representations together with IMAGENET show performance gains compared to an independent transfer setting, particularly when the target dataset has fewer training samples.
- We provide visual explanations of multilayered deep neural networks to reveal their learning mechanisms. Such an interpretability analysis enables algorithmic transparency, validates the robustness of trained models, and gains practitioners' trust in typically black-box models.

The paper is structured as follows: Section 2 first introduces standard IMAGENET models, and then concerns related work about damage detection using deep transfer learning and feature transfer using pre-trained models. Section 3 introduces the publicly available bridge inspection datasets. Section 4 explains the methodology of our approach, along with the experimental setup. Section 5 presents the results of the experiments and provides comparisons to the baselines. An interpretability analysis of the developed models is given in Section 6. Finally, the conclusions of this study are presented in Section 7.

2 Background and related work

This section first provides a brief introduction to pre-trained models and their architectures for transfer learning. Next, we introduce related studies about automatic damage detection using deep (transfer) learning. Then, we highlight studies about possible limitations of generic feature transfer using pre-trained models. Additionally, for completeness, we highlight related topics from deep learning domains that are often associated with transfer learning.

2.1 Pre-trained models and standard neural architectures

Deep convolutional neural networks (CNNs) are a special type of multilayered neural network that have contributed to spectacular advancements in the computer vision field. Even though CNNs have been used for object detection tasks for several years [35], their widespread recognition and adoption are relatively recent, when the AlexNet model achieved state-of-the-art performance in the IMAGENET classification challenge [33]. Owing to the AlexNet success, the practice of training a CNN on IMAGENET (i.e., pre-training) and then adapting (e.g., fine-tuning) it to a target task has become a norm for several computer vision problems. Since pre-training on a large dataset is computationally expensive and time-consuming, several pre-trained models have been made publicly available by academics and industry.

In this study, we evaluate three state-of-the-art deep CNN models, namely VGG16 [55], Inception-v3 [58], and ResNet50 [23], which are lately being employed for damage detection tasks in structural health monitoring. An in-depth explanation of CNNs is beyond the scope of this study. The interested reader may refer to [65]. The section provides an architectural overview of the selected pre-trained models.

VGG16 invented by Visual Geometry Group of Oxford University [55] won the first place for IMAGENET object localization and the second place in the image classification challenge in 2014. Figure 1 provides a schematic representation of the VGG16 network. VGG16 improves over the AlexNet model

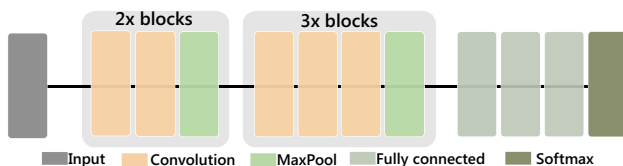


Fig. 1 Schematic representation of VGG16

by introducing a uniform structure of five convolution blocks having a fixed filter size of 3x3 with the stride of one. With its 13 convolutional layers and three fully connected layers, VGG16 demonstrated that the deeper models are desirable for improved classification accuracy. However, the VGG’s simple architecture comes at a high computational cost, with 138 million parameters (3x more than AlexNet) and prolonged training time. Inception-v3 improves the GoogleLeNet model that won the 2014’s IMAGENET classification challenge. In addition to classification accuracy, Inception-v3 focuses on reducing the computational expense, improving the training efficiency, and optimizing the network for easier adaptation to different use cases [59]. It is one of the first non-sequential models that emphasize a wider and deeper network by employing an *inception module*, as illustrated in Fig. 2. Instead of applying a single convolution with a fixed-size kernel, multiple convolutions with varying kernels (e.g., factorized 7x7) are applied to an input layer, which are then concatenated for another inception module. Additionally, to avoid the gradient vanishing problem, two auxiliary classifiers with batch normalization are included. Inception-v3 is 48 layers deep with 11 concatenated inception modules.

ResNet50 was the winner of famous 2015’s IMAGENET classification challenge. Residual network with several variants (such as ResNet10 and ResNet52) mainly dealt with the vanishing gradient problem. The residual network introduced an intuitive idea of skip connections with identity function mappings, as shown in Fig. 3. Specifically, the residual block learns the residual mappings, which captures the difference between actual input and approximate output rather than fitting the desired underlying mappings [24]. By approximating the identity function, residual learning does

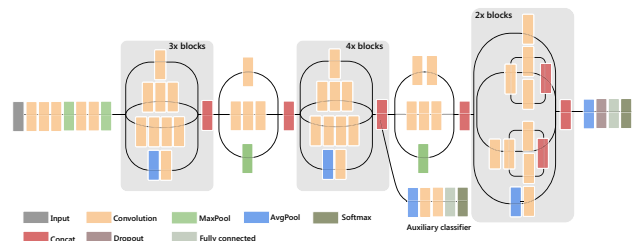


Fig. 2 Schematic representation of Inception-v3

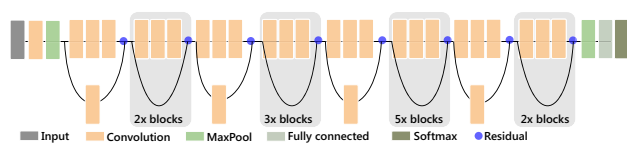


Fig. 3 Schematic representation of ResNet50

not introduce additional parameters and keeps the computational complexity similar to deep vanilla networks. ResNet50 consists of four stages, each containing two to six residual blocks with a kernel size of 1x1 and 3x3. In progressing from one stage to another, the channel width is doubled while the input size is reduced to half.

Various terminologies are interchangeably used to refer to the pre-trained models. The CNN architectures are referred to as *pre-trained models* or *IMAGENET models*, when pre-trained using IMAGENET weights. These networks without learned representations are referred as *standard architectures* due to their wide usage in multiple application domains.

2.2 Damage detection using deep (transfer) learning

Visual inspection datasets are small scale and are expensive to curate. Therefore, several studies adopt standard

architectures—such as VGG16, Inception-v3, LeNet, YOLO, or ResNet50 often pre-trained with IMAGENET representations—for the detection of cracks [31, 56, 61, 75], potholes [39], spalls [68], and multiple other damages including corrosion, seepage, and exposed bars [17, 28, 72, 78]. In Table 1, we provide a non-exhaustive list of recent literature studies that have used transfer learning for concrete damage detection tasks. It is notable that except for one, all studies have utilized IMAGENET as a source dataset. Additionally, pre-trained CNN models are also being utilized for vibration-based damage localization [1, 4, 69], condition assessment [30], and fault diagnosis [66].

Specific (visual) detection tasks in the damage recognition setting are strongly influenced by various operating conditions, such as surface reflectance, roughness, concrete materials, coatings, and weather phenomena for different components of a bridge [42]. Given the unique characteristics of visual inspection dataset, the problem of choosing an appropriate source dataset for damage detection remained an open question. This study seeks to address this by conducting a thorough investigation to evaluate the transferability of cross-domain representations (i.e., from IMAGENET) for damage detection tasks.

Table 1 List of recent studies that utilize transfer learning for concrete damage detection

Proposed method	Year	Task	Pre-trained model	Source dataset
CovNet Model [56]	2017	Concrete surface cracks	VGG16	IMAGENET
Custom CNN [3]	2017	Corrosion detection	VGG16	IMAGENET
CNN & edge detection [10]	2018	Crack detection in concrete	AlexNet	IMAGENET
Custom CNN [11]	2018	Crack detection in concrete	AlexNet	IMAGENET
Custom CNN [39]	2018	Road damage detection	MobiNet	IMAGENET
TERNAUSNET [5]	2019	Crack segmentation	VGG16	IMAGENET
Custom CNN [17]	2019	Structural damage detection	Inception-v3	IMAGENET
Custom CNN [78]	2019	Bridge defects detection	Inception-v3	IMAGENET
Multi-classifier [28]	2019	Concrete bridge defects	Inception-v3	IMAGENET
Custom CNN [12]	2019	Crack detection in concrete	VGG16	IMAGENET
Semantic seg. network [75]	2019	Crack detection in concrete	VGG16	IMAGENET
Custom CNN [64]	2020	Concrete surface roughness	ResNet	IMAGENET
Extension of YOLOv3 [73]	2020	Bridge damage detection	YOLOv3	COCO
Improved EfficientNetB0 [57]	2020	Crack detection in concrete	MobileNetV2	IMAGENET
Modified YOLO [61]	2021	Crack detection in concrete	YOLO	IMAGENET
Shallow CNN [31]	2021	Surface crack detection	LeNet-5	IMAGENET
Custom CNN [52]	2021	Road surface damage detection	ResNet34	IMAGENET
CNN for pixel level [7]	2021	Crack classification & seg.	Multiple models	IMAGENET

2.3 Feature transfer using pre-trained models

Transfer learning via IMAGE_{NET} representations is a widely adopted standard for learning from small-scale datasets across several domains. The availability of a multitude of pre-trained models such as MoBiNet, YOLO, ResNet, or GoogLeNet has further encouraged the application of transfer learning beyond the standard datasets consisting of day-to-day objects. The learned representations are typically utilized either via a fixed-feature extraction or fine-tuning methods depending on the target tasks.

Due to the widespread adoption of transfer learning, several fine-grained studies examine the transferability of features concerning the number of layers, the order of fine-tuning [38, 60, 71, 74], the generalizability of the architecture or learned weights [32, 48], and the characteristics of data used for pre-training [27]. Besides the popularity of transfer learning and its perceived usefulness, it has been argued that feature transfer does not necessarily deliver performance improvements compared to learning from scratch [23, 49]. Moreover, learned representations from pre-trained models may be less generic than often suggested [32], and the choice of data for pre-training is not as critical for the transfer performance [27].

2.4 Related topic

In the context of in-domain and cross-domain strategies, domain adaption is often misinterpreted as a general transfer learning method. In fact, domain adaption is a subcategory of transfer learning in which both target and source datasets are required during training, and the target dataset is weakly labeled or unlabeled [45]. However, in a standard transfer learning setting, the target dataset is supervised (labeled), and only pre-trained models can be used for learning without explicit access to the source dataset. Further details about transfer learning and proposed strategies are provided in the following sections.

3 Datasets

Due to the ubiquitous nature of images, several traditional application domains such as retail, automotive, or agriculture have benefited from massive labeled datasets to develop deep neural networks to solve various tasks. These natural datasets are relatively easy to label and do not require specific domain expertise. On the other hand, bridge inspection datasets are very scarce due to the vital domain knowledge required to identify and label damages. Moreover, the differences in human expertise for annotation make these datasets susceptible to noisy labels. To the

Table 2 Overview of (bridge) visual inspection datasets

Dataset	Instances	Classes	Problem
CDS [26]	1,027	2	Binary
SDNETv1 [11]	13,620	2	Binary
BCD [67]	5,390	2	Binary
ICCD [37]	60,010	2	Binary
MCDS [28]	2,411	10	Multi-label
CODEBRIM [42]	8,304	6	Multi-label

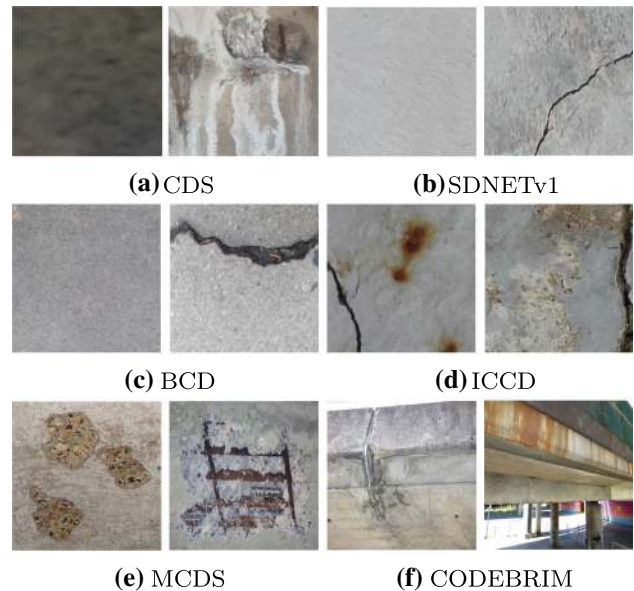


Fig. 4 Examples from the bridge inspection datasets. From left to right: (a) complex lighting with no crack, Crack with efflorescence, and exposed bars. (b) Intact concrete, minor crack, (c) Intact concrete, large crack on deck, (d) crack with rust stains, crack with minor scaling, (e) concrete scaling, corrosion with exposed bars, (f) spallation and efflorescence, rust stains

best of our knowledge, only six visual inspection datasets are publicly available. Datasets having less than 100 images are too small to train and evaluate CNNs and are therefore not considered here. Table 2 provides a brief overview of the datasets. A few typical example images are shown in Fig. 4.

Four of the datasets mainly focus on crack detection tasks. The other two datasets are for damage classification in a multi-label setting in which multiple damages coexist on a single input image. We briefly introduce the key characteristics of each dataset.

CDS [26]. The Cambridge dataset is a small bridge inspection dataset that seeks to detect concrete defects. The binary classes are divided into *healthy* and *unhealthy* of 691 and 337 images, respectively. Besides different luminous conditions on all images, the unhealthy class consists

of concrete damage such as cracking, graffiti, vegetation, and blistering.

SDNET [11]. The dataset has 230 images of reinforced concrete decks, walls, and pavements sub-segmented into 56000 images. However, we found misclassified label categories in the datasets. To utilize the data, we performed a manual cleaning of deck images only. The new dataset is referred as *SDNETv1*, which is highly imbalanced having 2, 025 *crack* and 11, 595 *uncracked* images.

BCD [67]. Bridge crack detection consists of 4, 058 *crack* and 2, 011 *background* images. The background images consist of healthy concrete surfaces. The dataset contains crack images with details like shading, water stains, and strong lights.

ICCD [37]. The image-based concrete crack detection dataset is one of the largest crack datasets having 60,000 images with an equal distribution of *crack* and *uncracked* classes. The images are captured using smartphone cameras under varying lighting conditions. In addition to cracks, the images predominately show corrosion strains.

MCDS [28]. The multi-classifier dataset has original inspection data and collected images from ten highway bridges. The authors defined the problem in a multistage classification manner. However, we used the dataset in a multi-label setting having ten classes, i.e., *crack* (789), *efflorescence* (311), *scaling* (168), *spalling* (427), *general defects* (264), *no defects* (452), *exposed reinforcement* (223), *no exposed reinforcement* (203), *rust straining* (355), and *no rust straining* (415).

CODEBRIM [42]. The COncrete DEfect BRidge Image Dataset provides the overlapping defects images of 30 bridges collected via camera and UAVs. The dataset consist of six classes, i.e., *cracks* (2507), *spalling* (1898), *efflorescence* (833), *exposed bars* (1507), *corrosion stain* (1559), and *background* (2506). The authors have framed the damage detection problem as a multi-target multi-class due to the overlapping damages in images. Here, we utilize the dataset in a multi-label setting only.

4 Methodology

4.1 In-domain and cross-domain transfer strategies

We propose to employ *transfer learning* as a natural solution for learning and improving the performance of damage detection models on small labeled datasets, as shown in numerous other vision tasks [45]. Transfer learning attempts to transfer the learned knowledge from a *source task* to improve the learning in a related *target task* [62]. Here, we consider variants of transfer learning, referred to as *cross-domain* and *in-domain*, and their

combination as shown in Fig. 5. The main difference between these approaches is the type of dataset used as a source task (also called the upstream dataset).

In *cross-domain transfer learning*, a convolutional neural network is trained over a large IMAGENET dataset having one million images of generic objects [8]. The learned representations are then fine-tuned for the specific downstream tasks, such as for damages detection. Given the computational requirements of network training using the massive IMAGENET dataset, several pre-trained models are made available by academia and industry. The pre-trained models, such as VGG16, Inception-v3, and ResNet50, have shown to yield state-of-the-art performance for object detection and image localization in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). The pre-trained models consist of representations (or weights) learned from IMAGENET and a standard CNN architecture.

For *in-domain transfer learning*, the upstream and downstream datasets belong to a similar application domain, for instance, in medical imaging and asset inspection, among others. For comparison purposes, we utilize standard CNN architectures for training on a visual inspection dataset, which are then fine-tuned for a specific task. It is important to note that the dataset used for upstream and downstream tasks are mutually exclusive as depicted by one less square in Fig. 5b. In theory, in-domain transfer learning should provide improved model performance compared to cross-domain due to the availability of similar visual concepts. However, cross-domain transfer learning is standard in the computer vision field as representations are learned from a large dataset having diverse classes.

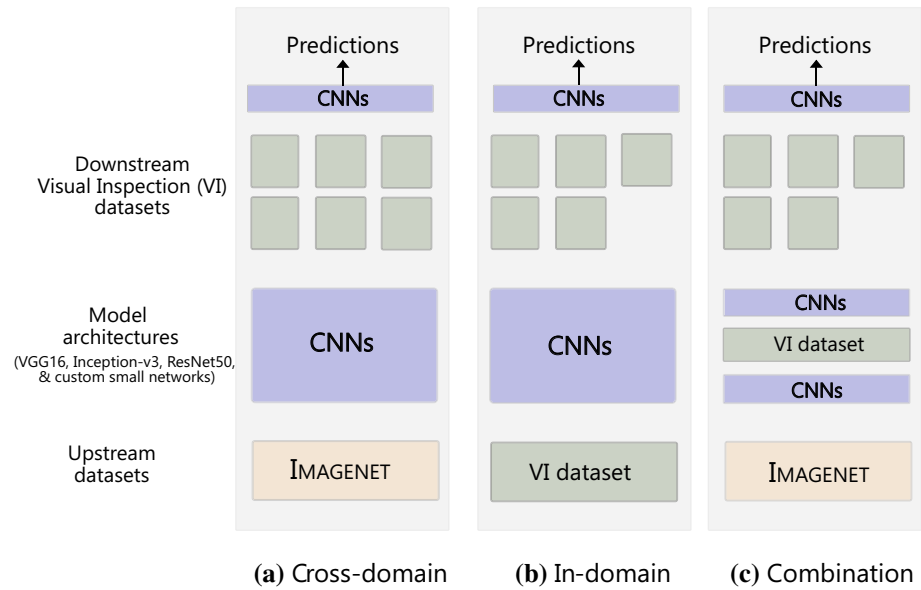
We also study the *combination of in-domain and cross-domain transfer learning* to enable further performance improvements for damage detection tasks. Here, the pre-trained models having IMAGENET representations are first trained on a visual inspection dataset before fine-tuning to a specific damage detection problem. The combination transfer strategy is depicted in Fig. 5c.

4.2 Details of the experiments

We conduct several experiments to evaluate different transfer strategies for improved damage detection tasks. The experiments seek to answer the following questions:

- Do features reused from standard pre-trained models prove helpful compared to learning from scratch for damage detection task?
- Does the standard architecture perform better than small custom CNN models?

Fig. 5 Schematic representation of transfer learning strategies



- Does in-domain transfer learning provide a performance benefit compared to the state-of-the-art cross-domain transfer approach?
- Does the combination of in-domain and cross-domain perform better than their stand-alone versions?
- How well do the representations generalize given an extremely small downstream training dataset?

To acquire pre-trained (IMAGENET) features, we utilize pre-trained models, namely VGG16, Inception-v3, and ResNet50. The choice of these models is motivated by the fact that they are widely used for damage detection tasks, as shown in the literature [6, 28, 34, 68, 77]. The traditional strategies for leveraging transfer learning include: (i) using a pre-trained model as a fixed-feature extractor (FE) in which pre-trained layers are kept frozen, (ii) fine-tuning (FT) all or a few of the layers of an existing model so that the weights are updated for the target task, and (iii) training IMAGENET weights from scratch and then fine-tuning on the target task. Since training on IMAGENET from scratch is computationally expensive, in our experiments, we opted for feature extraction and fine-tuning strategies appended with additional convolution (2D) and a dropout layer to avoid overfitting.

For the comparative evaluation of standard architecture, four small and simpler CNNs models are also adopted from [49]. [49] referred to these models as a *CBR* family and used them to evaluate pre-trained networks for medical imaging. The simpler network consists of four to five layers of convolution, batch normalization, and ReLU activation with varying numbers of filters and kernel sizes. Depending on the architecture configuration, the size of CBR CNNs networks ranges from a third of the IMAGENET architecture size to one-twentieth of the size as described in [49]. Due

to the extensive computation costs of training IMAGENET, we perform architectural comparisons with random initialization only.

Besides cross-domain transfer, we investigate in-domain transfer learning for structural damage detection where a dataset from a similar domain is utilized for upstream and downstream tasks. For comparison purposes, we evaluated the best performing random initialized and IMAGENET model for the specific downstream damage detection task. Finally, we conduct several experiments with fewer labeled examples from training sets to assess the quality of learned representation from cross-domain and in-domain learning within a low-data regime.

4.3 Evaluation approach and metrics

For the performance evaluation, we employ the held-out approach in which the complete dataset is split into training, validation, and test split with an approximate ratio of 70%, 10%, and 20%. For ICCD and CODEBRIM dataset, the predefined train/validation/test ratios are used. The validation set is used for the hyper-parameter tuning of the model. We perform five independent runs of the models for 30 epochs with a 64 batch size for all the experiments. The learning rate is kept as low as 10^{-4} with a binary or categorical cross-entropy loss function depending on the problem. It is important to reiterate that the main focus of this study is to evaluate the network architecture and the source dataset characteristics in the sense of transferring robust representations for structural damage detection. We deliberately avoid the extensive hyper-parameters tuning to compare dataset characteristics and architectural impact on performance. We use the standard parametric configuration

of the Keras framework unless explicitly stated above. For the hardware, we use a machine with two Nvidia Tesla T4 GPUs on the Google Cloud Platform.

Additionally, for the performance evaluation and comparison of the deep models, we compute the **AUC-ROC** (area under the receiver operating characteristic curve) score, which is one of the most robust evaluation metrics for classification tasks having skewed class distribution [16]. AUC-ROC computes the area under the curve across all classification thresholds and summarizes the ROC in a single number. We provide a comparison of models using the AUC-ROC score since it is a classification threshold invariant. This means that AUC-ROC score provides information about the model's prediction quality irrespective of the specifically chosen threshold value. The AUC-ROC score ranges in value from 0 to 1. The best performing model will achieve a score closer to 1. A classifier with a zero or 0.5 score is considered random with no predictive capability. For the sake of completeness, we also report accuracy, F-score, precision, and recall scores in Appendix 1.

5 Results and comparisons

5.1 Transferability of pre-trained models

Five independent runs (i.e., training and evaluation) of CBR models and pre-trained IMAGENET models are performed on six different bridge inspection datasets. The averaged AUC-ROC performance scores and standard deviation are shown in Table 3. First, we note that transfer learning with *fine-tuning* performed consistently well for

the majority of bridge inspection datasets, as depicted with bold entries in Table 3.

VGG16 with fine-tuned weights achieved the best performance on all datasets except for CODEBRIM dataset. For BCD, ICCD, and CODEBRIM datasets, the results from Inception-v3 and ResNet50 models are also comparable with the difference of 0.1 scores only. Next, in a *random initialization* setting, the vanilla CBR model, with their one-twentieth to the standard architecture size, performed best on four datasets, followed by VGG16 for the other two datasets as depicted with underlined entries. The relatively sub-optimal performance of standard architectures can be attributed to overfitting due to a heavily parameterized network.

Finally, we note that the *feature extraction* yielded a poor performance compared to random initialization and fine-tuning. This poor performance may be due to considerable differences in the IMAGENET and bridge inspection datasets regarding the number of classes, quality of images, size of the area of interest, and task specification.

5.2 Impact of in-domain representation learning

In-domain representation learning typically follows two steps [43]: (a) Learning of representations using in-domain data, called *upstream* training, (b) evaluation of learned representations by transferring them to a new and unseen task, termed as *downstream*. For upstream, we utilize the best performing models that were trained from scratch, and fine-tuned IMAGENET models (see Table 3). The learned representations are further fine-tuned on all the new and unseen downstream tasks.

Table 3 AUC-ROC performance scores of CBR models and standard IMAGENET architectures with different initializations. VGG16 with fine-tuning performed best for five datasets, followed by ResNet50, as shown with bold entries. Under random initialization and fixed-weight settings, the CBR models performed better than the standard IMAGENET architecture for four datasets followed by VGG16, as shown with underlined entries

	CDS	SDNETv1	BCD	ICCD	MCDS	CODEBRIM
Random Initialization						
CBR Tiny	<u>0.79</u> ± 0.02	<u>0.74</u> ± 0.01	0.97 ± 0.00	0.93 ± 0.13	<u>0.67</u> ± 0.02	0.80 ± 0.04
CBR Small	0.78 ± 0.02	0.72 ± 0.01	0.96 ± 0.0	<u>0.98</u> ± 0.0	0.67 ± 0.02	0.80 ± 0.03
CBR LargeW	0.75 ± 0.03	0.74 ± 0.01	0.97 ± 0.01	0.96 ± 0.02	0.67 ± 0.01	0.80 ± 0.04
CBR LargeT	0.74 ± 0.02	0.67 ± 0.0	0.96 ± 0.00	0.96 ± 0.02	0.64 ± 0.01	0.79 ± 0.01
VGG16	0.78 ± 0.04	0.50 ± 0.00	<u>0.98</u> ± 0.01	0.78 ± 0.26	0.59 ± 0.01	<u>0.82</u> ± 0.01
Inception-v3	0.69 ± 0.03	0.60 ± 0.06	0.90 ± 0.03	0.96 ± 0.01	0.61 ± 0.01	0.77 ± 0.02
ResNet50	0.69 ± 0.02	0.61 ± 0.08	0.60 ± 0.20	0.93 ± 0.03	0.57 ± 0.01	0.73 ± 0.02
Fine-tuning of pre-trained models.						
VGG16	0.82 ± 0.02	0.84 ± 0.0	0.99 ± 0.0	0.98 ± 0.0	0.77 ± 0.03	0.88 ± 0.01
Inception-v3	0.73 ± 0.02	0.78 ± 0.01	0.98 ± 0.00	0.98 ± 0.0	0.72 ± 0.01	0.89 ± 0.0
ResNet50	0.62 ± 0.02	0.82 ± 0.01	0.98 ± 0.01	0.98 ± 0.0	0.54 ± 0.01	0.90 ± 0.01
Feature extractions from pre-trained models.						
VGG16	<u>0.77</u> ± 0.01	<u>0.72</u> ± 0.02	<u>0.98</u> ± 0.01	<u>0.93</u> ± 0.00	<u>0.64</u> ± 0.01	<u>0.78</u> ± 0.00
Inception-v3	0.55 ± 0.01	0.50 ± 0.01	0.52 ± 0.02	0.67 ± 0.04	0.54 ± 0.0	0.55 ± 0.01
ResNet50	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.01	0.50 ± 0.00	0.50 ± 0.00

Table 4 AUC-ROC performance scores for in-domain transfer per dataset. The rows represent the upstream source model trained for the target task, and the column shows the results of in-domain transfer for

each dataset. The bold entries depict the best performing models compared to the random initialization (given in the first row)

	CDS	SDNETv1	BCD	ICCD	MCDS	CODEBRIM
Rand. Init.	0.79 ± 0.02	0.74 ± 0.01	0.98 ± 0.01	0.97 ± 0.0	0.67 ± 0.02	0.82 ± 0.01
CDS		0.73 ± 0.05	0.99 ± 0.00	0.96 ± 0.01	0.67 ± 0.02	0.80 ± 0.03
SDNETv1	0.75 ± 0.01		0.98 ± 0.00	0.97 ± 0.00	0.67 ± 0.01	0.77 ± 0.03
BCD	0.77 ± 0.04	0.81 ± 0.00		0.98 ± 0.00	0.61 ± 0.01	0.81 ± 0.00
ICCD	0.78 ± 0.02	0.80 ± 0.00	1.00 ± 0.00		0.69 ± 0.00	0.76 ± 0.03
MCDS	0.76 ± 0.02	0.72 ± 0.07	0.98 ± 0.00	0.97 ± 0.01		0.79 ± 0.02
CODEBRIM	0.75 ± 0.04	0.80 ± 0.01	0.98 ± 0.00	0.98 ± 0.00	0.70 ± 0.01	

Upstream model architecture: CBR Tiny (CDS, SDNETv1, MCDS) CBR Small (ICCD) VGG16 (BCD, CODEBRIM)

Table 4 provides the results of the in-domain transfer. For comparison purposes, the results from random initialization are also provided. The first column depicts the upstream datasets, and the rest of the columns’ names are downstream datasets. Out of six datasets, the in-domain transfer yields performance improvement for at least four datasets than learning from scratch. SDNETv1 and MCDS show notable performance improvements with an increase of 0.7 and 0.2 in the AUC-ROC score, respectively. Additionally, BCD and ICCD datasets predominantly showed good transfer of in-domain knowledge across all the target tasks, despite the considerable difference in their sizes.

We also evaluate the *combination of in-domain and cross-domain* transfer for performance gains. Compared to discrete IMAGENET or in-domain transfer, the combination shows further performance improvement for the majority of datasets, as shown with bold entries in Table 5. With the in-domain and cross-domain transfer combination, it is difficult to remark about the usefulness of a specific dataset since the performance scores across the different source datasets are very similar. When used as a source, most of

the datasets show comparable performance results irrespective of their size and number of classes. It is essential to mention that the combination of in-domain and cross-domain transfer can introduce additional computation costs due to fine-tuning of large pre-trained models. In the practical setting, the performance gains and computational cost can be compared to gauge the usefulness of this approach.

Figure 6 presents the comparison of loss and performance during training of network with four different initialization settings. These comparisons show the convergence capacity of the network with respect to a specific dataset and initialized weights. It is noted that the network initialization with in-domain and combination of in-domain and cross-domain converges faster than IMAGENET for binary classification datasets, namely for CDS, SDNETv1, and BCD. For multi-label dataset, the IMAGENET network shows better convergence capacity.

Table 5 AUC-ROC performance scores for in-domain and IMAGENET transfer. The rows represent the upstream model of in-domain and IMAGENET representations, and the column entries show the results of

transfer for each dataset. The bold entries depict the best performing models compared to transfer from IMAGENET only (given in the first row)

	CDS	SDNETv1	BCD	ICCD	MCDS	CODEBRIM
IMAGENET	0.82 ± 0.02	0.84 ± 0.0	0.99 ± 0.0	0.98 ± 0.0	0.77 ± 0.03	0.90 ± 0.01
CDS		0.84 ± 0.01	0.99 ± 0.0	0.99 ± 0.0	0.75 ± 0.02	0.885 ± 0.01
SDNETv1	0.76 ± 0.02		0.99 ± 0.0	0.98 ± 0.0	0.70 ± 0.02	0.86 ± 0.01
BCD	0.80 ± 0.03	0.85 ± 0.0		0.98 ± 0.0	0.75 ± 0.01	0.88 ± 0.01
ICCD	0.81 ± 0.02	0.84 ± 0.01	1.00 ± 0.0		0.70 ± 0.01	0.87 ± 0.01
MCDS	0.84 ± 0.02	0.83 ± 0.02	0.99 ± 0.0	0.98 ± 0.0		0.88 ± 0.0
CODEBRIM	0.82 ± 0.02	0.81 ± 0.01	0.99 ± 0.0	0.98 ± 0.0	0.79 ± 0.0	

Upstream model architecture: VGG16 architecture for all except for CODEBRIM which used ResNet50

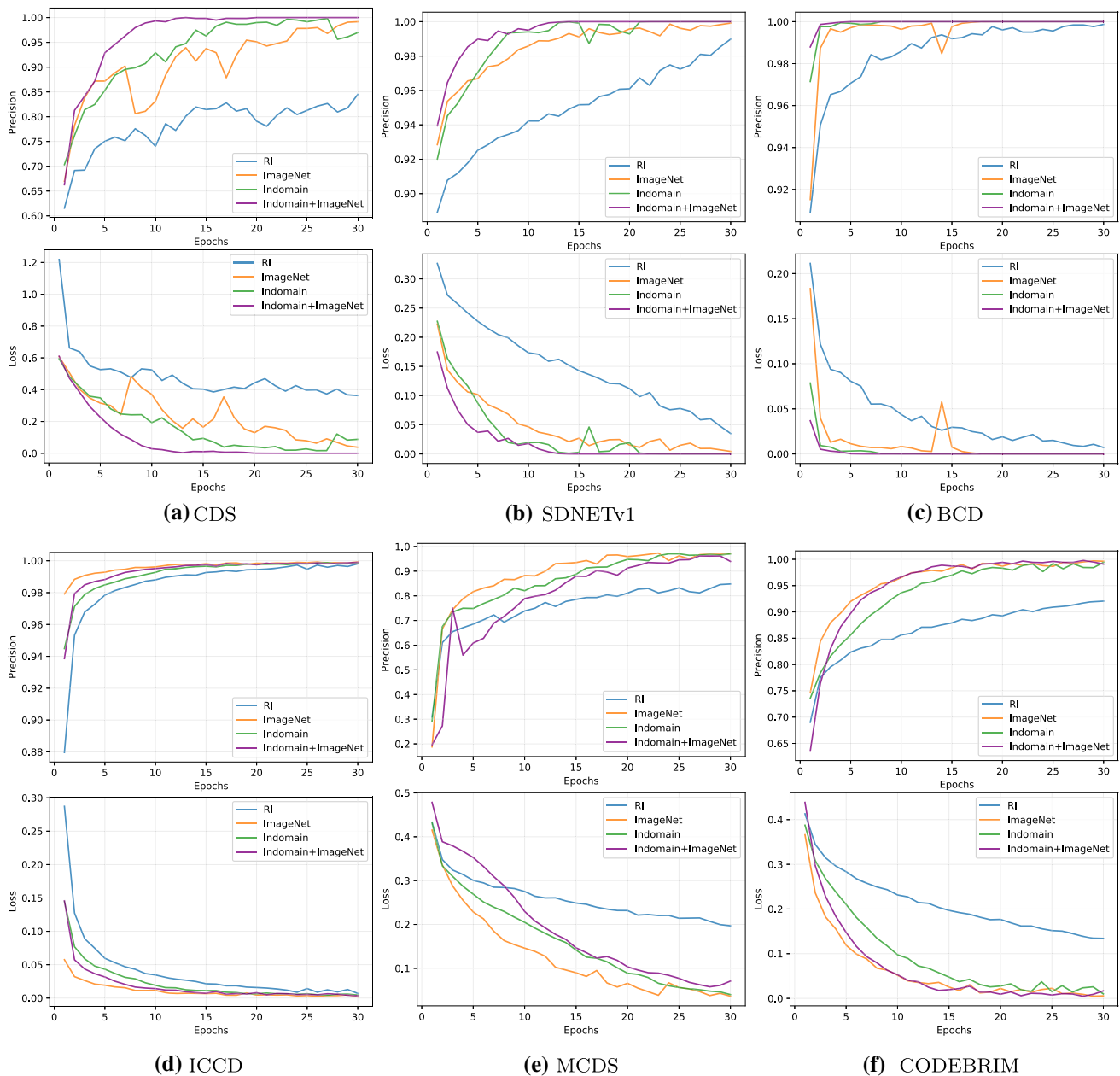


Fig. 6 Comparison of network training (with precision and loss) for 30 epochs with different initialization settings. RI: random initialization, IMAGE NET: pre-trained VGG16 with IMAGE NET, in-domain: pre-

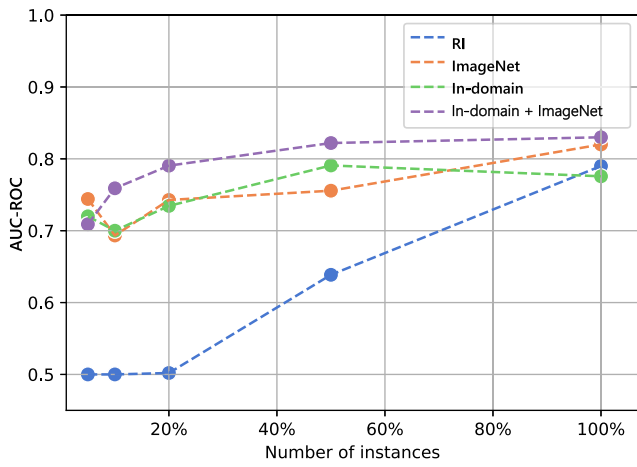
trained with one of the visual inspection datasets, in-domain+IMAGE NET: combination of in-domain and IMAGE NET pre-training

5.3 Effectiveness of transfer learning in a low-data regime

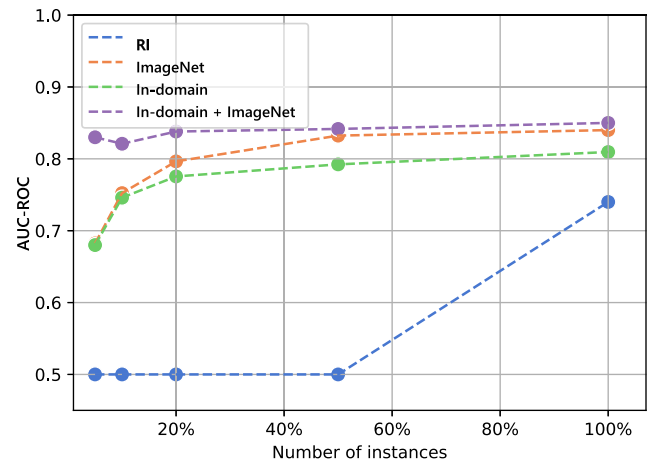
Bridge inspection datasets are typically smaller than IMAGE NET due to the laborious process of acquiring the labeled images. This lack of data necessitates the effective transfer of knowledge from pre-trained models. Thereby, to further explore the generalizability of in-domain and cross-domain representations for a low-data regime, we perform further

experiments with a smaller number of training samples ranging from 5%, 10%, 20%, and 50% of the datasets.

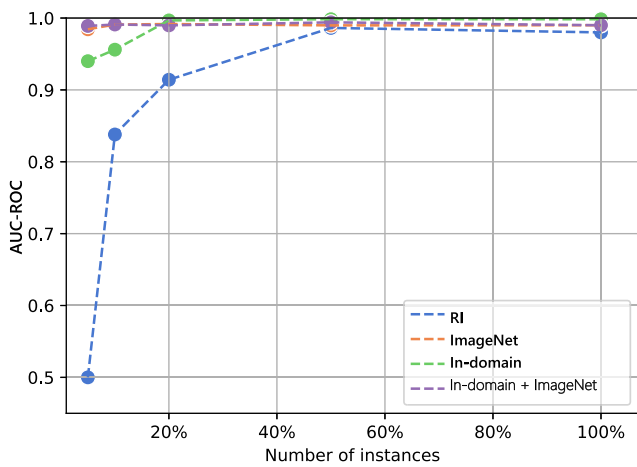
Figure 7 reports the AUC-ROC performance score with different initialization settings for fewer training samples of datasets. Transfer learning either with in-domain or cross-domain datasets provides significant performance gains with very small datasets compared to learning from scratch. The combination of in-domain and ImageNet transfer show significant performance with as few as 5% of training samples only.



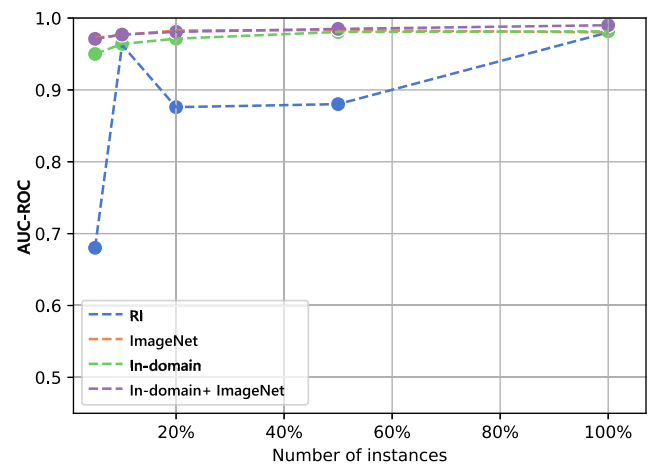
(a) CDS



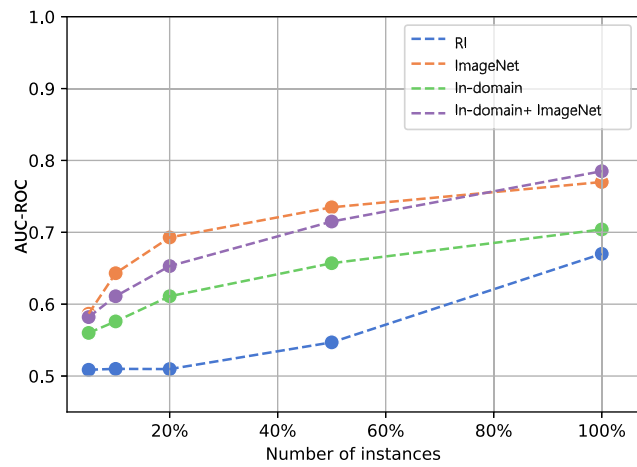
(b) SDNETv1



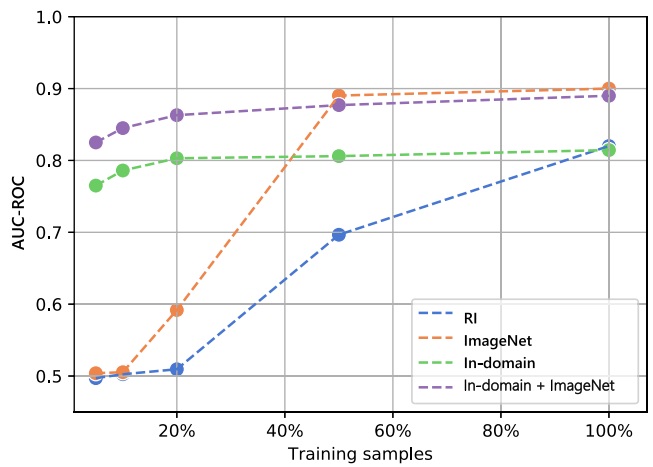
(c) BCD



(d) ICCD



(e) MCDS



(f) CODEBRIM

Fig. 7 AUC-ROC score on test set after training with different initialization over a limited number of training samples

Despite significant size differences of the IMAGENET dataset, having millions of images, and the largest in-domain dataset with 60,000 images, their transfer shows comparable performance for binary datasets. For multi-label datasets, i.e., MCDS and CODEBRIM datasets, the impact of different initialization on performance scores is notable. For the CODEBRIM dataset, the in-domain transfer performs notably well compared to IMAGENET when only 5% to 20% data is used. In contrast, with the increasing size of the training samples, the IMAGENET surpasses the other initializations.

To summarize, the combination of in-domain and cross-domain transfer is a desirable transfer setting when the target data is extremely small. However, with larger target datasets, the combination of in-domain and cross-domain transfer yield similar performance as when only IMAGENET is used as source dataset.

5.4 Comparison with baselines

This section conducts a comparison to the baselines to assess the performance of the proposed transfer learning strategies. Table 6 presents the accuracy scores for each dataset and provides the comparison to the baselines. Transfer learning with a combination of in-domain and cross-domain strategies shows improved results for at least four datasets as depicted by bold entries. For the other two datasets, i.e., [42] and [28], a direct comparison is not feasible due to the disparity in research methodologies. [42] treated the damage classification problem in multi-label multi-target settings in which the exact match of predicted and actual multi-labels is ensured. [28] dealt with damage classification in a multistage multi-classification manner, where several multi-class and binary classifier are defined. Instead, in our approach, we develop a single classifier for multiple damages detection. Besides transfer learning, the results provided in Table 6 can be used for future performance comparisons and improvements for the publicly available visual inspection datasets.

It is worth noting that different IMAGENET architectures and transfer learning strategies introduced varying computational costs. Pre-trained IMAGENET and in-domain

models can be efficient for feature extraction and fine-tuning. The combination of cross-domain and in-domain models can be computationally expensive. However, the combination of cross and in-domain transfer can provide superior performance when the target dataset is extremely small.

6 Interpretability of damage detection models

Transparency and interpretability of predictive models are indispensable to enable their use in practice. Deep neural networks are well known for exploiting millions of parameters, processed by several nonlinear functions and pooling layers to learn optimal weights. It is intractable for humans to follow the exact mapping of data from input to classification through these complex multilayered networks. To encourage CNN's usage in practice, the interpretability and explainability of deep neural networks are increasingly popular and an active research area. Several model-agnostic, visual explanations and example-based methods have recently been proposed [15, 41].

In this study, we employed gradient-weighted class activation mapping (Grad-CAM) to visualize and localize the areas of an input image that are most important for the models' prediction. Grad-CAM is a class-discriminative localization technique that, unlike its predecessor [76], does not require any change in the CNNs architecture or retraining for generating visual explanations [54]. Several studies have shown that the deeper layers of CNNs capture high-level abstract constructs; therefore, Grad-CAM utilizes the gradient information flowing into the last convolution layer to evaluate each neuron's importance. We apply importance scores with logistic regression and a nonlinear activation function to obtain coarse heatmaps for discriminative class mappings.

Figure 8 shows the visual explanation of the fine-tuned VGG16 model, which has predominantly performed well for most of the damage detection datasets (see Table 3 for results). We utilized the tf-explain implementation of Grad-CAM to generate visual mapping [51]. In addition to

Table 6 Accuracy scores of transfer learning strategies and comparisons with baselines. The bold entries depict the best performing transfer learning strategy

	CDS (%)	SDNETv1 (%)	BCD (%)	ICCD (%)	MCDS (%)	CODEBRIM (%)
Rand. Init.	82	94	98	98	68	82
Cross-domain	86	95	98	98	76	90
In-domain	83	95	99	98	69	82
Combination	87	96	99	99	79	90
Previous studies	–	92 [11]	96 [67]	99 [37]	85* [28]	72** [42]

*Multiple (binary and multi-class) classifiers approach

**Multi-target accuracy score

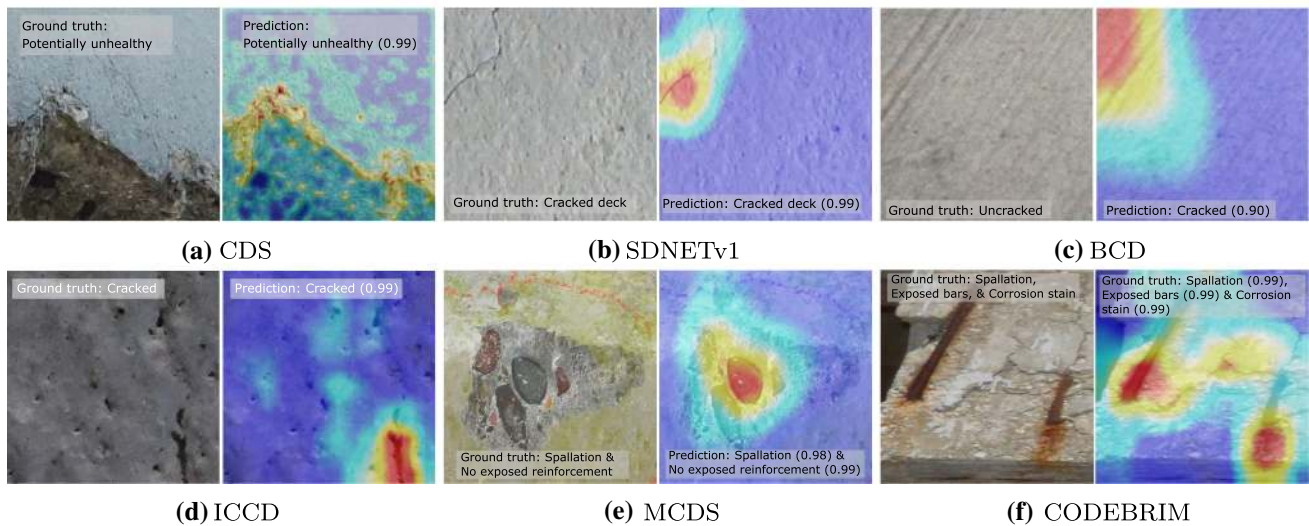


Fig. 8 Visual explanations for classification and localization of VGG16 models for all six datasets. Grad-CAM highlights class discriminative features. The heatmap localizes the specific damage regions

the last convolution layer of CNN, Grad-CAM can be used to extract discriminative class mapping with any intermediate convolution layer. We experimented with various convolution and pooling layers to generate visual explanations, as shown in Fig. 8. The visual explanations with the correct predicted class validate that the model is paying attention to the right areas of an image to identify and localize a specific damage category. It is interesting to notice Fig. 8, where the model misclassified the healthy patch of concrete as cracked by assigning higher weights to the textured concrete patch.

The visual interpretations of CNN models reveal *why the models predict, what they predict*. The visual explanation also provides localization of specific damage in an input image without additional labeling or segmentation activity. By revealing the decision logic of these complex models, the decision-makers and infrastructure managers can trust them for automatic damage detection tasks.

7 Conclusions

This paper presents transfer learning strategies and conducts comprehensive experiments to evaluate their usefulness for damage detection tasks. We compared different initialization settings, namely random initialization, in-domain, cross-domain, and their combination for transfer learning on six publicly available visual inspection datasets of concrete bridges. We found that cross-domain transfer yields performance improvements only when fine-tuned for the target damage detection task.

Our main message is that the combination of in-domain and cross-domain representations provide enhanced

performance compared to their stand-alone versions. Additionally, in contrast to pre-trained IMAGENET models, in-domain transfer provides training efficiency and flexibility in selecting relatively smaller yet powerful CNNs architecture. In our exploration of learning for tasks with a limited number of training samples, in-domain and IMAGENET representations show comparable performance. The results demonstrate considerable performance gains when in-domain and cross-domain (IMAGENET) representations are used jointly for the target dataset having fewer training samples.

We further assessed the best performing IMAGENET models by developing visual explanations using gradient-weighted class activation mapping. The visual explanation shows the class discriminative regions of an input that are most paramount for specific damage prediction. Additionally, the interpretability of models' prediction localizes the specific damages and enables decision-makers to understand the underlying intrinsic decision logic of the neural model. Such visual exploration of damage detection and localization also encourages the use of predictive models in practice.

Additional results

We provide additional results to compare different transfer learning strategies in Table 7. In addition to AUC-ROC scores, we provide accuracy, F-score, precision, and recall measures. *Accuracy* is essentially computed by taking a fraction of correctly predicted samples to the total number of samples. However, in the case of an imbalanced dataset, *accuracy* as performance metric can be misleading.

Table 7 Performance evaluation of transfer learning strategies with different initialization

		CDS	SDNETv1	BCD	ICCD	MCDS	CODEBRIM
RI	Acc.	0.804	0.890	0.976	0.788	0.589	0.821
	F-score	0.802	0.838	0.977	0.721	0.229	0.724
	Precision	0.812	0.791	0.981	0.688	0.194	0.741
	Recall	0.804	0.890	0.976	0.788	0.375	0.745
Cross-domain	Acc.	0.864	0.955	0.986	0.985	0.780	0.900
	F-score	0.860	0.952	0.986	0.984	0.629	0.847
	Precision	0.865	0.953	0.987	0.985	0.697	0.859
	Recall	0.864	0.955	0.986	0.985	0.638	0.850
In-domain	Acc.	0.711	0.948	0.975	0.980	0.690	0.800
	F-score	0.707	0.944	0.976	0.980	0.465	0.681
	Precision	0.708	0.945	0.979	0.980	0.579	0.751
	Recall	0.711	0.948	0.975	0.980	0.488	0.704
Combination	Acc.	0.869	0.955	0.989	0.986	0.785	0.885
	F-score	0.864	0.953	0.989	0.986	0.642	0.822
	Precision	0.871	0.953	0.990	0.986	0.680	0.825
	Recall	0.869	0.955	0.989	0.986	0.623	0.834

Therefore, we report weighted accuracy and weighted *F-score* here. The *weighted* aspect considers the data imbalance of the target class and assigns the weights to data samples accordingly. The *F-score* is a combination of precision (or positive predictive value) and recall (sensitivity) measures. The precision determines the exactness of the model, whereas the recall provides a measure of the model's completeness.

Author Contributions All authors contributed equally to the study conception and design. Material preparation, data collection, and analysis were performed by first author. The draft of the manuscript was written by first author and all authors reviewed and corrected multiple versions of the manuscript. All authors read and approved the final manuscript.

Funding This research has been partially funded by NWO under the grant PrimaVera NWA.1160.18.238.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Code available The source code is publicly available at <https://github.com/Zaharah/indomain-cross-domain-transfer-learning>

Material available The research has used publicly available dataset.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless

indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abdeljaber O, Avci O, Kiranyaz MS, Boashash B, Sodano H, Inman DJ (2018) 1-D CNNs for structural damage detection: verification on a structural health monitoring benchmark data. *Neurocomputing* 275:1308–1317
- Agdas D, Rice JA, Martinez JR, Lasa IR (2016) Comparison of visual inspection and structural-health monitoring as bridge condition assessment methods. *J Perform Construct Facil* 30(3):04015049
- Atha DJ, Jahanshahi MR (2018) Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Struct Health Monit* 17(5):1110–1128
- Azimi M, Pekcan G (2020) Structural health monitoring using extremely compressed data through deep learning. *Comput Aided Civ Inf* 35:597–614. <https://doi.org/10.1111/mice.12517>
- Benz C, Debus P, Ha HK, and Rodehorst V (2019) Crack segmentation on UAS-based imagery using transfer learning. In: *International Conference on Image and Vision Computing New Zealand (IVCNZ)*, 2019, pp 1–6. <https://doi.org/10.1109/IVCNZ48456.2019.8960998>
- Carr TA, Jenkins MD, Iglesias MI, Buggy T, Morison G (2018) Road crack detection using a single stage detector based deep neural network. In: *2018 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, pp 1–5. <https://doi.org/10.1109/EESMS.2018.8405819>
- Dais D, Bal IE, Smyrou E, Sarhosis V (2021) Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. *Autom Construct* 125:103606
- Deng J, Dong W, Socher R, Li L, Kai Li, Li Fei-Fei (2009) IMAGENET: a large-scale hierarchical image database. In: 2009

- IEEE Conference on Computer Vision and Pattern Recognition, pp 248–255. <https://doi.org/10.1109/CVPR.2009.5206848>
9. Dorafshan S, Maguire M (2018) Bridge inspection: human performance, unmanned aerial systems and automation. *J Civ Struct Health Monit* 8(3):443–476
 10. Dorafshan S, Thomas RJ, Maguire M (2018) Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construct Build Mater* 186:1031–1045
 11. Dorafshan S, Thomas RJ, Maguire M (2018) SDNET2018: an annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data Brief* 21:1664–1668
 12. Dung CV et al (2019) Autonomous concrete crack detection using deep fully convolutional neural network. *Autom Construct* 99:52–58
 13. Ellenberg A, Kotsos A, Moon F, Bartoli I (2016) Bridge deck delamination identification from unmanned aerial vehicle infrared imagery. *Automation in Construction* 72:155–165
 14. European Commission (2018) Transport in the European union, current trends and issues. Mobility and Transport. <https://bit.ly/2BuDRpy>. Accessed 5 May 2020
 15. Fan F, Xiong J, Wang G (2021) On interpretability of artificial neural networks: a survey. In: *IEEE Transactions on Radiation and Plasma Medical Sciences*. <https://doi.org/10.1109/TRPMS.2021.3066428>
 16. Fawcett T (2006) An introduction to roc analysis. *Pattern Recog Lett* 27(8):861–874
 17. Feng C, Zhang H, Wang S, Li Y, Wang H, Yan F (2019) Structural damage detection using deep convolutional neural network and transfer learning. *KSCE J Civ Eng* 23(10):4493–4502
 18. Fink O, Wang Q, Svensén M, Dersin P, Lee W-J, Ducoffe M (2020) Potential, challenges and future directions for deep learning in prognostics and health management applications. *Eng Appl Artif Intell* 92:103678
 19. Geirhos R, Rubisch P, Michaelis C, Bethge M, Wichmann FA, Brendel W (2018) ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*
 20. Geng Y, Liang R-Z, Li W, Wang J, Liang G, Xu C, Wang J-Y (2016) Learning convolutional neural network to maximize pos@top performance measure. In: *ESANN 2017 - Proceedings*, pp 589–594
 21. Geng Y, Zhang G, Li W, Gu Y, Liang R-Z, Liang G, Wang J, Wu Y, Patil N, Wang J-Y (2017) A novel image tag completion method based on convolutional neural transformation. *Lecture Notes in Computer Science*, pp 539–546. https://doi.org/10.1007/978-3-319-68612-7_61
 22. Han X, Laga H, Bannamoun M (2021) Image-based 3D object reconstruction: state-of-the-art and trends in the deep learning era. *IEEE Trans Pattern Anal Mach Intell* 43: 1578–1604. <https://doi.org/10.1109/TPAMI.2019.2954885>
 23. He K, Girshick R, Dollár P (2019) Rethinking imagenet pre-training. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp 4918–4927
 24. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 770–778. <https://doi.org/10.1109/CVPR.2016.90>
 25. Highways Agency (2007) Inspection manual for highway structures: Vol. 1: Reference manual. The Stationery Office
 26. Huethwohl P (2017) Cambridge Bridge Inspection Dataset [Dataset]. <https://doi.org/10.17863/CAM.13813>
 27. Huh M, Agrawal P, Efros AA (2016) What makes imagenet good for transfer learning? Preprint [arXiv:1608.08614](https://arxiv.org/abs/1608.08614)
 28. Hüthwohl P, Lu R, Brilakis I (2019) Multi-classifier for reinforced concrete bridge defects. *Autom Construct* 105:102824
 29. Jégou S, Drozdal M, Vazquez D, Romero A, Bengio Y (2017) The one hundred layers tiramisú: fully convolutional densenets for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp 11–19
 30. Khodabandehlou H, Pekcan G, Fadali MS (2019) Vibration-based structural condition assessment using convolution neural networks. *Struct Control Health Monit* 26(2):e2308
 31. Kim B, Yuvaraj N, Preetha KS, Pandian RA (2021) Surface crack detection using deep learning with shallow cnn architecture for enhanced computation. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-021-05690-8>
 32. Kornblith S, Shlens J, Le QV (2019) Do better imagenet models transfer better? In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2661–2671
 33. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, pp 1097–1105
 34. Lau SL, Wang X, Xu Y, Chong EK (2020) Automated pavement crack segmentation using fully convolutional u-net with a pre-trained ResNet-34 encoder. Preprint [arXiv:2001.01912](https://arxiv.org/abs/2001.01912)
 35. LeCun Y, Boser BE, Denker JS, Henderson D, Howard RE, Hubbard WE, Jackel LD (1990) Handwritten digit recognition with a back-propagation network. In: *Advances in neural information processing systems*, pp 396–404
 36. Lei B, Wang N, Xu P, Song G (2018) New crack detection method for bridge inspection using UAV incorporating image processing. *J Aerospace Eng* 31(5):04018058
 37. Li S, Zhao, X (2019) Image-based concrete crack detection using convolutional neural network and exhaustive search technique. In: *Advances in Civil Engineering*, vol 2019. <https://doi.org/10.1155/2019/6520620>
 38. Long M, Cao Y, Wang J, Jordan MI (2015) Learning transferable features with deep adaptation networks. In: *Proceedings International Conference on Machine Learning*, vol 37. <https://doi.org/10.5555/3045118.3045130>
 39. Maeda H, Sekimoto Y, Seto T, Kashiwayama T, Omata H (2018) Road damage detection and classification using deep neural networks with smartphone images. *Comput Aided Civil Infrastruct Eng* 33:1127–1141. <https://doi.org/10.1111/micc.12387>
 40. Middleton CR, Lea F (2002) Reliability of visual inspection of highway bridges. Department of Engineering, University of Cambridge, Cambridge, UK
 41. Molnar C (2019) Interpretable machine learning. A guide for making black box models explainable. lulu.com. <https://christophm.github.io/interpretable-ml-book/>
 42. Mundt M, Majumder S, Murali S, Panetsos P, Ramesh V (2019) Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp 11196–11205
 43. Neumann M, Pinto AS, Zhai X, Houlsby N (2019) In-domain representation learning for remote sensing. Preprint [arXiv:1911.06721](https://arxiv.org/abs/1911.06721)
 44. Omar T, Nehdi ML (2017) Remote sensing of concrete bridge decks using unmanned aerial vehicle infrared thermography. *Autom Construct* 83:360–371
 45. Pan SJ, Yang Q (2010) A survey on transfer learning. *IEEE Trans Knowl Data Eng* 22(10):1345–1359
 46. Phares BM, Washer GA, Rolander DD, Graybeal BA, Moore M (2004) Routine highway bridge inspection condition documentation accuracy and reliability. *J Bridge Eng* 9(4):403–413

47. Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. Preprint [arXiv:1511.06434](https://arxiv.org/abs/1511.06434)
48. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. *J Mach Learn Res* 21(140):1–67
49. Raghu M, Zhang C, Kleinberg J, Bengio S (2019) Transfusion: understanding transfer learning for medical imaging. In: *Advances in Neural Information Processing Systems*, pp 3342–3352
50. Railway Accident Investigation Unit (2010) Malahide viaduct collapse on the Dublin to belfast line, on the 21st August 2009. Irish Railways
51. Raphaël (2019) Interpretability Methods for tf.keras models with Tensorflow 2.x. <https://github.com/sicara/tf-explain>
52. Rateke T, Von Wangenheim A (2021) Road surface detection and differentiation considering surface damages. *Autonom Robots* 45(2):299–312
53. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. *Adv Neural Inform Process Syst* 28:91–99
54. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE international conference on computer vision*, pp 618–626
55. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: *Proceedings of International Conference on Learning Representations (ICLR)*. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
56. Stoński M (2019) A comparison of deep convolutional neural networks for image-based detection of concrete surface cracks. *Comput Assis Methods Eng Sci* 26(2):105–112
57. Su C, Wang W (2020) Concrete cracks detection using convolutional neuralnetwork based on transfer learning. *Math Probl Eng*
58. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2014) Going deeper with convolutions. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 1–9. <https://doi.org/10.1109/CVPR.2015.7298594>
59. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2015) Rethinking the inception architecture for computer vision. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp 2818–2826. <https://doi.org/10.1109/CVPR.2016.308>
60. Tamkin A, Singh T, Giovanardi D, Goodman N (2020) Investigating transferability in pretrained language models. Preprint [arXiv:2004.14975](https://arxiv.org/abs/2004.14975)
61. Teng S, Liu Z, Chen G, Cheng L (2021) Concrete crack detection based on well-known feature extractor model and the yolo_v2 network. *Appl Sci* 11(2):813
62. Torrey L, Shavlik J (2010) Transfer learning. In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pp 242–264. IGI Global
63. US Department of Transportation (2012) I-35 Bridge Collapse, Minneapolis. <https://www.fhwa.dot.gov/pressroom/fsi35.cfm>. Accessed 21 Apr 2020
64. Valikhani A, Jaber Jahromi A, Pouyanfar S, Mantawy IM, Azizinamini A (2021) Machine learning and image processing approaches for estimating concrete surface roughness using basic cameras. *Comput Aided Civ Infrastruct Eng* 36(2):213–226
65. Wang ZJ, Turko R, Shaikh O, Park H, Das N, Hohman F, Kahng M, Chau DH (2021) CNN explainer: learning convolutional neural networks with interactive visualization. *IEEE Trans Visual Comput Graphics* 27(2):1396–1406. <https://doi.org/10.1109/TVCG.2020.3030418>
66. Wen L, Li X, Gao L (2020) A transfer convolutional neural network for fault diagnosis based on ResNet-50. *Neural Comput Appl* 32:6111–6124. <https://doi.org/10.1007/s00521-019-04097-w>
67. Xu H, Su X, Wang Y, Cai H, Cui K, Chen X (2019) Automatic bridge crack detection using a convolutional neural network. *Appl Sci* 9(14):2867
68. Yang L, Li B, Li W, Liu Z, Yang G, Xiao J (2017) A robotic system towards concrete structure spalling and crack database. In: *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp 1276–1281. <https://doi.org/10.1109/ROBIO.2017.8324593>
69. Yang S, Huang Y (2021) Damage identification method of prestressed concrete beam bridge based on convolutional neural network. *Neural Comput Appl* 33:535–545. <https://doi.org/10.1007/s00521-020-05052-w>
70. Ye X, Jin T, Yun C (2019) A review on deep learning-based structural health monitoring of civil infrastructures. *Smart Struct Syst* 24(5):567–585
71. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'14)*. MIT Press, Cambridge, MA, USA, pp 3320–3328
72. Zhang C, Chang C-C, Jamshidi M (2018) Bridge damage detection using a single-stage detector and field inspection images. Preprint [arXiv:1812.10590](https://arxiv.org/abs/1812.10590)
73. Zhang C, Chang C-C, Jamshidi M (2020) Concrete bridge surface damage detection using a single-stage detector. *Comput Aided Civ Infrastruct Eng* 35(4):389–409
74. Zhang G, Liang G, Su F, Qu F, Wang JY (2018) Cross-domain attribute representation based on convolutional neural network. In: *Intelligent Computing Methodologies. ICIC 2018. Lecture Notes in Computer Science*, vol 10956. Springer, Cham. https://doi.org/10.1007/978-3-319-95957-3_15
75. Zhang X, Rajan D, Story B (2019) Concrete crack detection using context-aware deep semantic segmentation network. *Comput Aided Civ Infrastruct Eng* 34(11):951–971
76. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 2921–2929. <https://doi.org/10.1109/CVPR.2016.319>
77. Zhou F, Liu G, Xu F, Deng H (2019) A generic automated surface defect detection based on a bilinear model. *Appl Sci* 9(15):3159
78. Zhu J, Zhang C, Qi H, Lu Z (2020) Vision-based defects detection for bridges using transfer learning and convolutional neural networks. *Struct Infrastruct Eng* 16(7):1037–1049

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