Using Semantic Segmentation for the Damage Detection of Port and Marine Infrastructures

Frederic HAKE, Marvin SCHERFF, Ingo NEUMANN and Hamza ALKHATIB

Abstract

The ageing infrastructure in ports requires regular inspection. This inspection is currently carried out manually by sensing the entire infrastructure by hand. Such a process is costly as it requires a lot of time and manpower. To overcome these difficulties, we propose to map the harbour structure above and below water with a multi-sensor system and try to automate the classification process in terms of common damage types using deep learning approaches. In the images taken above water, damaged and undamaged zones are localised using a semantic segmentation approach. We make use of a real data set captured at JadeWeserPort Wilhemlshaven to test our approach. The images are divided into smaller sections of 512x512 pixels and these are propagated through the DeepLabv3+ architecture, a modern convolutional neural network for semantic segmentation tasks, which is trained in particular to detect corrosion or rust anomalies. We achieve with a pre-trained ResNet-50 backbone and fully supervised data set IoU scores of 96.0 % and 55.9 % for undamaged and damaged zones as well as F1-scores of 98.0 % and 71.7 %. We show that our approach can achieve a fully automated and reproducible image segmentation and damage detection which can analyse the whole structure instead of the sample-wise manual method.

1 Introduction and Motivation

The ageing infrastructure of sea and inland ports requires new technologies and methods in the preparation and implementation of life cycle management processes. The traditional processes are usually time- and labour-intensive, and should be replaced by new automated, smart and innovative measurement and analysis processes to ensure transparency, resource efficiency and reliability for a more dependable lifetime prediction.

Port infrastructure, such as quay walls for loading and unloading ships, bridges, locks and flood gates, are mostly made of concrete, bricks, steel and, in the case of very old structures, wood. They are subject to severe degradation due to special environmental conditions and human activities throughout their lifetime. The material of seaports is especially profoundly affected by saltwater, which damages the concrete structures, sheet pile walls or wooden constructions. It is crucial to detect any damage and categorise its importance to ensure the safety and stability of the infrastructure. Identifying structural damage in time allows the possibility of early maintenance to avoid expensive repairs or even more crucial damage or collapse of the infrastructure.

Nowadays, the monitoring of port infrastructural buildings is divided into the parts above and below water. The structural testing of port infrastructure above water is carried out by manual and visual inspections. The recording and documentation of the condition of damage below water involve considerably more effort; the infrastructure is tested sample-wise every 50 to 100 m; the divers slide down the structure and try to sense the wall with their hands. The results depend directly on human sensory tests. Therefore, damage inspections below water with divers are highly variable in quality and quantity. Damage classification and development are not reproducible due to subjective perception. In addition, there is usually no comprehensive inspection below water, thus, only a small percentage of the structure can be inspected by divers. One way to deal with this problem is by utilising cameras that capture the object surfaces.

The focus of this paper is on the general process of damage detection in images. We examined and optimised different image classification and segmentation methods in the field of machine and even deep learning in order to automate this task. For that purpose, one data set is created to validate the overall procedure. It is a real data set showing a concrete quay wall, which is regularly supported by steel pipes, and captured in the harbour of Wilhelmshaven, a city located in north Germany with the main focus on the detection of rust damages.

It is essential when monitoring harbour structures to assure a transparent, efficient and qualitycontrolled process. This can be achieved by a comprehensive visual inspection in short time intervals during the whole life cycle of the structure. However, a quality-controlled visual inspection is nearly impossible in regions such as the Ems, Weser and Elbe due to the high level of sedimentation. In this research, a fully automated, quality-controlled and reproducible mapping and damage detection of port infrastructures, above and under water, is proposed. Based on the results obtained, the port operator has more reliable information to efficiently plan maintenance and construction work. This approach will reduce the expenses significantly by lowering the downtimes of the port facilities and well-planned construction. Damage detection is usually performed in modern data processing based on pattern recognition methods (see HESSE et al. (2019) for more information). This is a reliable approach to detect any damages and make a well-founded assessment of the current state of the structure. Not only precise but also high-resolution data for the above and below water parts of the building are required for the acquisition of the building geometry and condition.

Various publications deal with comprehensive sensing methods for the structural health monitoring of infrastructural buildings. DIAZ et al. (2017) used edge detection algorithms in MATLAB in order to segment colour images. A segmentation pipeline based on wavelet transforms, principal component analysis and pattern classification was proposed by GHANTA et al. (2011). DUY et al. (2020) introduced a new architecture-based fully convolutional network for semantic segmentation to recognise rust on power electric towers. A Bayesian classifier in combination with Fisher indexes on simulated images with Perlin Noise was presented by ACOSTA et al. (2014). KHAYATAZAD et al. (2020) developed an artificial intelligence-based algorithm that can recognise corrosion damage in a series of photographic images. O'BYRNE et al. (2013) detected disturbances by the texture segmentation of colour images. GATYS et al. (2015) showed that neural networks trained on natural images learn to represent textures in such a way that they can synthesise realistic textures and even entire

scenes. Neural networks, as feature extraction, are, thus, preferred over hand-crafted ones (YOSINSKI et al. 2017, CARVALHO et al. 2017, ABATI et al. 2019). A novel data-driven fault diagnosis method is proposed based on convolutional neural networks (CNNs) by LI et al. (2019). HAKE et al. (2022) proposed a novel anomaly detection approach in which a point cloud is transferred into an image and then processed through a machine learning pipeline to detect concrete spalling.

In this work, we aim to detect structural damages in infrastructures based on colour images. We use anomaly detection algorithms due to the large imbalance between damaged and undamaged areas and the small amount of training data for the damaged areas. The automatic detection of rust damage in images is totally new in the context of structural health monitoring systems. It is now for the first time possible to detect damages in an automated manner. This opens the door for further research into digitally assisted building inspection. Therefore, the approach developed is applicable in all areas of damage detection for infrastructure objects.

The remainder of this paper is organised as follows. Section 2 presents the proposed segmentation pipeline. In Section 3, a real experiment is conducted to evaluate the effectiveness of the proposed method. The conclusions are given in Section 4.

2 Methodology

In this section, all steps from data acquisition to model training and optimisation are listed. All steps in the entire processing pipeline are illustrated in Figure 1. The remaining steps are presented later.

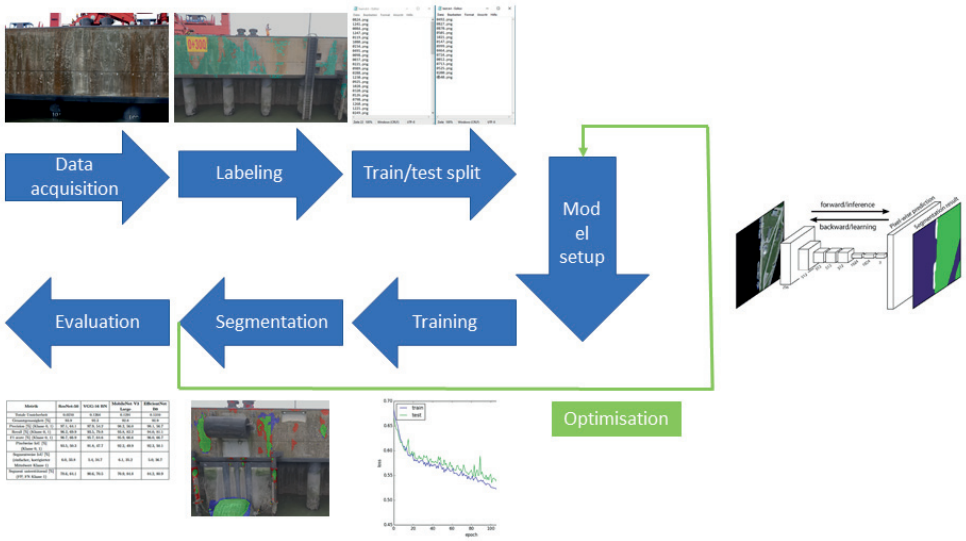

Fig. 1: Processing pipeline for machine learning related projects

For every supervised machine learning related project data availability and preparation are the key elements for a successful product. These steps help to enable the full potential of any considered model architecture. The more diligently the data is collected and pre-processed, the higher the chance to profit from the learning algorithms behind a selected method. After pre-processing, the next challenging step is to adjust the model structure and learning strategy to obtain high-quality results for general applications in the targeted field of work.

In the following subsections, we cover all the steps in Figure 1 except the model evaluation to highlight the results of different trained models afterwards. These steps include data collection of the actual measurement system, the labelling process and optimisations of the considered AI models. We have chosen the Random Forest (RF) (BREIMANN 1984) classfier as a classical machine learning model extensively examined by (TINGTING et al. 2019) and the Deeplab V3+ (CHEN et al. 2018), a state-of-the-art segmentation network developed by Google, to demonstrate the performance of self-learned image features.

2.1 Data acquisition

In order to achieve comparable detection results from the machine learning models with respect to the developed platform, a real-world data set is acquired above water using a camera-equipped drone. The images are taken orthogonally to the quay wall of the Jade-Weser-Port facility at a distance of a couple of meters while the drone hovers freely a couple of meters over the water surface. Therefore, a high-resolution digital image of the port can be analysed. We used a Canon EOS 5D Mark III camera with 35 mm fixed focal length. All settings were unchanged and the pictures were taken on one day. We estimate, that one pixel covers between two to five millimetres in each spatial dimension. The main purpose of this data set is the detection of corrosion on a concrete wall. Other types of damage occur so infrequently (maximum low double-digit number across the entire facility) that the models were unable to learn or to recognize them effectively. The detected damaged areas identified by our trained models can be directly integrated into the digital model of the port facility to track its life cycle status.

2.2 Data Preprocessing

The first task is to label the acquired image data set pixel-wise into two classes. These classes represent undamaged areas or background (sections outside the quay wall) and corrosion, which are treated as class 0 and 1 in the learning algorithms of the models. This time-consuming work was done manually by means of our own developed MATLAB program (MATLAB 2020). Each marked region is therein converted into a white segment and positioned successively onto a dark black/white 8-bit image. An example of this work is shown next to the related RGB image in figure 2.

The complete data set of the port in Wilhelmshaven consists of 1300 images. From all images, 80 are labelled manually. These images are almost uniformly spread over the facility to represent the characteristics of different corrosion variants and their appearing frequencies in a certain neighbourhood. This was done to ensure that the image and ground truth pairs ap-

Fig. 2: Cropped RGB (left) and created ground truth image (right) of the Jade-Weser-Port in Wilhelmshaven. Undamaged and corrosion class are shown in black and white, respectively.

proximate the real distribution of RGB pixel values and related classes of this quay wall. The machine learning models were able to recognize the damage type corrosion. We expect that they will be able to differentiate between the two classes on other image data sets as well. However, this claim has not yet been tested. The training and test sets were split 80/20 and randomly assigned. The data sets were checked to ensure that they covered the specific characteristics of the port, such as lighting conditions. To validate the trained models, the training set was divided into five subsets and the cross-validation method was applied.

When the models are evaluated on the test set, the uncertainty of the ground truth images caused by non-expert knowledge has to be taken into account. Thus, small differences in metrics, like accuracy or Intersection over Union (IoU) score, do not reflect objectively, that one model predicts the true corrosion segments with higher quality in terms of shapes or positions in the scene. The same level of label noise is also included in the training data set, but the deep learning approach is able to generalise the appearance of corrosion better due to it's ability to extract or learn its own features from the training images. The same phenomenon is even shown with a small corrosion data set in (NASH et al. 2019).

2.3 Segmentation with Random Forest

TINGTING et al. (2019) suggest that Random Forest outperforms other machine learning classifiers, such as Support Vector Machine, Bayesian Network, or even Multilayer Perceptrons, regarding corrosion detection by a significant amount. Thus, we adopted a Random Forest algorithm and applied it to our data set to see the segmentation performance of a classical machine learning method.

The RF classifier consists of several binary decision trees (ordered, directed graphs), which individually recursively separate the feature space into classified regions. These regions are represented by leaf nodes in each tree. To obtain a classification result for a new sample, it goes through each tree from top to bottom (root to leaf). On every depth level, the sample follows one out of two paths (edges in graph context) until a leaf node is reached. The covered way on every decision tree depends on the corresponding feature value compared to the learned threshold. The final classification is determined by the winner class of a majority

vote, in which each tree is treated equally. Every decision tree is trained on a data subset (bootstrapping) and the considered feature on each level is selected randomly. The threshold values are determined such that they lead to the highest information gain ($G(Q_m,\Theta)$) after splitting a subset of the feature space along the individual considered axis. This degree of separation is measured either using entropy or Gini impurity. Thus, they can be substituted for one of the listed H quantities in equation 1 (in accordance to the optimisation purpose).

$$G(Q_m,\Theta) = \frac{n_m^{left}}{n_m} H(Q_m^{left}(\Theta)) + \frac{n_m^{right}}{n_m} H(Q_m^{right}(\Theta)) \tag{1}$$

$$H_{entropy}(Q_m) = -\Sigma_k p_{mk} log(p_{mk}) \tag{2}$$

$$H_{gini}(Q_m) = \Sigma_k p_{mk}(1 - p_{mk}) \tag{3}$$

where Q is a subset of the considered n samples, m corresponds to a particular node in a decision tree, $\Theta = (j,t_m)$ indicates a possible candidate split consisting of the feature j and threshold t_m and, finally, *left* and *right* refer to the split subsets (BREIMANN 1984).

The Random Forest classifier uses RGB pixel values and the improved HLS colour space as features to differentiate between two classes. The IHLS colour space is created using non-linear functions from the original pixel values and has been shown to improve the F1 score by 3.5 % in the experiments of TINGTING et al. (2019). We transformed the data set using Principal Component Analysis (PCA) to improve the RF's performance.

To optimize the Random Forest classification model, we conducted a grid search for the number of decision trees and the class weights. The initial values were based on previous research (TINGTING et al. 2019) and the "balanced" setting of the RF class in the Scikit-learn Python package (PEDREGOSA et al. 2011). The search intervals for the number of trees were from 8 to 14 with a step size of 2, and the weights for the corrosion class ranged from 2 to 5. The weight for class 0 was always set to 1. This optimization was done with respect to the weighted F1 score between the two highly imbalanced classes, using the RGB features. Further parameters corresponding to bootstrapping, minimal samples per leaf node, maximal tree depth, considered features and information gain quantity were fixed for the entire process as 'False', 0.01 %, 14, None (all features) and 'entropy', respectively. To reduce the training time, we used a subset of the data with a few hundred million pixels, and we prevented overfitting by restricting the depth of the decision trees and optimizing the model in terms of the entropy metric (see chapter 4). We trained several models with different parameter pairs, and found that 10 decision trees with a class weight ratio of 1:3 achieved the highest F1 score. We then trained two additional instances of the same model, one using only the IHLS colour space and the other using all six colour features. We did not perform further hyperparameter optimization as we did not expect the model to require a more complex understanding of the data. To validate the model, we trained five different models on different parts of the training data set and evaluated them on the test set.

2.4 Segmentation with DeeplabV3+

The DeepLabV3+ architecture (CHEN 2018) was chosen for this study due to its success in various computer vision tasks. In order to adapt it to the specific task of identifying corroded areas in the images, the structure and some parameters of the network were modified (s. Figure 3). One such modification was the doubling of the dilation parameters within the Atrous Spatial Pyramid Pooling (ASPP) block. The modified network was then trained and evaluated using various feature extraction networks ("backbones") and loss functions with the goal of maximizing the IoU score for the corrosion class. The IoU score is a measure of the overlap between the predicted segment and the corresponding ground truth label, expressed as a percentage of their union. This is a common metric in the field of semantic segmentation.

2.4.1 Structure and Modifications of the DeepLabV3 Network

The Deeplab V3+ network is an encoder-decoder architecture that processes images. The encoder extracts features from the image and the decoder reconstructs the image by assigning class probabilities to each pixel. The encoder reduces the spatial resolution of the original image and increases the number of channels while producing a compressed representation of the image.

Initially, an image is passed through the encoder of a deep convolutional neural network (DCNN) and the extracted feature maps are then analysed in an ASPP block. The CNN in CHEN et al. (2018) is named Modified Aligned Xception Net and represents an optimised version of the original Xception network (CHOLLET 2016). For our testing, we also considered further CNNs, which consist of a significantly lower number of trainable parameters and their architecture are based on other basic blocks. Among them, especially the inverted residual block occurs frequently (MARK et al. 2019), which consists of lower number of parameters than the regular residual block and achieves the same or even superior performance. All of the networks are pretrained on the ImageNet-1K data set (DENG 2009) (s. torchvision documentation (MARCEL 2010)) to be more robust against overfitting in the training procedure and provide suitable low-level features, which are transferred directly to the decoder part in order to maintain the object boundaries in the final segmentation image. The feature maps at the end of the CNN extract more detailed information in regards to the image scale by using different types of convolution operations and a Global Average Pooling (GAP) layer.

2.4.2 Processing and Training the DeepLabV3 Network

To improve the accuracy of the Deeplab V3+ network, the input images are split into smaller patches of 512x512 pixels. This is done to avoid running out of memory during training, but also because the network is not able to effectively utilize larger images due to the restricted receptive field in the output layer or the maximal size of corroded areas. Currently, we construct the segmented image by blending the predicted images smoothly together (CHEVALLIER 2017). These image patches already provide a significant amount of contextual information for the segmentation task, and our experiments have shown that adding an overlap of 25 % to the training images can improve the model's performance by

up to 1-2 % for certain metrics listed in table 1. This is likely due to the different perspectives the model is able to gain on individual corrosion segments, especially when they are located near the edges of the original patch. However, its important to note that the performance may degrade due to overfitting if the overlapped area exceeds around 40 %.

To prepare the images for training with the Deeplab V3+ network, they are normalized based on the mean and standard deviation of the ImageNet data set. This helps to take full advantage of the pretraining process. In addition, we apply various image augmentation techniques to artificially expand the data set and reduce overfitting. These include random resized cropping, rotations, horizontal and vertical flipping, colour jittering, and Gaussian blur. These operations are applied with certain probabilities to each image, and are designed to create realistic port photos that can be used for training.

2.4.3 Optimization and Evaluation of the DeepLabV3 Network

In order to improve the performance of the Deeplab V3+ network, we made some changes to its structure and tested different feature extraction networks and loss functions. These changes included adding dropout layers and increasing the filter size within the ASPP block. A schematic overview is illustrated in Figure 3.

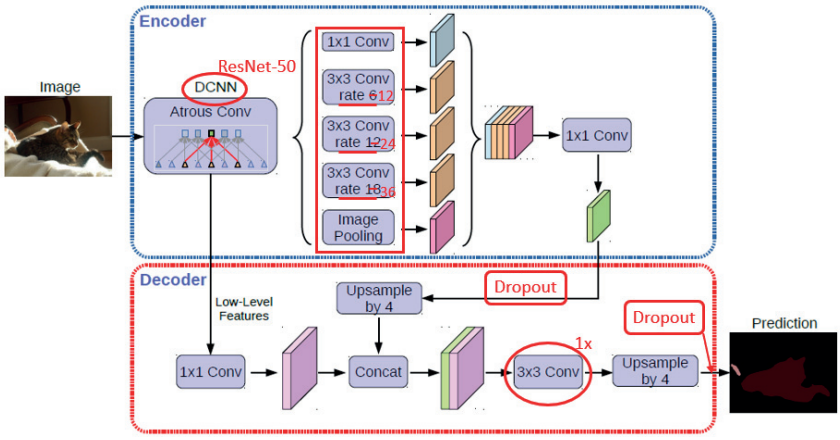

Fig. 3: Modified structure of the Deeplab V3+ architecture (edited figure from CHEN et al. 2018). Changes are highlighted in red.

We tested VGG-16 BN and ResNet-50 (version 1.5) as feature extraction networks and made our decisions based on the IoU score for the minority class. We tested different combinations of loss functions and feature extraction networks to optimize the performance of our deep learning model, specifically focusing on improving the IoU score for the minority class. The loss functions included both distribution-based and region-based methods, such as categorical cross-entropy and IoU loss. The backbones consisted of popular architectures from

2014 to 2019 that have been successful on large classification benchmark data sets, and have fewer parameters than the Xception Net. We tested ResNet-50, VGG-16 BN, MobileNet V3 Large, and EfficientNet B0, which were chosen for their availability in the Pytorch framework (PASZKE 2019) and to limit the number of experiments. To address the imbalanced data, we introduced class weights to the error functions, with a weight of 5 for the corrosion class in the ResNet model, which was found to be the best model according to the IoU score. The best performing combinations of backbones and loss functions were ResNet-50 with IoU loss, VGG-16 BN and MobileNet V3 Large with categorical cross-entropy loss, and EfficientNet B0 with Dual Focal loss (HOSSAIN et al. 2021). The model's performance was validated using the same procedure as in the Random Forest case, with model parameter states taken from the epoch with the lowest loss score on the remaining training subset.

3 Application to real Data

3.1 Segmentation with Random Forest

In this section, we evaluate the three trained Random Forest classifiers to assess how useful the Improved HLS colour space is for the corrosion task. The individual metrics of the models associated with the RGB and both colour spaces are listed in table 1.

The total uncertainty (model and data uncertainty) computed by means of the entropy based on the probabilistic or Softmax output (in case of the segmentation network) is quite similar between the three models. Since these were determined with the same images, this also corresponds directly to the distance in terms of model uncertainty. The model trained on the IHLS colour space achieves 0.2733 compared to 0.2538 by the RGB related classifier. This is also reflected in the other variables such as the overall accuracy, the F1 score and the pixel-wise agreement. In these rather general metrics, the RF trained on the RGB colour space outperforms the IHLS model by up to 2.0 % and even 8.0 % for the background and corrosion class, respectively. We have therefore not observed the same behaviour as TINGTING et al. (2019). While the PCA pre-processing has a positive effect on the RGB model by improving for instance the IoU of corrosion class by approximately 3 %, the IHLS results remain the same.

With the classifier trained on both colour spaces, we see results in between these RF classifiers, but closer to the RGB model performance. The three general metrics fall short by up to 0.6 % (only 0.3 % for background class). Further experiments showed that an increased maximum depth of each decision tree in this model had only a minimal effect on improving the results, but the RGB model's performance could not be matched. This shows that more information doesn't have to improve the problem understanding in general as stated in the earlier assumption. Due to the small difference some prediction images may profit from the additional knowledge nevertheless, but we can not say this with for sure. Figure 4b shows one example image from our data set with overlaid classification results. Correct classification of corrosion is coloured in green, blue are false positive (classified corrosion where there is none) and red is false negatives (did not detect the corrosion).

3.2 Segmentation with DeeplabV3+

The results of the different Deeplab V3+ versions are compared in the section below and table 1 is completed by the best and worst segmentation network to demonstrate the wide range of outcomes. The Deep Learning models were trained for a maximum of 35 epochs and their states of trainable parameters were saved and used afterwards where the mean error with respect to the test set was the smallest.

In terms of overall uncertainty, the ResNet model performs significantly better than the other models, particularly due to it's use of the intersection over union (IoU) loss function. When considering overall accuracy, the ResNet model performs the best, followed by the MobileNet and EfficientNet models with 95.5 % and 95.4 % accuracy, respectively. The VGG-16 Deeplab model has the lowest accuracy at 95.2 %. The majority class (class 0) has a F1 score of 97.4 % or higher for all models. However, the performance for class 1 is lower, with F1 scores ranging from 68.4 % to 71.7 % (in order from the VGG-16 to the ResNet model). The IoU values follow a similar trend, but generally have lower values. For the damage-free areas, the IoU values are about 2 % lower, and the predicted corrosion class overlaps with the ground truth by 53.9 % ± 2.0 %.

Figure 4a show a example image from our data set with overlaid classification results. Correct classification of corrosion is coloured in green, blue is false positive (classified corrosion where there is none) and red illustrates false negatives (did not detect the corrosion).

(a) Segmented images from ResNet-50 Deeplab V3+ segmentation model

(b) Segmented images from Random Forest classifier based on RGB features

Fig. 4: Correct classification of corrosion is coloured in green, blue are false positive (classified corrosion where there is none) and red false negatives (did not detect the corrosion)

Table 1 give an overview of different trained Random Forest and Deeplab V3+ models regarding common metrics calculated on the same corrosion test set.

3.3 Validation

To evaluate the generalization ability of the trained machine learning models, we applied the cross-validation method described in Chapter 2. The results for the best model are shown in Table 2, along with the same evaluation metrics as before.

Table 1: Results of chosen Random Forest and Deeplab V3+ classifiers

	Random Forest		Deeplab V3+	
Metric	**RGB**	**RGB & IHLS**	**ResNet-50**	**VGG-16 BN**
Total uncertainty	0.2538	0.2618	0.0161	0.1121
Overall accuracy [%]	94.1	93.8	96.2	95.2
F1 Score [%] (class 0, 1)	96.8, 59.9	96.6, 59.3	98.0, 71.7	97.4, 68.3
Pixel-wise IoU [%] (class 0, 1)	93.8, 42.8	93.5, 42.2	96.0, 55.9	95.0, 51.9

Table 2: Cross validation results of the RGB Random Forest and ResNet-50 Deeplab V3+ model

	RGB Random Forest		ResNet-50 Deeplab V3+	
Metric	**Mean**	**Standard dev.**	**Mean**	**Standard dev.**
Total uncertainty	0.2519	0.0084	0.0181	0.0015
Overall accuracy [%]	94.1	0.3	95.8	0.3
F1 Score [%] (class 0, 1)	96.8, 59.7	0.2, 1.0	97.8, 69.4	0.2, 0.5
Pixel-wise IoU [%] (class 0, 1)	93.8, 42.5	0.3, 1.0	95.4, 53.2	0.2, 0.6

The results of the cross-validation evaluation show that both the Random Forest and Deeplab models perform similarly overall (as indicated by the mean values in the table). However, the Deeplab model demonstrates a significant increase in the detection rate of corrosion (2.7 %increase) compared to the Random Forest model, which is almost unchanged by the extra data fold (20 % of the training set). Additionally, the results of the Deeplab model are more consistent across the five separate models, as indicated by the lower standard deviation values. In particular, the class 1 scores for the RF model vary around a factor of two and it's prediction uncertainty is almost six times higher compared to the trained Deeplab models. This demonstrates the superiority of self-learned features of the Deep Learning approach.

4 Discussion and Outlook

In this work, different machine and deep learning models have been investigated for the segmentation of damaged areas on a port facility image data set. In an attempt to promptly detect the emergence of damages and in order to be able to follow their emergence and temporal development in a Building Information Modeling (BIM) approach with high accuracy. These models are to be used to automatically evaluate the images captured by an Multi-Sensor-System (MSS) platform and drone, primarily to support the structural inspectors in assessing the facility.

Using the example of a real data set from Wilhelmshaven, which is characterised by various forms of corrosion damage, the machine learning architectures and their training procedures were optimised and evaluated.

Evaluation of the results shows that the approach with Deeplab V3+ and ResNet-50 delivers only 2.1 % better overall accuracy than the adapted approach with the Random Forest classifier. On the other hand, the precision score of the Deeplab V3+ approach with ResNet-50 for class 1 is 11.8 % higher than in the RF classifier and the precision for class 0 is still slightly better with 1.2 %. Similarly, when comparing the pixel-wise IoU score, the Deeplab V3+ approach is 13 % higher for class 1 than the RF classifier and the IoU for class 0 is also roughly 2 % higher. By comparing the results from the two models, it is clear that the Deeplab model performs better in terms of reducing false positives and increasing false negatives. False negatives are areas that were not identified as corrosion but were actually corroded. These areas are typically (in more than 75 % of the cases) located near true positives, which means that they are still within the range of the damage detected by the model. This is not a significant problem in the context of building inspection, as any detected areas of damage will still be examined by a building inspector. Overall, the performance of the Deeplab model demonstrates the superiority of self-learned features in comparison to the traditional machine learning approach.

Therefore, the approach with Deeplab V3+ delivers better results in the segmentation of rust spots in comparison to the classical approach represented by the Random Forest classifier. It is possible to improve the results by including additional scales. For example, in the ASPP block of the Deeplab encoder, it is not only possible to use the kernel spacing once set up to three times. This has so far been increased from six to twelve, as it has led to considerable improvements, but this makes it difficult to detect smaller damage with high geometric accuracy. This can be investigated in more detail in the future with a multi-dimensional grid search. In addition, the number of Atrous convolutions could also be increased to cover multiple scales regardless of the data set.

The comparison of the different feature spaces in table 1 has shown that the use of the IHLS colour space has not achieved the improvements in accuracy shown in (TINGTING et al. 2019). This is probably due to the different data sets and objects. But the RF model trained on the original image data could be improved strongly by means of the PCA pre-processing.

So far, we have only been able to test our algorithm on one real data set of a sheet pile wall with a concrete spar. However, we are working towards acquiring more real-world data with different materials and building types. The proposed strategy is also applicable to other infrastructure objects, such as bridges, high-rise buildings, and tunnels.

Funding: This research was funded by Federal Ministry of Transport and Digital Infrastructure grant number 19H18011C.

Acknowledgments: This work was carried out as part of the joint research project ”3DHydroMapper”. It consists of five partners and one associated partner: Dr. Hesse und Partner Ingenieure, WK Consult, Niedersachsen Ports, Fraunhofer IGP, Leibniz University Hannover and Wasserstraßen- und Schifffahrtsverwaltung des Bundes.

Bibliography

Abati, D., Porrello, A., Calderara, S. & Cucchiara, R. (2019): Latent Space Autoregression for Novelty Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Long Beach, CA, USA, 15-20 June 2019, pp. 215-232.

Acosta, M. R. G., Díaz, J. C. V., & Castro, N.S. (2014): An innovative image-processing model for rust detection using Perlin Noise to simulate oxide textures. In: Corrosion Science, pp. 141-151, DOI: 10.1016/j.corsci.2014.07.027.

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984): Classification and Regression Trees. Wadsworth 10.2307/2530946.

Breiman, L. (2001): Random Forests. In: Machine Learning 45, pp. 5–32, DOI: 10.1023/A:101093-3404324.

Carvalho, T., de Rezende, E. R. S., Alves, M. T. P., Balieiro, F. K. C. & Sovat, R.B. (2017): Exposing Computer Generated Images by Eye's Region Classification via Transfer Learning of VGG19 CNN. In: 2017 16th IEEE International Conference on Machine Learning and Applications, pp. 866-870, DOI: 10.1109/ICMLA.2017.00-47.

Chen, L., Zhu, Y., Papandreou, G., Schroff, F. & Adam, H. (2018): Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. arXiv 10.48550/ARXIV.1802.02611.

Chevallier, G. (2017): Using a U-Net for image segmentation, blending predicted patches smoothly is a must to please the human eye. Github Last visit: 12/22/2022. github.com/Vooban/Smoothly-Blend-Image-Patches.

Chollet, F. (2017): Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). pp. 1251-1258, 10.48550/arXiv.1610.02357.

Deng, J., Dong W., Socher, R., Li, L.-J., Kai, L. & Li, F.-F. (2009): ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition 248-255, 10.1109/CVPR.2009.5206848.

Diaz, J. A. I., Ligeralde, M. I., Jose, A. C. & Bandala, A. A. (2017): Rust detection using image processing via Matlab. In: 2017 IEEE Region 10 Conference, pp.1327-1331, DOI: 10.1109/TENCON.2017.8228063.

Duy, L. D., Anh, N. T., Son, N. T., Tung, N. V., Duong, N. B. & Khan, M. H.R. (2020): Deep Learning in Semantic Segmentation of Rust in Images. In: Proceedings of the 2020 9th International Conference on Software and Computer Applications, pp. 129–132, DOI: 10.1145/3384544.3384606.

Gatys, L., Ecker, A., Alexander S & Bethge, M. (2015): Texture synthesis using convolutional neural networks. In: Advances in neural information processing systems. pp. 262-270.

Ghanta, S., Karp, T. & Lee, S. (2011): Wavelet domain detection of rust in steel bridge mages. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1033-1036, DOI: 10.1109/ICASSP.2011.5946583.

Hake, F., Göttert, L., Neumann, I. & Alkhatib, H. (2022): Using Machine-Learning for the Damage Detection of Harbour Structures. In: Remote Sensing 14, DOI: 10.3390/rs14112518.

Hesse, C., Holste, K., Neumann, I., Hake, F., Alkhatib, H., Geist, M., Knaack, L. & Scharr, C. (2019): 3D HydroMapper: Automatisierte 3D-Bauwerksaufnahme und Schadenserkennung unter Wasser für die Bauwerksinspektion und das Building Information Modelling. In: Hydrographische Nachrichten 113, pp. 26-29.

Hossain, S., Betts, J. M. & Paplinski, A. P. (2021): Dual Focal Loss to address class imbalance in semantic segmentation. In: Neurocomput, pp. 69-87, DOI: 10.1016j.neucom.2021.07.055.

Khayatazad, M., De Pue, L. & De Waele, W. (2020): Detection of corrosion on steel structures using automated image processing. In: Developments in the Built Environment, pp. 100022, DOI: 10.1016/j.dibe.2020.100022.

Li, G., Deng, C., Wu, J., Xu, X., Shao, X. & Wang, Y. (2019): Sensor Data-Driven Bearing Fault Diagnosis Based on Deep Convolutional Neural Networks and S-Transform. In: Sensors 19, 2750. DOI: 10.3390/s19122750.

Marcel, S. & Rodriguez, Y. (2010): Torchvision the Machine-Vision Package of Torch. In: Proceedings of the 18th ACM International Conference on Multimedia, pp. 1485-1488. Association for Computing Machinery, Firenze, Italy. DOI: 10.1145/1873951.1874254.

MATLAB (2020): Version 9.8.0 (R2020a). The MathWorks Inc. Natick, Massachusetts.

Nash, W. T., Powell, C. J., Drummond, T. & Birbilis, N. (2019): Automated Corrosion Detection Using Crowd Sourced Training for Deep Learning. In: Corrosion 76 (2), pp. 135-141.

O'Byrne, M., Schoefs, F., Ghosh, B. & Pakrashi, V. (2013): Texture analysis based damage detection of ageing infrastructural elements. In: Computer-Aided Civil and Infrastructure Engineering 28 (3), pp. 162-177.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. & Chintala, S. (2019): PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. & Garnett, R. (Eds.): Advances in Neural Information Processing Systems, 32. Curran Associates, Inc., Vancouver, Canada.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011): Scikit-learn: Machine Learning in Python. In: Journal of Machine Learning Research 12, pp. 2825-2830.

Tingting, L., Kai, K., Fen, Z., Jialiang N. & Tianyun W. (2019): A corrosion detection algorithm via the random forest model. In: Zhaohui L. (Ed.): 17th International Conference on Optical Communications and Networks (ICOCN2018), International Society for Optics and Photonics, 16-19 November 2018, Zhuhai, China.

Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. (2014): How transferable are features in deep neural networks? In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. & Weinberger, K. Q. (Eds.): Advances in Neural Information Pro-cessing Systems, Curran Associates, Inc., San Francisco, USA.