# Improving Accuracy of Concrete Crack Detection Using Multi-Source Data in Deep Learning Models

Daniel Einarson
Department of Computer Science
Kristianstad University
Kristianstad, Sweden
daniel.einarson@hkr.se

Dawit Mengistu
Department of Computer Science
Kristianstad University
Kristianstad, Sweden
dawit.mengistu@hkr.se

***Abstract*— Deep learning models based on convolutional neural networks have been successfully applied in several image processing tasks. The level of success, however, is dependent on large number of data, with high quality images, and where datasets are balanced between features of investigations. Techniques to meet circumstances when this is not the case, include data adaption, where external datasets supplement deficient data, in terms of quantity or quality. This paper investigates cases of adaptation of Multi-source Datasets to improve accuracy in hard cases of detecting cracks in concrete of artefacts such as bridges, which is necessary for the predictive maintenance of such artefacts.**

***Keywords—Concrete Crack Detection; Convolutional Neural Networks; Deep Learning; Data Augumentation***

## INTRODUCTION

Deep learning models based on convolutional neural networks (CNN) have been successfully applied in image processing tasks such as face recognition, object identification, quality control and defect detection. Although these models performed remarkably well on many tasks, there are challenges to apply them in certain domains because they are heavily data dependent. A large amount of imaging data is needed to train a robust model, which might often be infeasible.

Image processing with CNN has shown promising results to support the predictive maintenance of artefacts such as bridges, roads, and buildings by identifying cracks. In our earlier work, we studied the use of CNN models to detect cracks in bridge concrete structures. In the study, we investigated the impact of the resolution and quality of the images on the model accuracy. A vast majority of the bridge images are not labelled and were available only in raw formats. Significant preprocessing was desired to use these images which were not shot from close distances due to lack of access or hazardous situations. We used the Mendeley dataset[1] to train the model in our preliminary experiments and evaluate its accuracy on images selected from that target. This publicly available was preprocessed to produce the image characteristics needed for this study. The model accuracy was evaluated using a test dataset prepared from the bridge images. Although the results were encouraging, we observed that it was not possible to achieve higher accuracy. Further investigation showed that this was so because the training and the target test data come from sources having different distributions.

Misprediction in this context means missing out serious damages that would require urgent maintenance or raising false alerts on the contrary. To overcome the above challenges, we expanded our training data by using two major approaches, data augmentation and domain adaptation.

The use of image augmentation to expand the target dataset and generate enough training data has been reported by researchers in other domains. Augmentation is an approach to enhance the quality of images and increase the quantity of training data with the help of a suite of preprocessing methods. The effectiveness of this approach was demonstrated in medical imaging applications [1]. We investigated the adoption of image augmentation by performing a number of transformations on our source images such as resizing and inversion [2]. The improvement achieved using this approach is limited because the increase in the training dataset is not enough to significantly affect the model.

Another approach is to extend the training dataset by incorporating images from related datasets. Studies in other problem domains show that domain adaptation techniques to build a generalized model that can learn from multiple but related data sources can produce better models that can perform reasonably well ([3], [4]). In our study, we used datasets from multiple sources to build a more robust model to identify cracks with better accuracy. The study shall be seen as an experimental basis to provide further knowledge to a domain of the addressed problems.

The rest of the paper is outlined as follows: A brief background of the problem domain is presented followed by introducing multi-source adaptation for image classification. The Methodology followed will present how this study will be approached, as well as outlining the structure of the CNN of use. while Implementation and Results will cover the results of the studies. Finally, the paper will summarize the studies of the contribution, and provide further discussions on future work.

[1] Concrete Crack Images for Classification, Contributor: Çağlar Fırat Özgenel, DOI:10.17632/5y9wdsg2zt.2

## BACKGROUND

### A. *Review of* Data sets

The background of these studies is addressed through [5], and [2], where datasets of images of parts of the Öresund Bridge, between Sweden and Denmark, has been analyzed for cracks in the concrete of that bridge ([5], and [2]), For the CNN to work well on such datasets, there are needs for those datasets to be qualitative enough, balanced, and moreover, large enough.

Image classification based on supervised learning assumes that the training data and the testing data are sampled from the same distributions. Impressive results were obtained using supervised models on popular benchmarks such as ImageNet [6]. However, such models perform poorly when deployed in practical applications due to the inherent difference expressed in terms of domain shift between the training data and the real-world target image. Domain adaptation is used to address this domain shift among the distributions of multiple sources used to train the model. The adaptation takes care of the shift caused by the differences in distributions of the data sources. For this reason, this approach wherein multiple labeled source domains are used to transfer the task knowledge to the unlabeled target domain is gaining active research interest [7]. In this approach, a shared feature extractor along with domain-specific classifier modules are built with the help of the multi-source datasets [3].

In this paper, we report the major contributions of our study which are based on running our experiments on four datasets acquired from different sources.

All datasets consist of images categorized under two groups, with one group containing images with cracks (*Positive*) and the other containing images without cracks (*Negative*). To facilitate development of an accurate model for the initial experiments, the images in the two categories were chosen to mimic an ideal scenario, that is, presence or absence of cracks are easily distinguishable. In the remaining two datasets, we included images that are relatively harder to categorize. We run our tests by building CNN models trained on different combinations of the four datasets. In each test, we held out a group of images from each dataset and excluded them from model training, to use them in testing and evaluation. There were also additional images collected by the authors to be used for additional studying purposes.

The background of the dataset of images from the Öresund Bridge, is the need for maintenance of the bridge, that is headed by the Øresundsbro Konsortiet[2]. That dataset originates from pictures taken with cameras on long distance, and with rather low qualities. Still, the approach of taking those pictures shall be seen as a step from workers hanging from ropes, physically inspecting the concrete of the bridge's fundament, to automatic inspections through drones scanning the bridge. Such automatic scanning methods clearly need a backend system with trained algorithms that efficiently differentiates between occurrences of cracks and their absence in the bridge's concrete structures. Samples of images from the Öresund Bridge dataset is labeled d) of Figure 1.

Starting to experiment with CNNs and images with or without cracks, can be advantageously done with idealized datasets, i.e., containing images with a clear difference between whether they have cracks or not. The contribution from [5] refers to a Mendeley dataset ([8]), with such clarity (sample images labeled a) of Figure 1). Furthermore, [2] refers to yet another Mendeley dataset ([9]), with five different, still distinct, shades of differences in clarity between cracks and non-cracks. This study will focus on images labeled no cracks, and low level of cracks (samples labeled b) of Figure 1).

In addition to the datasets described above, [2] considered a dataset representing images of asphalt ([10]), where it is more difficult to see the differences between cracks and non-cracks (samples labeled c) of Figure 1). These studies see it as essential to look at cases of more difficult image analysis against the background of real-world cases, where it is likely that one is faced with that type of problem.

Furthermore, [2], argues that further improvements of the accuracy of the CNN may be obtained by simply inverting the images, that is, the darker areas will become lighter, while the lighter will become darker. Here, it is especially significant to see that the originally darker cracks may be easier to find by the CNN while inverted. Figure 1, illustrates the datasets a) – d), presented above, in inverted versions. The clear gap of clearness between datasets a) and b), on one hand, and c) and d) on the other hand is also here illustrated.

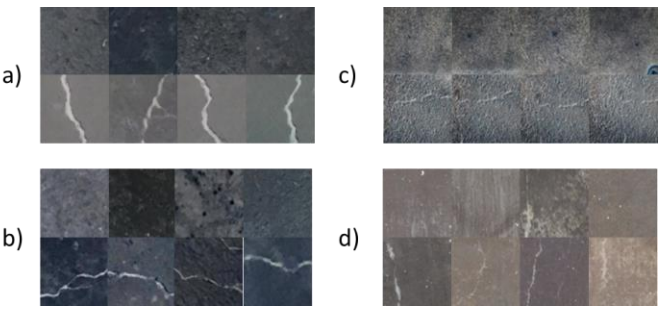


**Figure 1**, Inverted samples from datasets a), b), c), and d) (without cracks above, and with cracks below).

### *B. Review of CNN of Use*

As pointed out by [11] the CNN has undergone an evolution of complexity in structure, from early versions, such as, LeNet-5 (1998), with a structure consisting of a sequence of layers, to e.g., ResNeXt-50 (2017), with several in parallel organized sequences of layers. The studies of [5] and [2] showed that a less advanced AlexNet (2012), sufficiently enough handled the problem of finding cracks in concrete in datasets of images. It was shown in [5] that the resulting accuracy was fully comparable to previous studies using far more complex CNN-structures, and to a much lower cost in time. That AlexNet-based CNN consisted of 6 layers, as follows ([5]):

[2] Øresundsbro Konsortiet - https://www.oresundsbron.com/en/info/company

1. A Convolution layer of size 16 (C-16), an Activation function (A) of type Relu, a MaxPooling (P) of size 5x5, and a DropOut (D) of 20%

2. C-32, A-Relu, P-3x3, D-20%

3. C-64, A-Relu, P-2x2, D-20%

4. A Flatten Layer

5. A Dense Layer of size 32, and A-Relu

6. Dense-1, and a Sigmoid Activation Function finally providing output. ------.

## METHODOLOGY

A contribution of [5] was that downsizing images from (for instance) 256x256 pixels to 64x64 pixels did not have significant effect on the accuracy of the CNN. Moreover, as addressed in previous section, a contribution from [2] is that inverting the images, with respect to grayscale, will further improve the accuracy of the CNN. Aligned with this, the studies of this contribution will approach the problem using inverted images if size 64x64 pixels. Furthermore, to be consistent with the work of [5] and [2], the CNN described in the previous section will be used.

The number of images from each one of the datasets a) to d) are as follows:

a) 10 K + 10 K (cracks + no-cracks)

b) 5 K + 5 K

c) 200 + 200

d) 2 006 with cracks, and 4 633 without cracks.

The core aim of this study is to see how complementing a dataset with similar datasets may improve accuracy in cases of small datasets. For instance, the dataset of c) above may be too small for an efficient enough CNN and may therefore be supplemented with images from other similar datasets. The investigation will initially perform a training and validation phase for different combinations of the datasets a) – d). For each such phase there will be a testing phase where test samples consist of images from each one of the datasets a) to d). in addition to this, private pictures taken by the authors will be used to further investigate the trained CNNs.

Due to the limited number of images of dataset c), and for the sake of keeping the balance between the datasets, the number of images from each dataset that will be used in the training and validation phase will be kept rather small. Furthermore, images will also be needed for a test phase to provide results based on the confusion matrix. Thus, 180 negatives (without cracks) and 180 positives (with cracks) will be used for training and validation, and correspondingly 20 + 20 images will be used for testing.

The combinations for training and validation will be as follows (according to the order of Figure 1):

1. **a)**, **b)**, **c)**, **d)**, that is each one of a) – d) uniquely.

2. **a)+b)**, **a)+c)**, **a)+d)**, **b)+c)**, **b)+d)**, **c)+d)**, that is, combinations of two datasets (such as from a and b)

3. **a)+b)+c)**, **a)+b)+d)**, **a)+c)+d)**, **b)+c)+d)**

4. **a)+b)+c)+d)**

Moreover, for each of the training and validation phases (that is, the total amount of such phases = 15), image sets from each of a) – d), will be tested, which results in 60 different tests (15 * 4). For each one of those, the true positives (**TP**, True Positives, i.e., correctly detected cracks) and true negatives (**TN**, True Negatives, i.e., correctly detected no-cracks) will be calculated, as well as the test accuracy (**TACC**, i.e., number of **TP** + **TN** / number of positives, **P** + negatives, **N**), and sensitivity (**TPR**, i.e., num **TP** / num **P**).

## RESULTS

Initially TABLE I illustrates the results of the training phases with respect to accuracy and validation accuracy. The results of the Confusion Matrices are presented in TABLE II, showing the **TP** vs. **TN** for the different tests, and TABLE III, and IV capturing the test accuracies (**TACC**), and the sensitivities (**TPR**) respectively.

TABLE **I**, ACCURACY (*ACC*) AND VALIDATION ACCURACY (*VAL*) OF THE TRAINSETS.

| ***TrainSets*** | ***ACC*** | ***VAL*** |
|---|---|---|
| ***a*** | ***97,2*** | ***97,2*** |
| ***b*** | ***95,5*** | ***95,8*** |
| ***c*** | ***84,7*** | ***88,9*** |
| ***d*** | ***89,9*** | ***90,3*** |
| ***a_b*** | ***98,1*** | ***96,5*** |
| ***a_c*** | ***91*** | ***86,8*** |
| ***a_d*** | ***94,1*** | ***95,8*** |
| ***b_c*** | ***93,4*** | ***83,3*** |
| ***b_d*** | ***95,3*** | ***95,1*** |
| ***c_d*** | ***91,3*** | ***88,9*** |
| ***a_b_c*** | ***95,3*** | ***89,4*** |
| ***a_b_d*** | ***96,9*** | ***95,4*** |
| ***a_c_d*** | ***92,6*** | ***88,9*** |
| ***b_c_d*** | ***92,4*** | ***89,4*** |
| ***a_b_c_d*** | ***94,3*** | ***94,8*** |

With respect to the rather low datasets for training (360 for a) – d) each, and the sum of those for the combinations) the levels of the accuracies are fairly high. Still, the training phases were executed during 50 epochs, implying certain risks for problems with overfitting. Correspondingly, it is of importance to perform tests and check out the values of the confusion matrix. The absence of false positives (*FP*), and false negatives (*FN*), can be compensated and calculated as $FP = 20 - TN$, and $FN = 20 - TP$ (i.e., *20* is the total number of positives and negatives respectively in the test).

**TABLE II**, ACCURACY (ACC) AND VALIDATION ACCURACY (VAL) OF THE TRAINSETS

| | ***TP** for Tests TestSets* | | | | ***TN** for TestSets* | | | |
|---|---|---|---|---|---|---|---|---|
| **TrainSets** | **a** | **b** | **c** | **d** | **a** | **b** | **c** | **d** |
| **a** | 20 | 10 | 20 | 17 | 20 | 20 | 20 | 19 |
| **b** | 20 | 14 | 8 | 1 | 20 | 20 | 20 | 20 |
| **c** | 19 | 15 | 16 | 2 | 20 | 20 | 20 | 20 |
| **d** | 20 | 19 | 20 | 13 | 16 | 13 | 17 | 20 |
| **a_b** | 20 | 19 | 20 | 16 | 20 | 20 | 20 | 20 |
| **a_c** | 20 | 18 | 28 | 8 | 20 | 20 | 20 | 20 |
| **a_d** | 20 | 15 | 20 | 14 | 18 | 20 | 19 | 20 |
| **b_c** | 20 | 17 | 18 | 8 | 18 | 20 | 20 | 20 |
| **b_d** | 20 | 19 | 19 | 13 | 18 | 20 | 20 | 20 |
| **c_d** | 20 | 19 | 19 | 15 | 18 | 20 | 20 | 20 |
| **a_b_c** | 20 | 17 | 18 | 17 | 19 | 20 | 20 | 20 |
| **a_b_d** | 20 | 20 | 19 | 18 | 19 | 20 | 19 | 20 |
| **a_c_d** | 20 | 20 | 18 | 15 | 19 | 20 | 20 | 20 |
| **b_c_d** | 20 | 20 | 18 | 11 | 19 | 20 | 19 | 20 |
| **a_b_c_d** | 20 | 19 | 18 | 13 | 19 | 20 | 19 | 20 |

**TABLE III**, TEST ACCURACIES (**TACC**) FOR EACH TESTSET

| | ***TACC** for TestSets* | | | |
|---|---|---|---|---|
| **TrainSets** | **a** | **b** | **c** | **d** |
| **a** | 100 | 75 | 100 | 90 |
| **b** | 100 | 85 | 70 | 52,5 |
| **c** | 97,5 | 87,5 | 90 | 55 |
| **d** | 90 | 80 | 92,5 | 82,5 |
| **a_b** | 100 | 97,5 | 100 | 90 |
| **a_c** | 100 | 95 | 120 | 70 |
| **a_d** | 95 | 87,5 | 97,5 | 85 |
| **b_c** | 95 | 92,5 | 95 | 70 |
| **b_d** | 95 | 97,5 | 97,5 | 82,5 |
| **c_d** | 95 | 97,5 | 97,5 | 87,5 |
| **a_b_c** | 97,5 | 92,5 | 95 | 92,5 |
| **a_b_d** | 97,5 | 100 | 95 | 95 |
| **a_c_d** | 97,5 | 100 | 95 | 87,5 |
| **b_c_d** | 97,5 | 100 | 92,5 | 77,5 |
| **a_b_c_d** | 97,5 | 97,5 | 92,5 | 82,5 |

**TABLE IV**, SENSITIVITY(**TPR**) FOR EACH TESTSET

| | ***TPR** for TestSets* | | | |
|---|---|---|---|---|
| **TrainSets** | **a** | **b** | **c** | **d** |
| **a** | 100 | 50 | 100 | 85 |
| **b** | 100 | 70 | 40 | 5 |
| **c** | 95 | 75 | 80 | 10 |
| **d** | 100 | 95 | 100 | 65 |
| **a_b** | 100 | 95 | 100 | 80 |
| **a_c** | 100 | 90 | 140 | 40 |
| **a_d** | 100 | 75 | 100 | 70 |
| **b_c** | 100 | 85 | 90 | 40 |
| **b_d** | 100 | 95 | 95 | 65 |
| **c_d** | 100 | 95 | 95 | 75 |
| **a_b_c** | 100 | 85 | 90 | 85 |
| **a_b_d** | 100 | 100 | 95 | 90 |
| **a_c_d** | 100 | 100 | 90 | 75 |
| **b_c_d** | 100 | 100 | 90 | 55 |
| **a_b_c_d** | 100 | 95 | 90 | 65 |

In short it can be concluded from TABLE I that test images from dataset:

*a)* fits well with mostly all train cases, also the Öresund Bridge images (*d*), though some negative images are interpreted incorrectly.

*b)* surprisingly implies several false negatives from train set *a)*, and its own train set.

*c)* even though asphalt images, seems to gain very much from supplementary datasets. Initially this set contains the smallest number of images, thus motivating the technique covered in this contribution.

*d)* has very hard finding the positives in most cases, with an exception from train case a)

It should clearly be pointed out that low number of *TP*, i.e., correspondingly high number of *FN*, often is the most critical aspect of a trained network, and thus in special need for observation and control. TABLE III, and TABLE IV do also point out train set *a)* as a fairly good set of images to represent a possible supplementary base set for the Öresund Bridge images. Thus, further investigations based on that fact is motivated.

A further experiment has mapped the test set of *d)* to a train set of 1000 + 1000 images of type *a)*, and has given the following results:

1. Training the CNN for that set resulted in an accuracy and a validation accuracy both of about 99%
2. The confusion matrix based on the d) test set resulted in the surprisingly high accuracy level of 100%, that is, the number of *FN* and *FP* is zero.

That result is certainly encouraging but is nevertheless subject to further investigations. Private pictures taken by the authors of this contribution provide a further experimental basis, where those may be customized for purposes of testing the CNN in certain directions. Here sharp representations of negative and positive images fit well to the training sets, while nuances more in between those often are analyzed incorrectly. For instance, non-crack images, but structured in some way (Figure 2 a), may be considered positive, while cracks not sharp enough are not detected (Figure 2 b).

a) b)

**Figure 2**, Inverted samples from private pictures, a) non cracks but with inherent structures, b) weak crack structure.

There are two points of discussions to add to this:

1. The studies in this, and previous, contribution assumes a binary classification. The Mendeley dataset a) is clearly binary, while the nuances of the Öresund Bridge images rather point out a need for reconsidering that assumption.
2. It is critical that positive images shall be detected, that is *FN* shall be zero. Still, sharp cracks will certainly be detected, and those are the necessary ones. Other nuances rather represent cases that may be ignored.

## CONCLUSIONS AND FUTURE WORK

The study of this contribution shall be seen as an experimental basis to provide further knowledge improvements in a problem domain of hard cases of detecting cracks in concrete of artefacts such as buildings, bridges, or roads. The main focus has been on problems where images taken at the surface of concrete are of low quality, low in numbers, and imbalanced (low number of images with cracks in relation to images without). Here, for instance, in a modern society with automatic systems e.g., where drones are used to check for cracks there may initially, there may be no or rather a few, labeled data to use for training. An alternative is to use similar data from other image datasets for training and determine the state of the new images on the basis of those.

This work has provided several examples where CNNs have been trained with some image datasets and tested with images from other datasets. It was shown that such comparisons between images from different datasets generally match fairly well against each other, and therefore bring encouraging results to these studies. Still, cases of mismatches point out the importance of further in-depth investigations of the internal structures of the CNN of one hand, and image adaptations on the other hand, for the sake of even higher precision.

The experimental findings opened additional research inquiries for future work. Relevant questions to ask in this regard include: *How can we build an accurate prediction model on one dataset and use it on images from other sources than it trained on? How can images from one dataset be adapted to fit better relative to another dataset. How can we reduce the effect of dataset imbalance using supplementary data from other sources*?

## REFERENCES


[1] Shorten, Connor, and Taghi M. Khoshgoftaar. "A survey on image data augmentation for deep learning." Journal of big data 6, no. 1 (2019): 1-48.

[2] Einarson, D., Improvements to Deep Learning Approaches for Crack Detection in Bridge Concrete Structures, Submitted to the International Conference on Frontiers in Smart System Technologies (ICFSST – 2023), Veltech University, Chennai, India, 2023.

[3] Karimpour, M., Noori Saray, S., Tahmoresnezhad, J. et al. Multi-source domain adaptation for image classification. Machine Vision and Applications 31, 44 (2020). https://doi.org/10.1007/s00138-020-01093

[4] Shiliang Sun, Honglei Shi, Yuanbin Wu, A survey of multi-source domain adaptation. Information Fusion (Elsevier), Vol. 24, (2015), p84-92 https://doi.org/10.1016/j.inffus.2014.12.003.

[5] Einarson, D., Mengistu, D., Deep Learning Approaches for Crack Detection in Bridge Concrete Structures, Proceedings of the 2022 International Conference on Electronic Systems and Intelligent Computing, ICESIC 2022. Institute of Electrical and Electronics Engineers (IEEE), 2022.

[6] ImageNet, https://www.image-net.org/update-mar-11-2021.php

[7] Geiß C., Rabuske A., Aravena Pelizari P., Bauer S., Taubenböck H., " Selection of unlabeled source domains for domain adaptation in remote sensing", Array, Volume 15, 2022, https://doi.org/10.1016/j.array.2022.100233

[8] Özgenel, Çağlar Fırat (2019), "Concrete Crack Images for Classification", Mendeley Data, V2, doi: 10.17632/5y9wdsg2zt.2

[9] Qi, Zhanfeng (2020), "Concrete cracking level", Mendeley Data, V1, doi: 10.17632/bs7rjwywfm.1

[10] A, Jayanth Balaji; G, Thiru Balaji; M S, Dinesh; Nair, Binoy; D. S, Harish Ram (2019), "Asphalt Crack Dataset", Mendeley Data, V2, doi: 10.17632/xnzhj3x8v4.2

[11] Karim Raimi, Illustrated: 10 CNN Architectures - A compiled visualization of the common convolutional neural networks, towards data science, Jul 29, 2019, available at: https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d