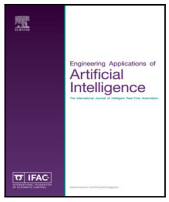
Engineering Applications of Artificial Intelligence 137 (2024) 109106

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

dacl1k: Real-world bridge damage dataset putting open-source data to the test

Johannes Flotzinger [a,*], Philipp J. Rösch [b], Norbert Oswald [b], Thomas Braml [a]

[a] Institute for Structural Engineering, University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, Neubiberg, 85577, Bavaria, Germany
[b] Institute for Distributed Intelligent Systems, University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, Neubiberg, 85577, Bavaria, Germany

ARTICLE INFO

Keywords:
Building inspection
Damage recognition
Computer vision

ABSTRACT

Recognising reinforced concrete defects (RCDs) is a crucial element for determining the structural integrity, traffic safety and durability of bridges. However, most of the existing datasets in the RCD domain are derived from a small number of bridges acquired in specific camera poses, lighting conditions and with fixed hardware. These limitations question the usability of models trained on such open-source data in real-world scenarios.

We address this problem by testing such models on our "dacl1k" dataset, a highly diverse RCD dataset for multi-label classification based on building inspections including 1,474 images. Thereby, we trained the models on different combinations of open-source data (meta datasets) which were subsequently evaluated both extrinsically and intrinsically. During extrinsic evaluation, we report metrics on dacl1k and the meta datasets. The performance analysis on dacl1k shows practical usability of the meta data, where the best model shows an Exact Match Ratio of 32%. Additionally, we conduct an intrinsic evaluation by clustering the bottleneck features of the best model derived from the extrinsic evaluation in order to find out, if the model has learned distinguishing datasets or the classes (RCDs) which is the aspired goal. The dacl1k dataset and our trained models will be made publicly available, enabling researchers and practitioners to put their models to the real-world test.

1. Introduction

Against the backdrop of an ageing structure stock as well as the steady increase in heavy traffic, regular and high-quality bridge inspections are indispensable. Simultaneously, affected countries hold to inspection processes that are out-of-date while being confronted with staff shortages. The final goal of building inspections is the building assessment included in the inspection report. Within this document the damage-informations and -valuations as well as consequential actions (e.g. restoration works, traffic load limitations or bridge closures) are recorded. The recommended actions are determined by the damage-valuation which is based on the damage-information, in addition to the inspector's expertise. Thus, the damage-information is the decisive element. Thereby, each visually and acoustically (Hollowareas) recognisable defect is classified, measured and localised. However, in accomplishing this task, the use of computer vision approaches for acquiring the defect-information, within the framework of digitised inspections (DIs), offers great potential for improvement in terms of cost-effectiveness and quality control.

Major contributions in the field of damage recognition on built structures were made with the advent of datasets for the task of binary (Dorafshan et al., 2018; Hüthwohl and Brilakis, 2018; Xu et al., 2019; Li and Zhao, 2019), multi-class (Hüthwohl et al., 2019; Bianchi and Hebdon, 2021), multi-label classification (Mundt et al., 2019) as well as object detection (Mundt et al., 2019; DANG et al., 2021) and semantic segmentation (Benz and Rodehorst, 2022; Benz et al., 2019; Kulkarni et al., 2023; Flotzinger et al., 2024; Fujishima et al., 2023). Kulkarni et al. (2023) combined multiple image datasets of cracked and uncracked surfaces which is, to the best of our knowledge, the only work intersecting with the RCD domain, that is making use of a dataset compilation.

In general, the research area of reinforced concrete damage (RCD) recognition still faces the problem that only few datasets with mostly binary classifications tasks exist. The datasets are often small in terms of size and class variety. Moreover, the images are taken under restricted laboratory conditions. They use only one camera with fixed focal length and a specific acquisition setup concerning the relative pose of camera and objects as well as lighting conditions. Real-world data, in contrast, is strongly heterogeneous because of the big variety of building types, environmental conditions and image qualities depending on the hardware and the inspector. This opens up the questions: how do models

* Corresponding author.
E-mail address: johannes.flotzinger@unibw.de (J. Flotzinger).

https://doi.org/10.1016/j.engappai.2024.109106
Received 22 January 2024; Received in revised form 6 June 2024; Accepted 2 August 2024
Available online 16 August 2024

Fig. 1. Example images from the four analysed open-source datasets and our dacl1k dataset. Firstly, we train models based on results of previous research (Baseline training) and then apply five improvements to the training process (Improved training). Secondly, models are evaluated on datasets (Extrinsic) and the best model is further analysed (Intrinsic).

trained on existing RCD datasets perform on real-world data? How can existing open-source knowledge be exploited best and which existing datasets are useful?

This work introduces the dacl1k (damage classification) dataset, the first RCD multi-label dataset that originates from real building inspections including 1474 RCD images which were labelled by civil engineers inspired by German inspection standards. In order to train baseline models for dacl1k dataset, we conduct a proven transfer learning strategy, also called baseline training, and an improved training setting. For the improved setting, we examine five steps to leverage performance, choosing a different transfer learning approach, optimising the data augmentation, raising the training resolution, optimising the validation resolution and applying multi-label oversampling. Models resulting from both training strategies are evaluated on dacl1k. Hereby, we examine which open-source dataset combinations (we call them *meta* datasets) are useful for our real-world defect images. The model showing best performance during extrinsic evaluation is subsequently evaluated intrinsically through a bottleneck feature analysis to further investigate the cause of its performance. Thereby, we use dimensionality reduction techniques to identify, if the trained image representations are clustered according to data labels or data source.

To summarise, we make the following contributions: (*i*) dacl1k, the first RCD dataset for multi-label classification originating from real-world inspections, (*ii*) models that result from computational expensive hyperparameter search and several improvements, (*iii*) an extrinsic and intrinsic evaluation of the baselines. The dacl1k dataset, the meta datasets and the baselines will be made publicly available (see https://github.com/phiyodr/building-inspection-toolkit).

2. Related work

Dataset compilations. The Scene UNderstanding (SUN) database (Xiao et al., 2010) was generated by collecting images retrieved from multiple search engines that were proposed after seeking for terms that describe scenes, places, and environments. In the field of few-shot classification ten popular heterogeneous datasets, such as ImageNet (Russakovsky et al., 2015) and MSCOCO (Lin et al., 2014) were combined to one meta dataset while two of them were used for validation (Triantafillou et al., 2020). Others combined cross-domain and in-domain data to generate meta datasets for unlabelled, weakly-labelled or sparsely labelled target datasets (Ullah et al., 2022; Xue et al., 2023). To the best of our knowledge, Kulkarni et al. (2023) is the only work that combines previously available datasets, inter alia, from the RCD domain. They compile a semantic segmentation dataset, called CrackSeg9k, with 9255 images of cracks from ten sub datasets on various surfaces. Before unifying the datasets, their individual problems (e.g. noise and distortion) are addressed by applying Image Processing. SDNET (Dorafshan et al., 2018), which is used in the underlying work, is also part of CrackSeg9k for which no Image Processing was utilised.

Transfer learning. Bukhsh et al. (2021) analysed combinations of six RCD (binary and multi-label) datasets from which four are used in our analysis (see Section 3). They aimed to find the most valuable transfer learning dataset for each RCD dataset using a model initialised with and without weights from ImageNet.

The best performance for all datasets was obtained when using models initialised with ImageNet weights. On CODEBRIM (Mundt et al., 2019) the best performance was achieved when only weights from ImageNet were used, but no other RCD dataset. Thus, no additional training with an in-domain dataset was beneficial. MCDS (Hüthwohl et al., 2019), in contrast, benefited from training on the CODEBRIM dataset. The performance on each binary crack dataset, including SDNET (Dorafshan et al., 2018) and BCD (Xu et al., 2019), benefited from training with a different crack dataset.

Improvements. There are several levers that can be applied to enhance model performance, e.g. increasing or decreasing model capacity, regularising features or improving optimisation. Many improvements regarding model capacity for datasets in the RCD domain were examined in previous work. In Flotzinger et al. (2022) an extensive hyperparameter tuning for several state-of-the-art CNNs and transfer learning strategies were analysed. E.g., they examined different constellations of hidden layers in the classifier, optimiser types, learning rates and learning rate schedulers. Yet, only basic image augmentation was used. Unlike these approaches, we focus on the problem of regularisation by utilising basic image transformations, such as random horizontal and vertical flip, Random Erasing (Zhong et al., 2017) as well as automatic augmentation methods combined with a multi-label oversampling approach. There are several advanced augmentation algorithms (Random Erasing (Zhong et al., 2017), AugMix (Hendrycks

et al., 2020), AutoAugment (Cubuk et al., 2019), RandAugment (Cubuk et al., 2020), TrivialAugment (Müller and Hutter, 2021)) available, which use a large set of image mutations and help to improve classification performance. Most of these methods provide a set of augmentations which are applied at a number of so-called "strength bins". In addition, the range of their augmentation strengths can be defined. The final augmentation, Random Erasing, selects an arbitrary image region and overwrites its pixels with random values.

Furthermore, current work (Mundt et al., 2019) suggests to increase the train crop size in the RCD domain. Recent work demonstrated that directly integrating multiple resolutions inside the network at train and test time raises performance, especially in category-level detection (Lin et al., 2017).

Increasing evaluation resolution compared to training resolution improved performance in previous work (Touvron et al., 2019). This may be beneficial for our models, although, we do not expect a notable disparity in the size of objects observed by the network between train and test phase since our image transformation, in both phases, includes no random resizing and cropping but resizing and centre-cropping.

3. Datasets

For our experiments we use four open-source RCD datasets. We call each of their combination "meta dataset" from which three versions exist. Moreover, we introduce our real-world dataset dacl1k. This dataset is used for the evaluation of models trained on the meta data with respect to practical use. Example images are shown in Fig. 1.

Open-source datasets. We use four open-source datasets which are relevant to us. There are two binary crack datasets, **BCD** (Xu et al., 2019) and **SDNET** (Dorafshan et al., 2018). While BCD targets bridge cracks, SDNET includes images of cracks on walls, decks and pavement. Both are highly standardised datasets regarding hardware, object distance and camera angle which is orthogonal to the reference plane. Since SDNET has many incorrectly labelled data, we used a cleaned version (Rösch and Flotzinger, 2022).

The largest and most realistic dataset in terms of damage appearance is **CODEBRIM** (Mundt et al., 2019). The images were taken in a less standardised setting in comparison to BCD and SDNET. CODEBRIM and dacl1k share the same damage classes, which are very relevant for real-world inspections. There are two issues considering the practical transferability of this dataset to real-world scenarios. Firstly, CODEBRIM is made up from image crops, thus, single images are split into rectangular patches depending on the maximum size of the defects. In addition, undamaged surface is extracted to act as background (*No Damage*), leading to atypical image shapes. Long patches are often associated with cracks (see most left CODEBRIM image in Fig. 1) and second most frequently with long shaped exposed reinforcement bars. Secondly, due to this cropping approach, some resulting patches become very small. The minimum image height and width is 22 and 40 pixels respectively.

As the second dataset including multiple classes, we use an updated version (Rösch and Flotzinger, 2022) of **MCDS** (Hüthwohl et al., 2019). From originally eight classes, the labels "scaling" and "spalling" are merged, since they show the same defect type and only differ with regard to the cause of the damage. Moreover, the class "general" is removed which summarises graffiti and vegetation. This is done due to the fact that "general" neither represents severe damage nor can the non-existence of this class in the other datasets be guaranteed. Consequently, the other multi-class datasets would have to be screened for general defects and labelled to avert false labels in the meta dataset compilations.

dacl1k. We release a novel multi-label classification dataset called dacl1k. This dataset focuses on real-world inspection images in the RCD domain. While the heterogeneity of the dataset presents challenges for model training, it ensures that successful models have practical value in real-world scenarios. Our dataset includes five damage classes *Crack*, *Efflorescence*, *Spalling*, *Bars Exposed*, and *Rust* and the label *No Damage*. These classes are inherited from CODEBRIM with small changes to the original nomenclature. CODEBRIM is the most similar open-source dataset to dacl1k. In contrast to CODEBRIM and MCDS, we supply uncropped images. Examples of dacl1k images are displayed in Fig. 1 where in the first two rows of the image tiles the following defects are shown: *Cracks* with *Efflorescence* (top-left), *Crack* (top-right), *Spalling* with *Bars Exposed* and *Rust* (bottom-left) and *No Damage* (bottom-right). The dataset comprises a total of 1474 images, each with a unique set of challenges including variations in camera types, poses, lighting conditions, and resolutions. The total number of labels amounts to 2367. Our images derive from real inspections and were sourced from databases at authorities and engineering offices. We partitioned the dataset based on an equal label distribution into three subsets, with 67% for training, 13% for validation, and 20% for testing (see Fig. 2). Thus, dacl1k not only allows for testing but also training which is necessary due to lack of performance as described in Section 4.

Meta datasets. Our meta datasets are collections of aforementioned open-source data. The last three columns of Table 1 indicate in which meta dataset the according open-source dataset is included. In order to build the meta datasets, all datasets are transformed to a six-class dataset according to dacl1k. The datasets are sequentially merged with the aim of assessing the impact of each additional data batch. We start with CODEBRIM because it is the most realistic and currently largest multi-label RCD dataset. Subsequently, MCDS is added (*meta2*) since it has the same classes, but being smaller in size. Afterwards, we append BCD (*meta3*) to increase the amount of crack images and healthy surface slightly. Finally, we add SDNET to create *meta4* which is the meta dataset with the strongest class imbalance due to the high amount of binary crack data.

4. Models

This section provides an overview of the trainings conducted during the development of the baselines for dacl1k. First, we examine a proven transfer learning approach named baseline training (Flotzinger et al., 2022). Second, we apply improvement steps to the training pipeline to further leverage performance.

Baseline training. The settings of the default training are derived from previous work in RCD domain (Flotzinger et al., 2022). Here, three different CNN architectures and three transfer learning strategies were compared. Moreover, a computationally expensive hyperparameter search was conducted. In our baseline training we use their best MobileNetV3-Large (Howard et al., 2019) model. MobileNet represents the best trade-off between parameter count and performance for the underlying domain according to results from (Flotzinger et al., 2022). Furthermore, their best transfer learning strategy called "head then all" (HTA) is applied. Here, in the first step the model base is frozen and only the classification head is trained. In the second step, all parameters can be updated. They applied basic image augmentation including resizing, cropping, random rotating and flipping. The training and validation crop size is 224×224. The models are initialised with weights from ImageNet (Deng et al., 2009). Compared to Flotzinger et al. (2022), we adjust the learning rate after a grid search while evaluating on the meta4 dataset.

Improved training. To further improve model performance five additional setups are evaluated. Thereby, we focus on leveraging the performance of models fine-tuned on the most promising datasets, meta2+dacl1k and meta3+dacl1k as well as dacl1k itself. In order to improve the performance, we decided to examine another transfer learning approach applying different learning rates for model head and base (DHB) (Flotzinger et al., 2022; Howard and Ruder, 2018).

Furthermore, in initial experiments, we examined five different automatic augmentation methods: AugMix (Hendrycks et al., 2020), AutoAugment (Cubuk et al., 2019), RandAugment (Cubuk et al., 2020), TrivialAugment (Müller and Hutter, 2021) (TA) and TA with a custom

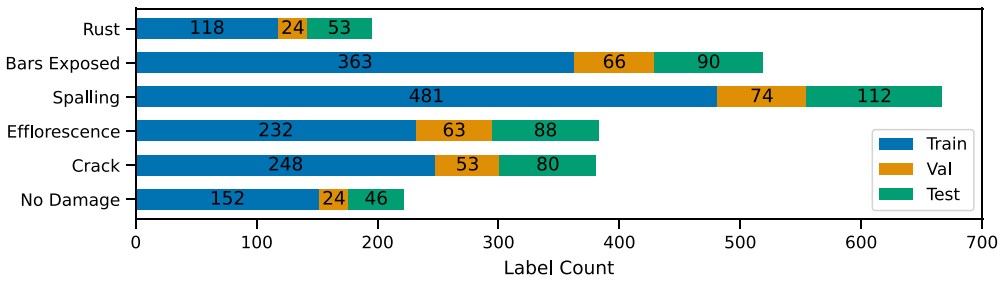


Fig. 2. Label distribution according to train, validation and test split in dacl1k.

Table 1
Dataset statistics of the four open-source datasets as well as dacl1k, displaying the number of classes, samples, image size (height × width) and their meta affiliation.

Dataset	Class	Samples	Image size (min, median, max)	meta2	meta3	meta4
CODEBRIM	6	7729	(22, 359, 3638) × (40, 826, 5997)	✓	✓	✓
MCDS	8	2597	(24, 356, 2830) × (47, 692, 4585)	✓	✓	✓
BCD	2	6069	(224, 224, 224) × (224, 224, 224)	–	✓	✓
SDNET	2	55 449	(256, 256, 256) × (256, 256, 256)	–	–	✓
dacl1k	6	1474	(245, 1024, 5152) × (336, 1365, 6000)	–	–	–

augmentation policy. The custom augmentation policy originates from our presumption that not all augmentations and their ranges inherited from the TA wide-custom augmentation space are beneficial to our data. Hence, we selected representative images from the dataset and evaluated each augmentation on the selection of data manually, based on subjective visual criteria. The custom policy neglects solarisation, posterisation and colourisation because they are considered as non-beneficial. Also, the minimum value of contrast augmentation is raised to 0.5 because the original range results in greying out the image completely. Before applying the augmentation method, random horizontal flip is applied and after the automatic augmentation method, Random Erasing (Zhong et al., 2017) is used. Supplementary, we execute a grid search for the ideal amount of magnitude bins in TrivAug. The search space is uniformly distributed in the interval 25 and 35. The best validation loss is obtained at a number of 34 magnitude bins.

In addition, training crop size is raised to 512 × 512 in expectation of losing less image information due to resizing. This may be beneficial to dacl1k with a relatively large resolution in comparison to the meta datasets (see Table 1). Another improvement step is the increase of the test resolution. Therefore, we test the models on different test crop sizes, starting from a size of 512 × 512 and subsequently adding 16 pixels until 656.

Finally, we tackle the problem of class imbalance by applying a multi-label oversampling strategy. This is especially useful when fusing multi-class datasets with cracks-only datasets, such as SDNET, to correct the predominance of images showing cracks. The algorithm upsamples minority classes regarding the following steps: (*i*) calculate class counts in the current dataset and then calculate the standard deviation based of the count values, (*ii*) randomly draw an image from the dataset and update counts and standard deviation, (*iii*) if the new standard deviation is reduced, this sample is added to the dataset, if not, another sample is drawn. This is repeated a fixed number of times.

As in the default setting, both learning rates (for head and base) were adjusted based on a grid search.

5. Evaluation

In the following, we examine which dataset combination is the most valuable for practical use. Moreover, the best training settings regarding the improvement steps are analysed. In Section 5.1 the models from the baseline and improved training are evaluated. Section 5.2 includes an intrinsic analysis of the model showing the best performance according to its extrinsic evaluation on dacl1k.

Table 2
EMR on the model's source test set (itself) and on dacl1k with the according classwise Recall on dacl1k. All values in percent.

Trained on	EMR		Recall on dacl1k					
	itself	dacl1k	NoDam.	Crack	Effl.	Spall.	BExp.	Rust
CODEBRIM	70.57	16.89	**73.91**	15.00	34.09	24.44	**9.43**	**38.39**
meta2	70.41	**17.35**	63.04	15.00	37.50	**28.89**	7.55	14.29
meta3	**81.52**	**17.35**	**73.91**	12.50	**40.91**	**28.89**	5.66	16.07
meta4	77.84	16.44	71.74	**21.25**	38.64	27.78	1.89	7.14
dacl1k	23.29	23.29	**65.22**	22.50	43.18	44.44	35.85	70.54
meta2+dacl1k	49.22	**27.85**	63.04	31.25	**53.41**	**61.11**	**41.51**	74.11
meta3+dacl1k	75.22	27.40	60.87	**32.50**	48.86	**61.11**	**41.51**	68.75
meta4+dacl1k	**76.81**	26.94	63.04	31.25	43.18	53.33	35.85	**75.89**

5.1. Extrinsic evaluation

We report the Exact Match Ratio (EMR) and classwise Recall to evaluate the models extrinsically. From a machine learning perspective, the EMR is a challenging metric for multi-label classification problems:

$$EMR = \frac{1}{n}\sum_{i=1}^{n} I(y_i = \hat{y}_i), \tag{1}$$

where n is the number of samples, I is the indicator function, y_i is the ground truth or target value and $\hat{y}_i$ is the predicted label. A correct prediction is only given if all classes of an image are classified correctly. In the underlying work all six damage predictions of a given sample have to match the ground-truth. This metric is provided for the according source datasets (*itself*) and for *dacl1k* in Table 2. The classwise Recall indicates how many defects of the according class are overlooked, which is especially interesting from a civil engineer's perspective. The Recall is also called True Positive Rate or Sensitivity. It is the ratio of the true positives, and the sum of true positives and false negatives.

We analyse results from models trained on CODEBRIM, meta2, meta3, meta4 and the combination with dacl1k datasets. It is important to note that the models are always trained once on one dataset. These datasets – treated as one set in one training step – are listed in the first column of Table 2. There is no incremental training on the sub-datasets of the meta datasets or their combinations with the dacl1k set performed.

Baselines results. The models trained on CODEBRIM and meta datasets show similar results, when evaluated on their own test split,

Table 3
Best improvement setting regarding augmentation (Aug), test crop size (TestCS) and multi-label oversampling (MO). EMR is reported on the test split of the datasets the models were trained on (itself). EMR and classwise Recall is reported for dacl1k test set.

Trained on	Improvements			EMR		Recall on dacl1k					
	Aug	TestCS	MO	itself	dacl1k	NoDam.	Crack	Effl.	Spall.	BExp.	Rust
dacl1k	default	528	–	31.51	31.51	**73.91**	**42.50**	52.27	68.89	56.60	68.75
meta2+dacl1k	custom	560	–	61.14	31.51	65.22	31.25	**54.55**	65.56	56.60	**74.11**
meta3+dacl1k	triv-aug	624	✓	**74.57**	**32.42**	67.39	36.25	50.00	**73.33**	**60.38**	**74.11**

compared to performances reported by others [REFS] ranging from 70.41% to 81.52% EMR (see upper part of Table 2). Weak performance is reported when the networks are evaluated on dacl1k (16.44% to 17.35%). The best results are obtained by meta2 and meta3.

The model trained on dacl1k achieves an EMR of 23.29% (see lower part of Table 2), which is approximately a 6 percent points improvement over meta2 or meta3 . When meta datasets and dacl1k are combined, the performance on dacl1k raises. Models benefit from the additional amount of real-world data. The classwise Recall shows weak performance for models solely fine-tuned on meta data. Especially the classes *Bars Exposed*, *Rust*, and *Crack* indicate that the domain shift between meta and target data is big. In other words, knowledge gathered from open-source data only, is in the current setting properly transferable to the real-world dataset. Like the analysis on EMR, only the combinations of dacl1k and meta data leverages Recalls compared to the dacl1k trained model. Considering all displayed metrics, training on meta2 together with dacl1k is the most promising approach, followed by meta3+dacl1k. Thus, we apply the improvements described in Section 4 to raise the model performance.

Improved results. Table 3 presents the best performances achieved after applying the improvements described in Section 4. All models show better results when being trained at a crop size of 512 compared to 224. Additionally, all models benefit from an increase in test resolution. The best performance was achieved by training on meta3+dacl1k in combination with TrivialAugment, a test crop size of 624 and multi-label oversampling (32.42%). This is about a 1 percent point increase in comparison to training on dacl1k dataset only. However, training on meta2+dacl1k did not lead to a better performance compared to the model trained on dacl1k only. With respect to the best result from the default training (see Table 2), EMR is increased by 4.57%. Regarding the classwise Recall, it can be stated that – apart from *Crack* and *Efflorescence* – good performance is achieved. For comparison, previous work (Flotzinger et al., 2022) reported on the balanced version of CODEBRIM an EMR of 74% and the following classwise Recalls: 95% (*No Damage*), 88% (*Crack*), 76% (*Efflorescence*), 89% (*Spalling*), 93% (*Bars Exposed*), 85% (*Rust*).

5.2. Intrinsic evaluation

Performance on our new dacl1k dataset is weak. Therefore, we want to analyse the capabilities of our best model intrinsically. We want to understand if our model mainly gained information from the image content or the dataset source. In the desired setting the model should be able to differentiate between image content, which are the six labels, and not between image sources.

Approach. Our implementation is as follows. We extract bottleneck features from our best model according to EMR on dacl1k (see Table 3) and a model initialised with ImageNet weights for all five datasets. We randomly keep 330 images per dataset to obtain an evenly distributed number of images per data source. Then, we run a non-linear dimensionality reduction from 960 to 2 dimensions using t-SNE (van der Maaten and Hinton, 2008). Here, we carefully select the t-SNE hyperparameters. We found a perplexity of 20, 5000 optimisation steps, and a learning rate of 200 useful.

Results. In the visualisation of our best model in Fig. 3 we see clear clusters for the *Crack* and *No Damage* class of the BCD dataset. Moreover, a dedicated cluster containing SDNET images only. Here, no clear distinction between the two classes is visible. Right to the centre, there is a mixed cluster for CODEBRIM and MCDS. No clear differentiation between the classes is recognisable. On the far right one cluster for dacl1k datasets exists. Compared to the ImageNet visualisation on top, denser clusters are visible. This shows that the learned features have adapted to the underlying RCD domain. However, the created clustering is still coarse and does not show clearly separated clusters according to damage types, but mainly relative to datasets. To summarise, our best model mainly learned to differentiate between datasets and not between image content.

5.3. Discussion

Our work refers to the portability of features learned by open-source RCD data to our real-world dataset dacl1k. During the development of our baselines, we face domain shifts regarding the underlying datasets. The first and largest shift is present between ImageNet and meta data because of the differences regarding their feature distribution and marginal distribution (6 vs. 1000 classes). The second and smaller shift appears as we combine the four different RCD datasets, which do not share the same marginal distribution by default (binary crack vs. multi-class). A third domain shift arises as we evaluate the models on real-world data while having pre-trained on ImageNet and trained – apart from the meta-dacl1k combinations – on meta data that is limited in terms of diversity (Weiss et al., 2016; Pan and Yang, 2010). The performance displayed by the extrinsic evaluation (see Tables 2 and 3) indicates that models have difficulties to predict the samples in the dacl1k test set. Furthermore, the intrinsic analysis shows that the latent image representation adapts to the RCD domain but does not sufficiently learn features in order to clearly differentiate between damage classes. Another reason for the underlying performance can be shortcuts, or rather decision rules, learned from the source dataset hindering generalisation. These shortcuts can lead to the failure of models, especially, when they are tested on real-world data (Geirhos et al., 2020). Additional experimental results and discussions are presented in Appendix C.

6. Conclusion

In this work, we presented dacl1k, a novel real-world dataset for the multi-label classification of defects occurring on massive bridges. We investigated compilations of open-source datasets and improved the training process significantly over previous work. Selected models were evaluated extrinsically and intrinsically. Results lead to the conclusion that in the RCD domain transferring knowledge from open-source data to real-world data shows weak performance. This also holds after having applied multiple improvement steps. The intrinsic evaluation – of the model showing the highest EMR – underlines that the meta and dacl1k image representation strongly differs from each other. Moreover, achieving a successful domain transfer between ImageNet, meta, and our dacl1k dataset requires further research. The applied improvement steps raised the performance significantly to an Exact Match Ratio of over 32%. Yet, this is still insufficient for practical use in the digitised inspection framework. The classwise Recall shows that, apart from *Crack* and *Efflorescence*, 60 to 74% of the real-world defects are recognised.

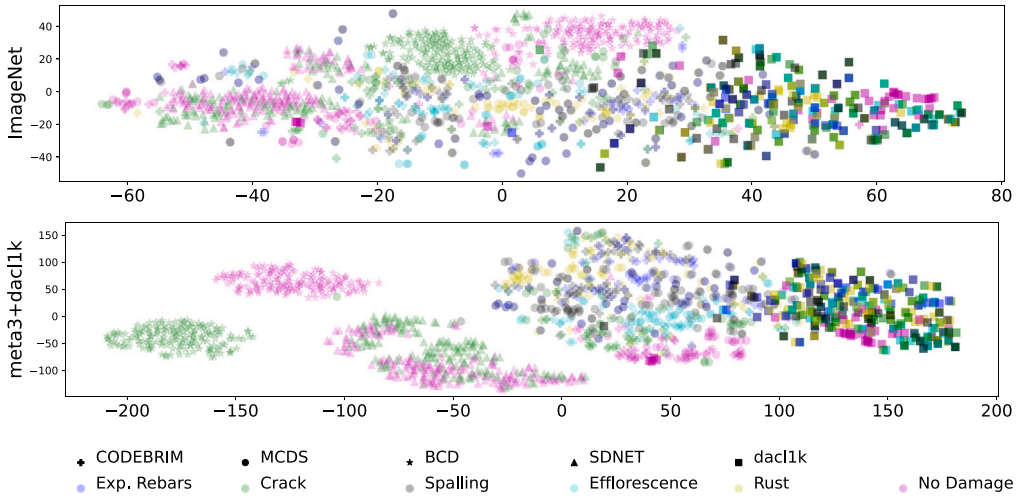


Fig. 3. Clustering of images from all five datasets using bottleneck features after t-SNE dimensionality reduction. Features come from our best model trained on meta3+dacl1k (see Table 3).

Our work can be a starting point to label further real-world data in a semi-automated fashion (Yu et al., 2015). Here, our real-world dataset and corresponding models can be used to pre-label unseen images. In a subsequent step, labels can be assigned automatically when a certain prediction probability has been reached. If this is not achieved, the image must be annotated manually. In addition, dacl1k enables estimating and comparing the usability of models for practical use. This is especially of great value for authorities which are confronted with products offering "damage recognition through AI". Our work acts as a benchmark for evaluating such applications.

CRediT authorship contribution statement

Johannes Flotzinger: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Philipp J. Rösch:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software. **Norbert Oswald:** Supervision. **Thomas Braml:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All code and the dataset is available on GitHub, as mentioned in the paper.

Acknowledgements

We are most grateful to Heiko Neumann and Johannes Kreutz for their insights and suggestions. We thank the Bavarian Ministry of Economic Affairs for funding the MoBaP research project (IUK-1911-0004// IUK639/003) from which this work originates, the engineering offices and authorities for providing data. We gratefully acknowledge the computing time granted by the Institute for Distributed Intelligent Systems and provided on the GPU cluster Monacum One at the University of the Bundeswehr Munich.

Appendix A. Datasets

A.1. dacl1k

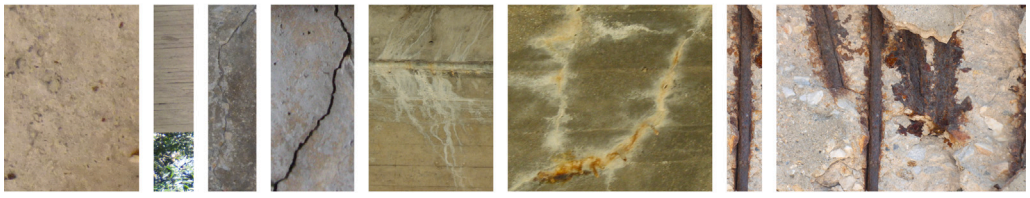
During labelling and quality assessment for dacl1k we followed a two stage annotation process. First, civil engineering students labelled the real-world inspection images after having completed an initial training. Additionally, a detailed class and labelling guideline as well as a sample batch of labelled data was handed out. On the one hand, the class guideline clearly describes each damage class by naming the abbreviation, a detailed description of the visual appearance and the defect cause (see Table A.4). On the other hand, the labelling guideline points out the caveats based on previous labelling processes. Defects often overlap with each other, which often led to false negative labels because only the most obvious defects were recognised. The most common case for overlapping defects is *Spalling* exposing the reinforcement (*Bars Exposed*) which is covered with *Rust*. Another common co-appearance is the combination of *Efflorescence* and *Crack*. The efflorescence stains complicate the detection of the subjacent cracks leading to many un-labelled cracks. Another error source is the presence of heavy weathering on surfaces that also show defects. After each initially submitted batch of data (≈100 images) the labelling team continually received feedback and subsequently repaired their labels.

In a second quality assurance step, we appended samples showing *No Damage* because the original label distribution showed an underrepresentation of healthy concrete surfaces. This is especially important for testing the models with regards to false positives. Furthermore, Fig. A.4 provides an overview of dacl1k's and the meta datasets' images in high resolution. The unified nomenclature from dacl1k and the meta datasets' original label names are displayed in Table A.5.

A.2. Open-source datasets

This section provides details on data acquisition of the open-source datasets. The authors of **BCD** (Xu et al., 2019) used one camera. Based on our manual data validation, the samples in BCD were acquired under constant object distance, camera angle and lighting conditions. The bridge deck samples in **SDNET** (Dorafshan et al., 2018) stem from a number of bridge deck sections that were stored in a laboratory. Images of walls and pavements were taken on a university campus. They used one camera. The surface illumination was between 1500

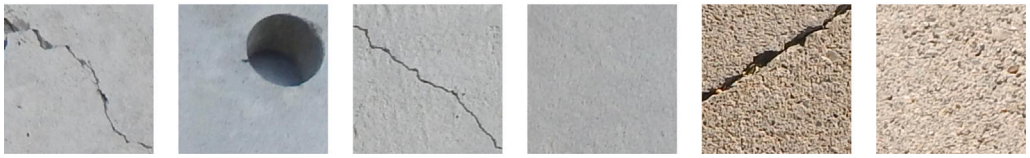
(a) dacl1k. Top row, from left to right: *No Damage*, *No Damage*, *Crack*, *Crack*. Bottom row, from left to right: *Efflorescence* with *Crack*, *Efflorescence* and *Rust*, *Spalling* with *Bars Exposed* and *Rust*, *Rust*.



(b) CODEBRIM. From left to right pairs of: "Background", "Crack", "Efflorescence", "Spallation" with "ExposedBars" and "CorrosionStain".

(c) MCDS. One sample for each damage class. From Left to right: "NoDefect", "Cracks", "Efflorescence", "Scaling", "Spalling", "Exposed Reinfor." without "Rust Staining", "Rust Staining", "General".

(d) BCD. Three samples of the label "Crack" (left) and "NoCrack" (right).



(e) SDNET. Pairwise "Crack" (cracked) and "Uncracked" surfaces of different building parts, from left to right: wall, deck and pavement.

Fig. A.4. Detailed view of all five datasets used in our experiments. The sub-captions include the according dataset's original damage names (see Table A.5).

and 3000 lx. **CODEBRIM** (Mundt et al., 2019) includes images from 30 bridges. They tried to acquire data at varying distance and angles on purpose. In total, four different cameras were used. To homogeneously illuminate the darker bridge areas, they utilised a diffused flash. The authors of **MCDS** (Hüthwohl et al., 2019) aimed for consistent lighting conditions and a perpendicular camera angle to the surface during data acquisition. They used one camera in combination with a set of four lenses.

Appendix B. Training settings

In the following, additional information regarding the training procedure is documented, concentrating on parameters deviating from Flotzinger et al. (2022).

Baseline training. For the activation of the network's output layer, a Sigmoid layer is applied while using Binary Cross Entropy to compute the loss. For the first training step (HO), the learning rate is chosen according to a grid search, while evaluating on meta4 dataset that is the largest meta dataset. The learning rate is varied in the interval $1e^{-2}$ and $1e^{-4}$ which resulted in the best value at $5e^{-4}$. The second training step (HTA), which includes the training of all layers of the model, has a learning rate of $1e^{-5}$. We make use of the Adam optimiser with weight decay (Loshchilov and Hutter, 2017) together with a learning rate scheduler (cosine with warm-up). We trained the models for 100 epochs. The image pre-processing was held constant as follows: Resizing to 1.1*train resolution using bilinear interpolation and centre-crop according to the chosen test crop size. We repeated each training with the same setting two times at different seeds. Finally, the best of the two models with respect to the maximum EMR is reported.

Improved training. In contrast to the baseline training, for the improved training, 50 epochs were considered and the DHB approach was utilised. As in the default setting, both learning rates (for head

Table A.4
Descriptions of dacl1k's damage classes.

Damage	Description
No damage	No damage label describes images that contain healthy concrete surface or irrelevant content.
Crack	Cracks appear when the concrete's tensile strength is exceeded or during hardening, if the post-treatment or the concrete recipe was inadequate.
Efflorescence	Efflorescence is usually whitish, or yellowish. It appears when salts (calcium, sodium, potassium) of the cement stone get dissolved. Cement stone is the hardened cement paste (cement and water) that binds the other concrete components, sand and gravel (aggregate), together. This is a frequently occurring defect emerging when the according building part is constantly in contact with running water.
Spalling	Spalling can appear due to freeze thaw changes, corrosion of the subjacent reinforcement or impact, e.g. from cars that hit the structure.
Rust	Rust appears on metallic objects such as reinforcement and concrete. Rust that is visible on the concrete surface originates from neighbouring metallic parts or the subjacent reinforcement. Reinforcement can corrode as a result of loss of the alkaline protective layer provided by un-carbonated concrete. If the pH value drops due to the carbonation of the concrete, which is unavoidable over time, the reinforcement can oxidise.
BarsExposed	Reinforcement that is visible due to the spalling of the overlying concrete, which is usually the consequence of corrosion of the reinforcement. Another cause for visible reinforcement are rockpockets which arise if the cement paste did not fill all volume between the coarse aggregate. Rock pockets can follow, if the concrete's rheological properties or the compacting of the concrete was inadequate.

Table A.5
Our unified nomenclature in the first column and the original class names in BCD, SDNET, MCDS, and CODEBRIM. General damage class from MCDS is not considered.

meta4/dacl1k	BCD	SDNET	MCDS	CODEBRIM
NoDamage	NoCrack	Uncracked	NoDefect	Background
Crack	Crack	Crack	Cracks	Crack
Efflorescence	∅	∅	Efflorescence	Efflorescence
Spalling	∅	∅	Scaling; Spalling	Spallation
Rust	∅	∅	Rust Staining	CorrosionStain
BarsExposed	∅	∅	Exposed Reinfor.	ExposedBars
∅	∅	∅	General	∅

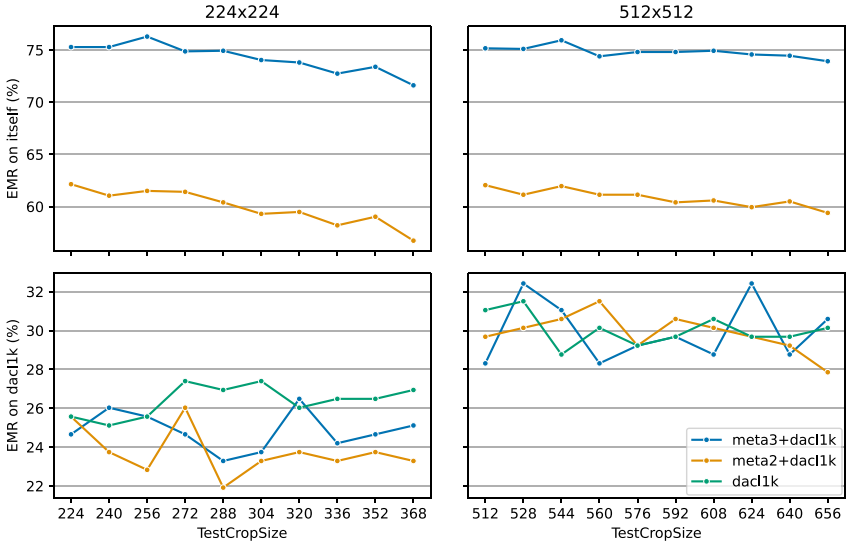


Fig. C.5. EMR at varying test crop size on the according model's test split (itself) and on dacl1k. Three differently trained models are evaluated (dacl1k, meta2+dacl1k and meta3+dacl1k). Each model was trained in compliance with the best training setting (see Table 3 in the main paper for resolution 512 × 512 and Table C.6 for 224 × 224). The charts share the y-axis row-wise and the x-axis column-wise.

and base) were adjusted based on a grid search. This led to a base learning rate of $1e^{-5}$ and a head learning rate of $1e^{-3}$. Regarding the data augmentation, it is important to note that before applying the automatic augmentation method, random horizontal flip is applied with a probability of 50%. After the augmentation method, Random Erasing (Zhong et al., 2017) is used with a probability of 10%. The exact setting of the used TrivAug parameters are shown in Table C.8.

Appendix C. Further extrinsic evaluation

In the following, further extrinsic results are reported. The displayed models were trained on images with a resolution of 224 × 224 (see Table C.6) and 512 × 512 (see Table C.7). We only show results for models trained and tested with the same resolution in the tables. The investigations regarding the test resolution are displayed in Fig. C.5.

Table C.6
Improvement setting regarding augmentation (Aug), and multi-label oversampling (MO) as well as test results on dacl1k (except EMR on itself) for models fine-tuned on three different datasets with a train resolution of 224 × 224. Underlined values represent the best results depending on the training dataset. Bold values represent the best overall result for the according metric.

Trained on	Improvements		EMR		Recall on dacl1k					
	Aug	MO	itself	dacl1k	NoDam.	Crack	Effl.	Spall.	Bexp.	Rust
dacl1k	default	–	25.11	25.11	58.70	27.50	43.18	47.78	43.40	72.32
		✓	24.20	24.20	67.39	28.75	51.14	44.44	60.38	73.21
	triv-aug	–	24.66	24.66	60.87	26.25	38.64	51.11	49.06	68.75
		✓	25.11	25.11	**71.74**	28.75	44.32	43.33	**69.81**	**75.89**
	custom1	–	23.74	23.74	63.04	27.50	43.18	43.33	50.94	70.54
		✓	25.57	25.57	69.57	37.50	50.00	43.33	71.70	67.86
meta2+dacl1k	default	–	62.42	23.29	56.52	31.25	55.68	58.89	50.94	54.46
		✓	63.34	21.92	54.35	21.25	72.73	55.56	49.06	67.86
	triv-aug	–	64.80	25.57	58.70	30.00	56.82	50.00	50.94	63.39
		✓	65.54	24.66	65.22	22.50	52.27	52.22	47.17	66.96
	custom	–	64.71	**26.03**	60.87	25.00	54.55	52.22	49.06	62.50
		✓	64.53	24.20	63.04	32.50	55.68	44.44	50.94	61.61
meta3+dacl1k	default	–	75.75	22.83	45.65	21.25	68.18	45.56	30.19	75.00
		✓	75.64	19.63	32.61	27.50	57.95	50.00	28.30	59.82
	triv-aug	–	77.17	25.11	63.04	22.50	46.59	**60.00**	33.96	63.39
		✓	77.41	23.29	50.00	32.50	**76.14**	36.67	47.17	61.61
	custom1	–	76.64	24.66	58.70	**36.25**	56.82	54.44	39.62	69.64
		✓	**78.06**	22.83	56.52	28.75	63.64	40.00	54.72	61.61

Table C.7
Improvement setting regarding augmentation (Aug), and multi-label oversampling (MO) as well as test results on dacl1k (except EMR on itself) for models fine-tuned on three different datasets with a train resolution of 512 × 512. Underlined values represent the best results depending on the training dataset. Bold values represent the best overall result for the according metric.

Trained on	Improvements		EMR		Recall on dacl1k					
	Aug	MO	itself	dacl1k	NoDam.	Crack	Effl.	Spall.	Bexp.	Rust
dacl1k	default	–	31.05	**31.05**	69.57	40.00	51.14	71.11	56.60	68.75
		✓	27.40	27.40	**71.74**	25.00	43.18	48.89	45.28	78.57
	triv-aug	–	30.14	30.14	69.57	32.50	45.45	52.22	47.17	74.11
		✓	28.77	28.77	**71.74**	35.00	53.41	43.33	49.06	68.75
	custom1	–	23.74	23.74	56.52	32.50	39.77	46.67	60.38	74.11
		✓	28.31	28.31	67.39	**42.50**	50.00	38.89	56.60	65.18
meta2+dacl1k	default	–	61.96	28.31	56.52	38.75	55.68	58.89	54.72	69.64
		✓	62.24	24.20	52.17	35.00	52.27	40.00	50.94	75.00
	triv-aug	–	63.61	28.77	67.39	35.00	51.14	65.56	62.26	70.54
		✓	63.34	29.22	65.22	30.00	**60.23**	55.56	60.38	72.32
	custom	–	63.43	29.68	63.04	33.75	59.09	71.11	60.38	72.32
		✓	63.06	26.48	63.04	28.75	57.95	46.67	58.49	73.21
meta3+dacl1k	default	–	74.51	24.66	54.35	25.00	44.32	64.44	47.17	**79.46**
		✓	75.16	25.57	45.65	31.25	54.55	68.89	**66.04**	67.86
	triv-aug	–	75.93	28.77	58.70	36.25	51.14	**74.44**	52.83	73.21
		✓	75.93	28.31	65.22	30.00	52.27	67.78	60.38	71.43
	custom1	–	76.17	28.31	56.52	32.50	46.59	70.00	49.06	72.32
		✓	**76.40**	24.20	43.48	30.00	**60.23**	63.33	52.83	62.50

Comparing both tables makes clear that training on a resolution of 512 × 512 shows better results on dacl1k compared to training on 224 × 224. This cannot be stated for models that are evaluated on their affiliated test set. Here, the EMR on itself is reduced by approximately 2% for meta+dacl1k and meta3+dacl1k. Comparing the on-itself EMR, the dacl1k model is the only one profiting from the higher train resolution. With respect to the bigger average resolution of samples in dacl1k, this is reasonable. The best EMR on dacl1k is achieved by the model trained on meta2+dacl1k making use of custom data augmentation (26.03%). The best performance of models trained on the 512 × 512 train crop size can be observed for the dacl1k model making use of no improvement step (31.05%).

Fig. C.5 shows the EMR of the models at varying test crop size. We analyse models trained on a resolution of 224 × 224 and 512 × 512, again, trained on dacl1k, meta2+dacl1k and meta3+dacl1k. Regarding the EMR on itself, it can be stated that the performance of the models trained with a train crop size of 224 × 224 is nearly identical to the performance of models trained with 512 × 512. Considering models trained with 512 × 512 images and tested on dacl1k, it can be stated

Table C.8
Our custom augmentation pipeline based on Trivial Augment Wide (Müller and Hutter, 2021) with manipulated range of contrast in bold.

	Augmentation	Range/Probability
Basic augmentation	Random horizontal flip	0.5
Trivial augment	Shear X	0.0–0.99
	Shear Y	0.0–0.99
	Translate X	0–32
	Translate Y	0–32
	Rotate	−135°–+135°
	Brightness	0.01–2.0
	Contrast	**0.5–1.8**
	Sharpness	0.5–1.8
Advanced augmentation	Random erasing	0.1

that the best resolution for meta2+dacl1k is achieved at 560 while for meta3+dacl1k two peaks at a value of 528 and 624 with the same EMR are visible. The model trained on dacl1k shows its best performance at a test crop size of 528.

References

Benz, C., Debus, P., Ha, H.K., Rodehorst, V., 2019. Crack segmentation on UAS-based imagery using transfer learning. In: 2019 International Conference on Image and Vision Computing New Zealand. IVCNZ, pp. 1–6. http://dx.doi.org/10.1109/IVCNZ48456.2019.8960998.

Benz, C., Rodehorst, V., 2022. Image-based detection of structural defects using hierarchical multi-scale attention. In: DAGM German Conference on Pattern Recognition. GCPR, Springer, pp. 337–353.

Bianchi, E., Hebdon, M., 2021. Bearing Condition State Classification Dataset. University Libraries, Virginia Tech, http://dx.doi.org/10.7294/16628698.

Bukhsh, Z.A., Jansen, N., Saeed, A., 2021. Damage detection using in-domain and cross-domain transfer learning. Neural Comput. Appl. 33, http://dx.doi.org/10.1007/s00521-021-06279-x.

Cubuk, E.D., Zoph, B., Mané, D., Vasudevan, V., Le, Q.V., 2019. AutoAugment: Learning augmentation strategies from data. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 113–123. http://dx.doi.org/10.1109/CVPR.2019.00020.

Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V., 2020. Randaugment: Practical automated data augmentation with a reduced search space. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. CVPRW, pp. 3008–3017. http://dx.doi.org/10.1109/CVPRW50498.2020.00359.

DANG, J., MIZUMOTO, T., jo CHUN, P., LIU, J., FUJISHIMA, T., 2021. Multi-type bridge damage detection method based on yolo. Artif. Intell. Data Sci. 2 (J2), 447–456. http://dx.doi.org/10.11532/jsceiii.2.J2_447.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. http://dx.doi.org/10.1109/CVPR.2009.5206848.

Dorafshan, S., Thomas, R.J., Maguire, M., 2018. SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. Data Brief 21, 1664–1668. http://dx.doi.org/10.1016/j.dib.2018.11.015.

Flotzinger, J., Rösch, P.J., Braml, T., 2024. Dacl10k: Benchmark for semantic bridge damage segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. WACV, pp. 8626–8635.

Flotzinger, J., Rösch, P.J., Oswald, N., Braml, T., 2022. Building inspection toolkit: Unified evaluation and strong baselines for bridge damage recognition. In: 2022 IEEE International Conference on Image Processing. ICIP, pp. 1221–1225. http://dx.doi.org/10.1109/ICIP46576.2022.9897743.

Fujishima, T., Dang, J., jo Chun, P., 2023. Training images for semantic segmentation of bridge damage detection. http://dx.doi.org/10.50915/data.jsceiii.24750210.v1.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A., 2020. Shortcut learning in deep neural networks. Nat. Mach. Intell. 2 (11), 665–673. http://dx.doi.org/10.1038/s42256-020-00257-z.

Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B., 2020. AugMix: A simple method to improve robustness and uncertainty under data shift. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=S1gmrxHFvB.

Howard, J., Ruder, S., 2018. Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Melbourne, Australia, pp. 328–339. http://dx.doi.org/10.18653/v1/P18-1031, URL https://aclanthology.org/P18-1031.

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Le, Q., Adam, H., 2019. Searching for MobileNetV3. In: Proceedings of the IEEE International Conference on Computer Vision. 2019-October, pp. 1314–1324. http://dx.doi.org/10.1109/ICCV.2019.00140, arXiv:1905.02244.

Hüthwohl, P., Brilakis, I., 2018. Detecting healthy concrete surfaces. Adv. Eng. Inf. 37, 150–162. http://dx.doi.org/10.1016/j.aei.2018.05.004.

Hüthwohl, P., Lu, R., Brilakis, I., 2019. Multi-classifier for reinforced concrete bridge defects. Autom. Constr. 105, http://dx.doi.org/10.1016/j.autcon.2019.04.019.

Kulkarni, S., Singh, S., Balakrishnan, D., Sharma, S., Devunuri, S., Korlapati, S.C.R., 2023. CrackSeg9k: A collection and benchmark for crack segmentation datasets and frameworks. In: Karlinsky, L., Michaeli, T., Nishino, K. (Eds.), Computer Vision – ECCV 2022 Workshops. Springer Nature Switzerland, Cham, pp. 179–195.

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

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 936–944. http://dx.doi.org/10.1109/CVPR.2017.106.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.), Computer Vision – ECCV 2014. Springer International Publishing, Cham, pp. 740–755.

Loshchilov, I., Hutter, F., 2017. Decoupled weight decay regularization. In: International Conference on Learning Representations.

Müller, S.G., Hutter, F., 2021. TrivialAugment: Tuning-free yet state-of-the-art data augmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. ICCV, pp. 774–782.

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. CVPR.

Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22 (10), 1345–1359. http://dx.doi.org/10.1109/TKDE.2009.191.

Rösch, P.J., Flotzinger, J., 2022. Building inspection toolkit. https://github.com/phiyodr/building-inspection-toolkit.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV) 115 (3), 211–252. http://dx.doi.org/10.1007/s11263-015-0816-y.

Touvron, H., Vedaldi, A., Douze, M., Jégou, H., 2019. Fixing the train-test resolution discrepancy. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. pp. 8250–8260, URL https://proceedings.neurips.cc/paper/2019/hash/d03a857a23b5285736c4d55e0bb067c8-Abstract.html.

Triantafillou, E., Zhu, T., Dumoulin, V., Lamblin, P., Evci, U., Xu, K., Goroshin, R., Gelada, C., Swersky, K., Manzagol, P.-A., Larochelle, H., 2020. Meta-dataset: A dataset of datasets for learning to learn from few examples. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=rkgAGAVKPr.

Ullah, I., Carrión-Ojeda, D., Escalera, S., Guyon, I., Huisman, M., Mohr, F., van Rijn, J.N., Sun, H., Vanschoren, J., Vu, P.A., 2022. Meta-album: Multi-domain meta-dataset for few-shot image classification. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (Eds.), Advances in Neural Information Processing Systems. Vol. 35, Curran Associates, Inc., pp. 3232–3247, URL https://proceedings.neurips.cc/paper_files/paper/2022/file/1585da86b5a3c4fb15520a2b3682051f-Paper-Datasets_and_Benchmarks.pdf.

van der Maaten, L., Hinton, G., 2008. Visualizing data using t-SNE. J. Mach. Learn. Res. 9 (86), 2579–2605, URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Weiss, K., Khoshgoftaar, T.M., Wang, D., 2016. A survey of transfer learning. J. Big Data 3 (1), 1–40. http://dx.doi.org/10.1186/s40537-016-0043-6.

Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A., 2010. SUN database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 3485–3492. http://dx.doi.org/10.1109/CVPR.2010.5539970.

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), http://dx.doi.org/10.3390/app9142867, URL https://www.mdpi.com/2076-3417/9/14/2867.

Xue, Z., Yang, F., Rajaraman, S., Zamzmi, G., Antani, S., 2023. Cross dataset analysis of domain shift in CXR lung region detection. Diagnostics 13 (6), http://dx.doi.org/10.3390/diagnostics13061068, URL https://www.mdpi.com/2075-4418/13/6/1068.

Yu, F., Zhang, Y., Song, S., Seff, A., Xiao, J., 2015. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365.

Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y., 2017. Random erasing data augmentation. arXiv:1708.04896.