



# CNN-Based Deep Architecture for Health Monitoring of Civil and Industrial Structures using UAVs <sup>†</sup>

Thomas Harweg <sup>1</sup>, Annika Peters <sup>2</sup>, Daniel Bachmann <sup>1</sup> and Frank Weichert <sup>1</sup>\*

<sup>1</sup> Department of Computer Science, TU Dortmund University, 44227 Dortmund, Germany

<sup>2</sup> Faculty of Mechanical Engineering, TU Dortmund University, 44227 Dortmund, Germany

\* Correspondence: frank.weichert@tu-dortmund.de; Tel.: +49-231-755-6122

<sup>†</sup> Presented at the 6th International Electronic Conference on Sensors and Applications, 15–30 November 2019; Available online: <https://ecsa-6.sciforum.net/>

Published: 14 November 2019

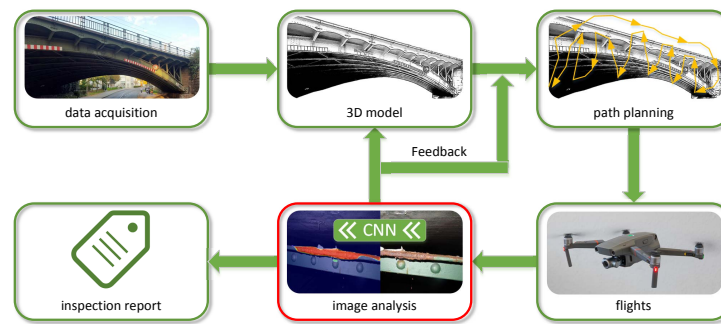
**Abstract:** Health monitoring of civil and industrial structures has been gaining importance since the collapse of the bridge in Genoa (Italy). It is vital for the creation and maintenance of reliable infrastructure. Traditional manual inspections for this task are crucial but time consuming. We present a novel approach for combining Unmanned Aerial Vehicles (UAVs) and artificial intelligence to tackle the above-mentioned challenges. Modern architectures in Convolutional Neural Networks (CNNs) were adapted to the special characteristics of data streams gathered from UAV visual sensors. The approach allows for automated detection and localisation of various damages to steel structures, coatings and fasteners, e.g. cracks or corrosion, under uncertain and real-life environments. The proposed model is based on a multi-stage cascaded classifier to account for the variety of detail level from the optical sensor captured during an UAV flight. This allows for reconciliation of the characteristics of gathered image data and crucial aspects from a steel engineer's point of view. To improve performance of the system and minimise manual data annotation, we use transfer learning based on the well-known COCO data set combined with field inspection images. This approach provides a solid data basis for object localisation and classification with state-of-the-art CNN architectures.

**Keywords:** health monitoring; civil and industrial structures; convolutional neural networks; unmanned aerial vehicles

## 1. Introduction

In 2019 there are 39,671 bridges in road traffic in Germany. In the field of steel- and steel composite bridges this amounts to a total bridge area of 3 999 000 m<sup>2</sup>, 65% of these are much older than 20 years. About 46% of these bridges are in the lower half of the rating scale regarding their status of structure [1]. These figures only concern road bridges. Rail traffic, and accordingly, the rail bridges are not included. There are further 25,677 (12,2018) bridges, 4,899 of them are steel constructions [2]. The *Federal Ministry of Transport and Digital Infrastructure's* traffic forecast for 2030 states, that there is an increase in 43% in rail traffic concerning tonne-kilometres. The passenger service on rails should be up by almost 39% [3].

These figures illustrate the importance of a substantial healthy and reliable infrastructure. For ensuring the continuously increasing road- and rail traffic, there is a regulation for periodic monitoring of bridges [4–6]. *DIN 1076 "Civil Engineering Structures for roads – Monitoring and Examination"* [5] regulates the inspection and control of all bridges concerning road traffic. For rail bridges, Ril 804.80 [4,6] is an analogue.



**Figure 1.** Connections between the different stages in automated bridge inspection

Both rules mandate a building examination every six years. This examination has to be carried out visually, at a very close distance to the structure, allowing tactile feedback [5,6]. DIN 1076 calls this examination *general inspection* (H). Inspection is conducted by qualified specialists. To guarantee complete access for the specialist, technical devices are necessary. The use of these big units make (partial) road closure inevitable and are thus time consuming.

In addition to the described *general inspection*, DIN 1076 requires a *simple inspection* (E) every six years. This should be an extended visual inspection without tools, which is three years shifted to the *general inspection* (H). The defects found in the *general inspection* (H) should be found and compared in the *simple inspection* (E). A change of the existing defects is the focal point of this inspection (E). In those years when there is no *general inspection* (H) nor a *simple inspection* (E), an *observation* (B) is required. Further, a *special investigation* (S) is mandatory by order, or after general storms, fires or accidents. These considerations demonstrate that currently a bridge inspection in Germany is tied to large expenses in personnel, equipment, and money.

Based on these challenges, the project *InÜDos*<sup>1</sup> was initiated to automate monitoring, inspection and documentation of steel constructions. Figure 1 shows the interconnection of the different stages in our automatic monitoring approach. The scope of the project *InÜDos* covers different aspects that arise in the process of monitoring steel structures with UAVs. This includes data acquisition and 3D model reconstruction, automatic path planning, and image analysis. In this paper we put our focus on the aspect of automatic image analysis (outlined in red).

There is a large number of literature on the subject of (semi-) automatic detection of damages on civil and industrial structures. *Koch et al.* [7] gave a comprehensive overview of current practice in visual inspection and damage types of civil structures including associated severity scales. The authors state that listed defect classes are detectable with computer vision based approaches. Further, the state-of-the-art in computer vision defect detection and assessment in the pre-DCNN era is presented. As one of the main open challenges for automatic defect retrieval, an automated process of image and video collection is identified. Recently *Zhang et al.* [8] propose vision based damage detection and classification using an adapted YOLOv3 network and *Huethwohl et al.* [9] use a multi-classifier for reinforced concrete bridge defect detection. Furthermore automatic detection of cracks is discussed [10–13]. The novelty of the presented work besides the definition of a hierarchy of damage classes is, that compliance with the rules

<sup>1</sup> InÜDosS - "Inspektion, Überwachung und Dokumentation von stahlbaulichen Strukturen" – Inspection, monitoring and documentation of steel constructions

<sup>2</sup> All images courtesy of PSP – Prof. Sedlacek & Partner, Planung und Entwicklung im Bauwesen, Büro Dortmund



Figure 2. Example images from the data set<sup>2</sup>

and regulations of DIN 1076 is discussed. Especially the *observations* (B) and the *simple inspection* (E) should both be performed only visually and without the use of tools (cf. [5]).

We discuss a vision based approach in (semi-) automatic damage detection in bridges and steel constructions using UAVs. Main Contributions of this work are:

- Development of a semantic classification scheme for bridge inspection in public space, which is compatible to the specifications of DIN 1076
- Devising a hierarchy of defect classes
- Implementation and evaluation of a corresponding cascaded classifier based on state-of-the-art CNN object detection

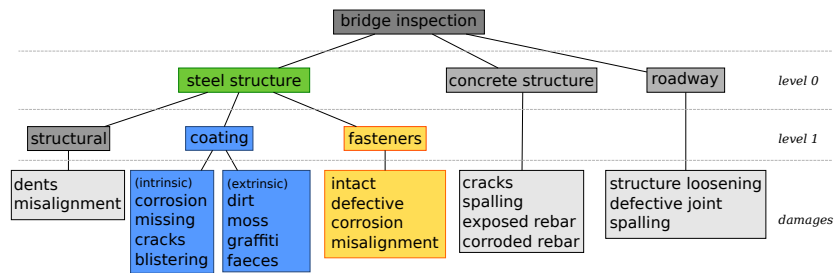
## 2. Materials and Methods

The special nature of the image data gathered demands for a custom approach to tackle the problem. The images at hand show great variety concerning the characteristics of the structures being investigated (cf. Figure 2), as well as the level of detail. Depending on the distance between the UAV and the examined object, the different types of damages only add to the complexity of the problem. Our approach is based on a hierarchy on a conceptual level, as displayed in Figure 3, which we directly transfer to our computational method. The proposed hierarchy consists of two stages. At the first stage we aim to identify relevant structures. Roughly speaking, this comprises all (coated) supporting steel structures. If such a structure is identified, it will then be analysed for damages. At the second stage, we consider two main categories of potential damage, coating damages and faulty fasteners. Coating damages are further subdivided into *blistering*, *cracks*, *missing coating*, and *corrosion*. Fasteners can be *intact*, *defective* (e.g. loose nuts), *tilted*, *missing*, or *corroded*.

We transfer this concept directly to the computational domain by using a two-stage convolutional neural network (CNN) approach, based on *Mask-RCNN* [14]. We use three separate instances of the Mask-RCNN (two on the second stage). All three networks use the ResNet50-backbone [15] and *Feature Pyramid Networks* [16] (FPN). Our implementation is based implementation by *Facebook Research* [17]. The proposed two-stage architecture is shown in Figure 4. The first network ( $\phi_{st}^0$ ) is trained to recognise coated steel structures, resulting in bounding boxes of recognised structures. These bounding boxes are used as regions of interest (ROI) to restrict networks of the second stage. In the second stage, we run two separate networks, one for coating damages ( $\phi_{co}^1$ ) and the other one ( $\phi_{fa}^1$ ) for fastener damages.

More formally, for an input image  $I^0$ , the first-stage detector  $\phi_{st}^0(I^0)$  determines a set  $R_{st}^0 = \{B_0^0, \dots, B_n^0\}$  of bounding boxes  $B_i^0$  which define the areas of relevant structures. For each bounding box  $B_i^0$ , the corresponding sub-image  $I_i^1$  is cut from the full image  $I^0$ , resulting in the set of images  $I^1 = \{I_0^1, \dots, I_n^1\}$ .

Second-stage detectors  $\phi_{co}^1$  and  $\phi_{fa}^1$  are then applied to each sub-image  $I_i^1$ , again resulting in sets of bounding boxes  $R_{co}^1 = \{\{B_{0,0}^{1,co}, \dots, B_{0,m_0}^{1,co}\}, \dots, \{B_{n,0}^{1,co}, \dots, B_{n,m_n}^{0,co}\}\}$  and  $R_{fa}^1 =$



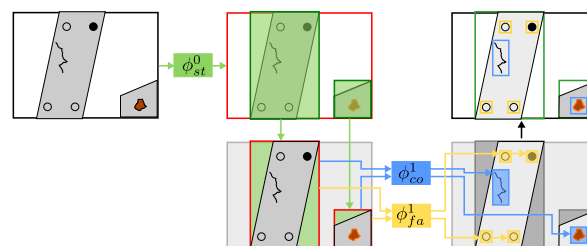
**Figure 3.** Aspects of bridge inspection and proposed semantic hierarchy. Components relevant to this paper are highlighted in colour

$\{\{B_{0,0}^{1,fa}, \dots, B_{0,k_0}^{1,fa}\}, \dots, \{B_{0,n}^{1,fa}, \dots, B_{n,k_n}^{1,fa}\}\}$ . All the results contained in  $R^1 = R_{co}^1 \cup R_{fa}^1$  are then projected back onto the input image  $I^0$ , giving the final prediction results  $R^0$ .

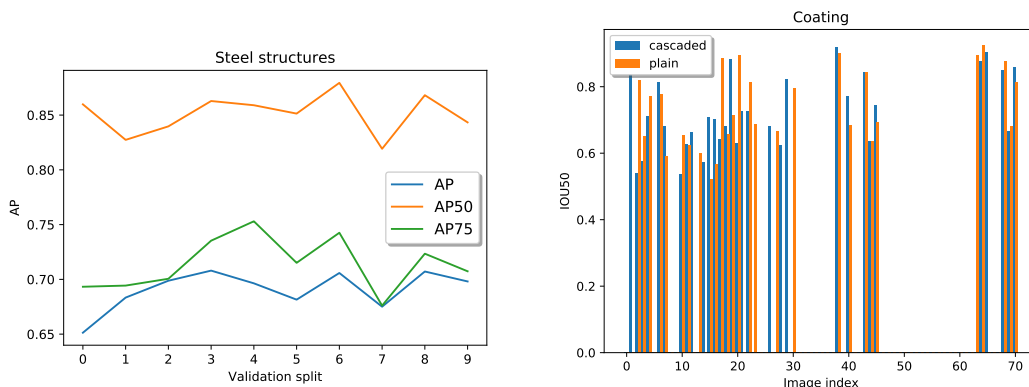
As a data basis, we used approximately 1000 images of steel constructions, consisting mostly of bridges and supporting steel constructions, many of them showing at least one type of damage we are considering. These images are divided into three categories, *steel structures*, *coating damages*, and *fasteners*. Note that the images of the three data sets are not necessarily disjoint, but the corresponding annotations are. To reduce computational costs while increasing accuracy, we use transfer learning. We initialise all three networks with network weights pre-trained on the COCO (Common Objects in Context) data set [18]. We trained each network for 80 epochs, at a learning rate of  $2 \times 10^{-3}$  and a batch size of 2. Further, an initial warm-up phase of 500 iterations (batches) is used to overcome difficulties which may arise in early phases of optimisation [19]. During training, validation tests were performed each epoch. Data augmentation included horizontal and vertical flip, as well as variation in brightness, contrast, and saturation. Data sets were split into 11 parts of (near-)equal size. One part was used as a test set and the other ten parts for cross-validation (10-fold split).

### 3. Results

Figure 2 shows examples of the images used in this paper. For each validation test, we determined the model(s) with the best AP, AP50, and AP75, respectively, Figure 5a shows the average precision on the validation set of the ten-fold split (cf. section 2). This image shows the evaluation for the first stage (steel constructions). To make results of the cascaded and plain version comparable, we first apply the trained detector for supporting structures and then apply the same model as in the simple step on all areas suggested by the first detector. The results of this step are then projected back on the original (full) image.



**Figure 4.** Two-stage CNN pipeline: (coated) steel structures and their positions are identified and errors are classified on the associated bounding boxes



(a) Average precision on validation sets (stage 1) (b) IOUs of simple and cascaded variants for category coating

This allows for directly comparing the results of the plain and the cascaded detector. We selected models according to the best performance regarding average precision.

To test our proposed two stage method, we exemplarily performed tests regarding the second-tier category *coating* in a simple version and a cascaded version. For the second level coating, we additionally trained networks on the same data but with reduced number of classes, in order to test how well the network is able to only recognise and spot damages without further classifying them. In these tests we used the classes *intact* and *defective*.

For the simple version, we applied the trained model of the given category on the corresponding test data set as is common practice. In this case, the detector is simply applied to the whole image. Figure 5b shows IOUs on image level for a test set of 70 images. The cascaded version shows an improvement of nine percent towards the plain version. Figure 6 shows detection results for coatings and fasteners. Red boxes indicate detection results for the first stage (steel constructions), while green boxes and outlines show results of the second stage (coating or fasteners).

#### 4. Discussion and Conclusions

Detection of steel structures works very well, while detection and especially classification of damages is more difficult. This is mainly due to the fact that the data and the semantic categories are inherently difficult. Some damages are very subtle and hard to spot from images, even for an expert, while others depend on a significant amount of context information and interpretation. Furthermore, classes are not always clearly distinguishable, and objects may adhere to several classes at once (fasteners may be defective and corroded, for example). Overall spotting works well, and classification may be sensible, even when differing from ground truth annotation. This applies when classes overlap, or classification by a human expert is fuzzy by itself.

An automatic visual detection of defects on steel bridges complies the requirements of DIN 1076 and Ril 804.80. Required inspections are, with the exception of the *main inspection* (H), only based on visual monitoring. Numerous areas of the bridge cannot be reached and inspected without big units like aerial work platforms or units to inspect bridges from below. For *observations* (B) and *simple inspections* (E) these tools are not intended. Thus, a periodic flyover with an UAV could improve and specify both inspections (B, E). Further, existing defects are observed and checked regularly and small changes would be detected immediately.



**Figure 6.** Cascaded prediction examples: red boxes indicate results from the first stage, green boxes and outlines show results from the second stage

The presented work shows the potential of the developed approach in designing an DIN 1076 compliant visual inspection pipeline for automated damage detection. Due to more precise annual inspections by UAV, which would replace the superficial and visual inspections without tools (B, E), a process of change of a defect would strike immediately and maintenance measures can be taken. The structure and its durability could be improved and maximised. This presents a further step towards smart city infrastructures.

**Author Contributions:** All authors contributed substantially to the presented work.

**Funding:** This work was funded by the Forschungsvereinigung Stahlanwendung (FOSTA e.V.) – project number P1326/17/2018 “Inspektion, Überwachung und Dokumentation von stahlbaulichen Strukturen”.

**Acknowledgments:** Parts of the images used are courtesy of PSP – Professor Sedlacek & Partner, Planung und Entwicklung im Bauwesen, Büro Dortmund.

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

## References

1. Bundesanstalt für Straßenwesen BASt, Bergisch Gladbach. Brücken an Bundesfernstraßen, Brückenstatistik. [https://www.bast.de/BASt\\_2017/DE/Statistik/Bruecken/Brueckenstatistik.pdf](https://www.bast.de/BASt_2017/DE/Statistik/Bruecken/Brueckenstatistik.pdf), 2017. Accessed: 10.14.2019.
2. DB Netz AG. Brückenkarte. <https://bruecken.deutschebahn.com/brueckenkarte>. Accessed: 10.14.2019.
3. Bundesregierung. Verkehrsprognose 2030 . <https://www.bundesregierung.de/breg-de/aktuelles/zunehmender-verkehr-erfordert-investitionen-440698>, 2014. Accessed: 10.14.2019.
4. DB Netz AG. DB Richtlinie 804, Modul 8002: Ingenieurbauwerke planen, bauen u. instand halten – Inspektion von Ingenieurbauwerken – Eisenbahnüberführungen, 2008.
5. Deutsches Institut für Normung e.V. DIN 1076: Ingenieurbauwerke im Zuge von Straßen und Wegen – Überwachung und Prüfung, 1999.
6. DB Netz AG. DB Richtlinie 804, Modul 8001: Ingenieurbauwerke planen, bauen u. instand halten – Inspektion von Ingenieurbauwerken – Allgemeine Grundsätze, 2008.
7. Koch, C.; Georgieva, K.; Kasireddy, V.; Akinci, B.; Fieguth, P.W. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics* **2015**, *29*, 196–210.
8. Zhang, C.; Chang, C.; Jamshidi, M. Bridge Damage Detection using Single-Stage Detector and Field Inspection Images. *CoRR* **2018**, *abs/1812.10590*.
9. Huethwohl, P.; Lu, R.; Brilakis, I. Multi-classifier for reinforced concrete bridge defects. *AUTOMATION IN CONSTRUCTION* **2019**, *105*. doi:{10.1016/j.autcon.2019.04.019}.
10. Yang, X.; Li, H.; Yu, Y.; Luo, X.; Huang, T.; Yang, X. Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network. *Computer-Aided Civil and Infrastructure Engineering* **2018**, *33*, 1090–1109. doi:10.1111/mice.12412.



11. Dorafshan, S.; Thomas, R.J.; Maguire, M. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *CONSTRUCTION AND BUILDING MATERIALS* **2018**, *186*, 1031–1045. doi:{10.1016/j.conbuildmat.2018.08.011}.
12. Turkan, Y.; Hong, J.; Laflamme, S.; Puri, N. Adaptive wavelet neural network for terrestrial laser scanner-based crack detection. *AUTOMATION IN CONSTRUCTION* **2018**, *94*, 191–202. doi:{10.1016/j.autcon.2018.06.017}.
13. Ni, F.; Zhang, J.; Chen, Z. Zernike-moment measurement of thin-crack width in images enabled by dual-scale deep learning. *COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING* **2019**, *34*, 367–384. doi:{10.1111/mice.12421}.
14. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. Proceedings of the International Conference on Computer Vision (ICCV), 2017.
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* **2015**, pp. 770–778.
16. Lin, T.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944. doi:10.1109/CVPR.2017.106.
17. Massa, F.; Girshick, R. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. <https://github.com/facebookresearch/maskrcnn-benchmark>, 2018. Accessed: 10.14.2019.
18. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context, 2014.
19. Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; He, K. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour **2017**.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).