remote sensing

MDPI

Article

Using Machine-Learning for the Damage Detection of Harbour Structures

Frederic Hake *, Leonard Göttert, Ingo Neumann and Hamza Alkhatib

Geodetic Institute, Leibniz Universität Hannover, Nienburger Str. 1, 30167 Hannover, Germany; leonard.goettert@stud.uni-hannover.de (L.G.); neumann@gih.uni-hannover.de (I.N.); alkhatib@gih.uni-hannover.de (H.A.)

* Correspondence: hake@gih.uni-hannover.de; Tel.: +49-511-762-4469

Abstract: The ageing infrastructure in ports requires regular inspection. This inspection is currently carried out manually by divers who sense the entire below-water infrastructure by hand. This process is cost-intensive as it involves a lot of time and human resources. To overcome these difficulties, we propose scanning the above and below-water port structure with a multi-sensor system, and by a fully automated process to classify the point cloud obtained into damaged and undamaged zones. We make use of simulated training data to test our approach because not enough training data with corresponding class labels are available yet. Accordingly, we build a rasterised height field of a point cloud of a sheet pile wall by subtracting a computer-aided design model. The latter is propagated through a convolutional neural network, which detects anomalies. We make use of two methods: the VGG19 deep neural network and local outlier factors. We showed that our approach can achieve a fully automated, reproducible, quality-controlled damage detection, which can analyse the whole structure instead of the sample-wise manual method with divers. We were able to achieve valuable results for our application. The accuracy of the proposed method is 98.8% following a desired recall of 95%. The proposed strategy is also applicable to other infrastructure objects, such as bridges and high-rise buildings.

Keywords: damage detection; machine-learning; infrastructure; laserscanning; multibeam echo-sounder

check for updates

Citation: Hake, F.; Göttert, L.; Neumann, I.; Alkhatib, H. Using Machine-Learning for the Damage Detection of Harbour Structures. *Remote Sens.* **2022**, *14*, 2518. https://doi.org/10.3390/rs14112518

Academic Editor: Fernando Vicente-Guijalba

Received: 31 March 2022
Accepted: 21 May 2022
Published: 24 May 2022

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 especial 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 structures. It is crucial to detect any damage and categorize its importance to ensure the safety and stability of the infrastructure. Identifying structural damage in time allows early maintenance and avoids expensive repairs and the 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 the subjective perception. In addition, there is usually no comprehensive inspection below water, thus, only a small percent of the structure can be inspected by divers. One way to deal with this problem is by utilising sensors that detect the shape of the object. Such sensors provide point clouds and include laser scanners for surfaces above water and echo-sonars for those below water. The focus in this paper is on the general process of damage detection in point clouds. We use two datasets to validate the overall procedure. The first is a simulated dataset of a sheet pile wall below water. The second is a real dataset of a concrete quay wall from the northern German city harbour of Lübeck, measured with a laser scanner above water. It is not possible to detect small damages, such as cracks, especially in the area below water, due to the point spacing, which is 20 mm. Therefore, the main focus is on the detection of spalling damages larger then 20 mm.

It is essential when monitoring harbour structures to assure a transparent, efficient and quality-controlled process. This can be achieved by a comprehensive visual inspection at 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 three-dimensional (3D) sensing and damage detection of port infrastructures, above and below 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 [1] 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 exact but also high-resolution 3D 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 concrete or other materials above water or in clear offshore regions. A static underwater multibeam scanner for 3D reconstruction of subaquatic environments was introduced by [2]. Robert et al. [3] used a multibeam echo-sounder and underwater stereo cameras to create a 3D point cloud of vertical marine structures. Hadavandsiri et al. [4] introduced a new approach for the automatic, preliminary detection of damage in concrete structures with terrestrial ground scanners and a systematic threshold. An automatic classification for underwater geomorphological bed forms was presented by Ref. [5], which achieved an overall accuracy of 94%. A long-term monitoring approach for zigzag-shaped retaining structures is propose by Ref. [6]. Aldosari et al. [7] used a ultra-high accuracy wheel-based mobile LiDAR mapping system for monitoring mechanically stabilized earth walls. O'Byrne et al. [8] detected disturbances by the texture segmentation of colour images. Gatys et al. [9] showed that neural networks trained on natural images learn to represent textures in such a way that they can synthesize realistic textures and even entire scenes. Neural networks, as feature extraction, are, thus, preferred over hand-crafted features [10–12]. A novel sensor data-driven fault diagnosis method is proposed based on convolutional neural networks (CNNs) by [13]. However, the limitation of such a transfer of features remains an open research question, especially when the input domain has the same topological structure but different statistical behaviour. The detection of non-normal instances within datasets is often called anomaly detection.

The definition of outliers first mentioned by [14] for outlier detection varies widely nowadays. Anomalies are no longer just understood as incorrect readings, but are often associated with particularly interesting events or suspicious datasets. The original definition was, therefore, extended by [15]:

1. Anomalies are different from the norm regarding their features;
2. Anomalies are rare compared to normal instances in a dataset.

Two widely used methods in anomaly detection are transfer learning and local outlier factors (LOF). Transfer learning adopts pretrained neural networks based on a different domain [16]. This results in advantages such as faster creation, better model quality, and less use of resources (training data). Breunig et al. [17] describe a method called LOF, which judges a sub-element on how isolated it is regarding the local neighbourhood.

Nowadays, anomaly detection algorithms are often used in many application domains. García-Teodoro et al. [18] describe a method using anomaly detection algorithms to identify network-based intrusion. In this context, anomaly detection is also often called behavioural analysis, and these systems typically use simple but fast algorithms. Other possible scenarios are fraud detection [19], medical applications (such as patient monitoring during electrocardiography [20]), data leakage prevention [21] and other more specialised applications, such as movement detection in surveillance cameras [22] or document authentification in forensic applications [23].

In this work, we aim to detect structural damages in infrastructures based on point clouds. 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 novel detection approach we use can classify defective from non-defective features in a simulated data environment. The procedure of transferring features from natural images to point clouds and then performing a novel detection 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 the use of pretrained neural networks for range sensor data. Therefore, the approach developed is applicable in all areas of damage detection for infrastructure objects.

2. Methodology

We first need to preprocess the data, because unstructured, large 3D point clouds are unsuitable for most anomaly detection algorithms. Therefore, we transfer features learnt from natural images to height maps from a range sensor. A height map or height field (also called a digital elevation model (DEM) [24]) in computer graphics is a raster image that is mainly used as a discrete global grid in secondary height modelling. Each pixel records values, such as surface elevation data. In contrast to natural images, the characteristics of height maps depend on the scan resolution and the object scanned itself, which makes transferability difficult. A way to overcome this drawback is to train height map neural networks from scratch [25].

In our system, three different sensor types are merged into one kinematic multi-sensor system (k-MSS) for the mapping task: a high-resolution hydro-acoustic underwater multibeam echo-sounder, an above-water profile laser scanner and five high dynamic range cameras. In addition to the IMU-GNSS-based georeferencing method known from various applications, hybrid referencing with automatically tracking total stations is used for positioning (Figure 1). Although the individual sensors record in a grid pattern, the resulting point cloud is not grid-shaped due to the movements of the carrier platform.

The choice of damage types depends on the application and the relevant task within the life cycle management. In this study, we focus on geometrical damages and, for the time being, we only use point cloud data and no images from the cameras. The point clouds should have the smallest possible distance between the points, but still large enough so that there is no correlation between the points due to overlapping laser footprints. The head of the laser scanner rotates with 100 hz, the platform moves as slow as possible and the position and orientation is obtained from the GNSS/IMU system. Furthermore, the following state-of-the-art does not address mapping and data collection due to the fact that the research contribution lies in the damage detection area. It focuses on damage detection (see [1] for more details on mapping and data collection).

The method starts with a point cloud of typical structures (see Sections 3 and 5 for details). Firstly, we transform the point cloud into a height field, which is described in Section 2.1. Secondly, in Section 2.2, we extract features with a CNN. The third step is the defect detection using two different approaches: transfer learning and LOF (Section 2.3).

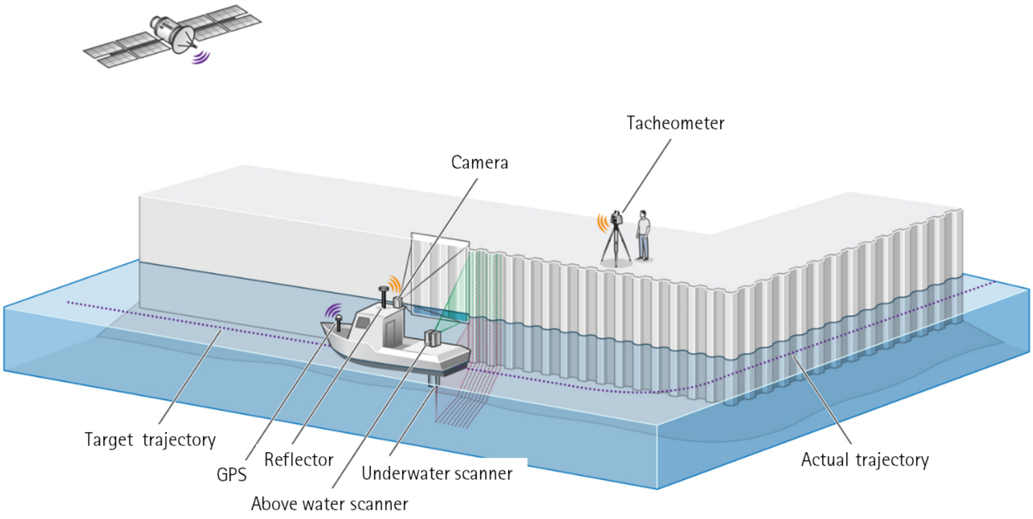

Figure 1. 3D mapping of a port structure above and below water [1].

Both methods yield outlier scores, which can be thresholded to achieve a binary classification. In contrast to other common outlier detection methods, these do not make any assumptions about the distribution of the outliers. They are, thus, well-suited for port infrastructural monitoring where each damage is expected to be unique.

2.1. Height Field Generation

Input variables for the machine learning approach are equally sized and rasterised distances between the point cloud and the original damage-free structure. In an optimal scenario, one can use a computer-aided design (CAD) or building information model (BIM) and determine deviations between the model and the point cloud. Unfortunately, no models are available for most existing infrastructural objects. There are two possibilities to overcome this challenge: the manual or (semi-)automatic generation of a CAD model or the use of an approximated local surface, for example, using a moving-window approach (e.g., [26]).

In the case of the simulated dataset, we use a mathematical model of a sheet-pile wall to create the simulated dataset and the corresponding CAD model. The distances from each point to the corresponding plane in the model are determined according to [27] with Equation (1) and are then rasterised into a two-dimensional (2D) height field with an equal 2 cm raster size,

$$dist = \frac{n_X \cdot p_x + n_y \cdot p_y + n_z \cdot p_z - d}{|\mathbf{n}|}, \tag{1}$$

where $\mathbf{n}$ is the normal vector of the plane with the entries n_x, n_y and n_z. d is the distance to the origin and p_x, p_y and p_z are the co-ordinates of the point.

There is no existing CAD model of the quay wall for the real dataset, therefore, we had to create the model ourselves. For this purpose, regular shapes are fitted into the point cloud and the distance from the points to the geometry is determined. A simple plane according to [27] is used as reference geometry in this work. Firstly, the point clouds are rotated in a consistent direction using principal component analysis. Regular square sections are then cut from the point cloud. These sections overlap by 50% each in the X and Y directions. The cutting into smaller sections is useful to be able to estimate (well) fitting geometries into the point cloud. After cutting, a plane is estimated in each of the sections. We only used the points of the quay wall and the damaged areas for the plane estimation. The distance to the plane is set manually to small values for points that are located on additional objects, such as ladders or fenders. This allows deviations due to damage in the grey value differences to be more clearly visible.

The raster size depends on the resolution of the point cloud and must be adapted to the respective dataset. Empty cells, which occur due to data gaps or inappropriate point distribution, are interpolated according to [28] with natural neighbour interpolation to avoid interference in the feature extraction step (cf. Equation (2))

$$G(x) = \sum_{i=1}^{n} \frac{A(x_i)}{A(x)} f(x_i) \tag{2}$$

where $G(x)$ is the estimate at x, $f(x_i)$ the known data at x_i, $A(x)$ is the volume of the new cell centred in x, and $A(x_i)$ is the volume of the intersection between the new cell centred in x and the old cell centred in x_i.

The median value of the distances is used in overpopulated cells, which occur due to inappropriate point distribution. The whole process is implemented in MATLAB and Python and summarised in a flowchart form in Figure 2.

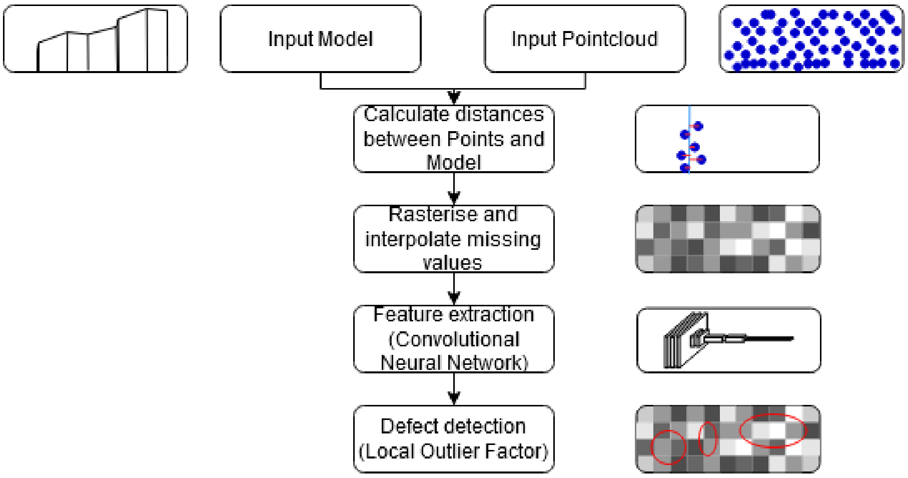
Input Model
Input Pointcloud
Calculate distances between Points and Model
Rasterise and interpolate missing values
Feature extraction (Convolutional Neural Network)
Defect detection (Local Outlier Factor)

Figure 2. Flow chart of the automatic damage detection process.

The height field of the infrastructure object obtained is interpreted as a scalar function defined on a 2D grid, denoted by $H(x, y)$. Afterwards, patches are extracted from the grid and rearranged into data vectors. The latter $\mathbf{x}$ are organised as matrix $\mathbf{X}$ with shape $N \times p$, where N is the number of patches and p the number of pixels. Figure 3 shows an example of such a height field in grey scale, where a lighter grey value represents a greater deviation from the nominal CAD model.

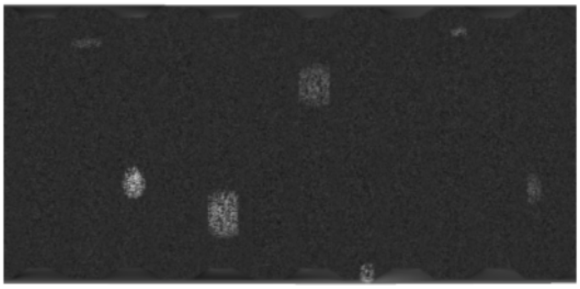

Figure 3. Visualising an exemplary height field on a 2D grid in grey scale.

2.2. Feature Extraction

A deep learning network requires a large amount of high-quality annotated data. However, as damage to port structures is relatively rare, it takes a very long time and a lot of measurements until a sufficient amount of annotated data is available. To overcome this problem, we chose a truncated version of the VGG19 network as the basic backbone for feature extraction and transferred its pretrained parameters on ImageNet to our dataset of port structures. The VGG19 neural network is a standard CNN, pretrained on natural images [29,30]. The network consists of 19 layers and is trained in a classification scenario. It is well-known for achieving superhuman performance on the extensive scale image database ImageNet [31]. The latter consists of more than a million labelled natural images of life scenes (such as cats, people, bicycles), which is very different from the dataset used in this paper. Therefore, we do not use the original network, but a variant that we modified. We only keep the first convolutional layers of the network, including layer pool_4, to prevent overfitting. The reason for this is that deeper layers in the network tend to learn higher order features, such as objects and faces, than lower layers that learn lower order features, such as edges and structures. A comprehensive visualisation can be found in [32]. We focused in this work on the detection of geometric damage such as spalling, which can be described well with lower order features, therefore, we obtained the best results with the network truncated after layer pool_4.

In contrast to the scalar function of the height field, the VGG19 network requires three-channel input (RGB-colour). Therefore, the signal is broadcast over three channels. We may encounter large height fields depending on the length of the wall scanned. We split the height fields into smaller tiles to compensate for hardware limitations. Dividing a large scan into smaller tiles not only increases the computational efficiency but also creates the possibility of achieving more than one label for the whole area. As a result, defects can be more efficiently located based on the smaller size of the tiles. If a defect is located at the border of a tile, affecting more than one tile, a criterion of 50% overlap in both directions is defined for a more reliable defect detection. Every vector $\mathbf{x_I}$ is propagated through the network, and the intermediate activation of the jth layer is stored.

Afterwards, the Gramian matrix of each activation is computed (see [9] for details). We only keep the diagonal of the Gramian matrix, which relates to the energy per feature, for computational efficiency and because we are not interested in synthesising new data. This leads to the feature vector $\mathbf{z} \in \mathbb{R}^k$, where k is the number of feature maps in the jth layer of the network. Note that this procedure always leads to a dimensionality k independent of the input size p. Again, we organise all feature vectors as rows in a matrix, resulting in a feature matrix $\mathbf{Z}$ with shape $N \times k$.

2.3. Defect Detection

The last step of the damage detection process transforms the features that were extracted from the height fields into a single prediction label. Two different but interchangeable methods were used: transfer learning and LOF. Their performance is evaluated and compared in Section 4.

2.3.1. Transfer Learning

We use a three-layer feed-forward neural network to transform the extracted features into a single output label.

Firstly, it consists of two fully connected layers of the same size as the extracted features. These layers use the widely used ReLu (rectified linear unit) activation function [33] to allow for non-linear modelling. Furthermore, a dropout rate of 20% was chosen to help prevent overfitting during the training of the neural network. Secondly, there is a layer with a single neuron that is fully connected to the previous layer. This network is appended to the feature extraction network from Section 2.2. The value of the single output neuron is then used for threshold-based classification.

2.3.2. Local Outlier Factor

This second type of discriminator uses a standard approach called a LOF [17]. It is capable of detecting outliers in data (outliers are data points that do not fit in with the rest of the data). In order to achieve outlier detection, the LOF method constructs a reachability graph in feature space to estimate the density of the neighbourhood. It then computes an outlier score for each data point from this density. There are two different ways of using this discriminator:

1. The first way is to feed an untrained LOF discriminator with new data and let it compute outlier scores for each data point. Using these scores, outliers can be found in new data without any prior training;
2. The second way uses training on clean data (only showing the normal state without any defects) to create the reachability graph. Afterwards, outlier scores can be computed for new data by comparing it with the reachability graph of the trained normal case.

Either way, the outlier score is then used for threshold-based classification.

3. Simulation and Application

3.1. Creating the Dataset

Using a machine learning approach requires a large set of labelled training data. Four steps are necessary for the generation of a simulated point cloud of a sheet pile wall with damages:

1. Generation of a large number of datasets with randomly located and sized damages with a mathematical model for a sheet pile wall;
2. Computation of Cartesian co-ordinates of each point on the planes of the sheet pile wall by projecting rays from the k-MSS in vertical increments. The third dimension results from the movement of the sensor along a given trajectory. We used a straight line with equidistant sensor positions in this study for the sake of simplicity;
3. Addition of a random number of damages onto the planes of the sheet pile wall. Each damage has an ellipsoidal shape with random values for the principal axis;
4. The result is a noise-free point cloud. The ranges of each beam are then contaminated with random instrumental measurement uncertainty. The distance measurement uncertainty is assumed to be a normally distributed standard deviation and set according to the manufacturer's specifications to 20 mm. The uncertainties of the angles are neglected, because they lead to a small shift in the plane direction and a resulting small distant error, which is already included in the 2 cm noise. The resulting resolution of the point cloud is around 2 cm. Figure 4 illustrates the simulation procedure.

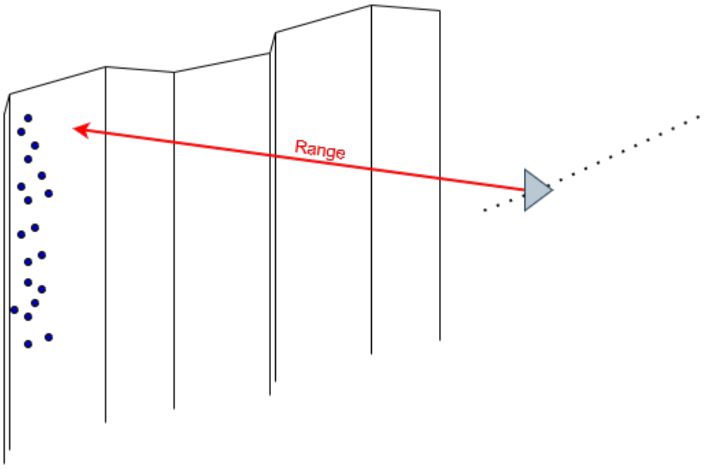

Figure 4. Simulation principal used in this study: at each sensor position, all rays in vertical increments according to the manufacturer's specifications are intersected with the planes of the sheet pile wall.

A second dataset was generated: it indicates for every position whether it is damaged or not, and is this the ground truth. Since the height field is a 2D raster image for the anomaly detection, we also use a 2D binary label image where a value of one stands for damage and zero for undamaged zones (right-hand side of Figure 5).

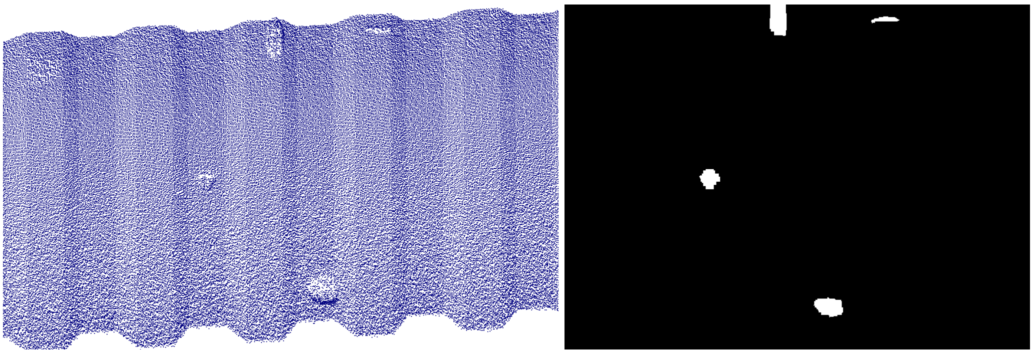

Figure 5. Left is the simulated point cloud and right is the corresponding label image where a value of one stands for damage and zero for undamaged zones.

The raster is generated in the XZ-plane of the point cloud. The image is filtered with the morphological operators erosion and dilation to avoid coarse shapes or misclassified raster cells [34]. Erosion is used to separate two near clusters, and dilation fills small holes. Using erosion after dilation is also called 'opening' in mathematical morphology. Figure 5 shows the simulated point cloud with the corresponding label image.

3.2. Training

The defect detection algorithms were implemented in Python 3.7. The TensorFlow library [35] was used for the pretrained VGG19 network and transfer learning approach. The implementation provided by the library sklearn [36] was used for the LOF approach. The datasets are split into 70% training and 30% test data for both approaches.

3.2.1. Transfer Learning

The training of this network requires height field samples without damages. The network needs to be trained on examples of normal and abnormal data to learn how to discriminate between the two. A different approach had to be used because there are no large datasets with an even amount of normal and abnormal data available. A new dataset was created where half of the examples are undamaged and the other half had the pixels of the height map inside a rectangular region of random size rearranged randomly. Additionally, some additive white Gaussian noise was added in that region. The network is expected to learn the statistical properties of a normal height field.

The network was trained on this new dataset. The unaltered, normal examples get a label of zero, and a damaged one gets a label of one. The network was trained to learn these labels using transfer learning. This means that the pretrained neurons of the truncated VGG19 network are kept fixed, while only the new neurons are trained. A dropout rate of 20% was used to decrease the risk of overfitting.

3.2.2. Local Outlier Factor

The LOF approach can be used in two different ways (see Section 2.3.2):

1. With training: A dataset without defects is required for training the LOF discriminator. The large height field strips are tiled and propagated through the feature extraction stage, as described in Section 2.2. The LOF learns the distribution of the features of the normal case. Even if the training data was not completely free of defects, the training of the LOF discriminator can compensate for a small amount of contamination in the training data. This feature was not used since we have full control of our training data

because it is created artificially through simulation. The trained LOF discriminator can then be saved for inference on new data;

2. Without training: The LOF discriminator can also be used without training. In this case, the LOF discriminator is trained on-the-fly at inference time. In this manner, the LOF discriminator can detect anomalies in the new data without prior training of the normal case.

3.3. Output Merging

Both methods take a large height map image as input, which is split into smaller overlapping tiles. A pretrained truncated VGG19 network extracts features from those tiles. Subsequently, either a neural network or a LOF discriminator is used to convert these features into a scalar value that indicates whether the respective tile shows a defect or not (see Section 2).

The tiles overlap by 50%, thus, each source pixel from the original height map is evaluated four times. The scores for the tiles are merged onto a regular non-overlapping grid of values to allow for a better interpretation, analysis and evaluation of the classification results. The grid cells have the same size as the individual overlapping regions.

Two functions were tested for merging the tiles' values into the grid: min and mean. These functions operate on the values of all tiles, which share the overlapping region at the position of the grid cell and directly determine its value.

The advantage of this method is that the results can be interpreted as a grey-scale image and, thus, can be visualised easily. Figure 6 shows an example of a simulated height map with defects applied (top) and the corresponding outlier activation scores that result from merging the scores of the tiles using the mean function (bottom). The threshold-based classification is also performed on this raster as opposed to directly on the tile's score. This may improve the classification results as it reduces noise. Oversampling is eliminated, therefore, each defect is only considered once, making the evaluation more accurate.

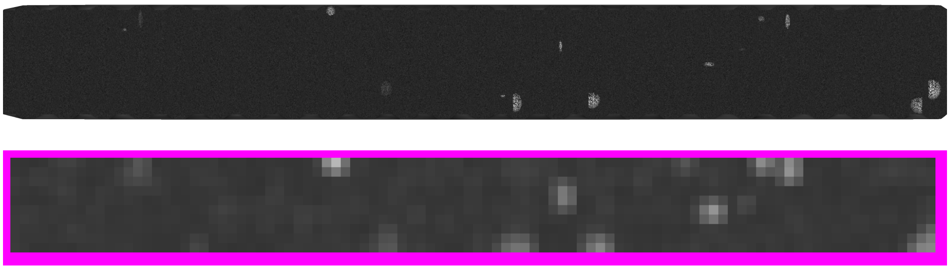

Figure 6. (**Top**) example of a simulated height map with defects applied. (**Bottom**) corresponding outlier activation scores that result from merging the scores of the tiles using the mean function.

4. Evaluation

We have evaluated our algorithms on our own synthetic datasets, which we created as explained in Section 3.1. The damages in this dataset are ellipsoidal in shape and the three axes of the ellipsoid are randomly sized between 5 and 50 cm.

4.1. Tile Score Merging Method

As discussed in Section 3.3, the outlier score computed for each individual tile needs to be merged into a non-overlapping grid. Two different functions were used. Their performance was compared on a dataset with known labels.

Figure 7 shows how the false positive and false negative rate relate to each other for varying chosen threshold values using the LOF approach. It can be seen that for any false negative rate, the false positive rate of the mean function is lower than for the min function. Lower false positive rates are of interest for this application, as a higher value may mean more manual work to check a larger number of candidates for damage that are, in fact, in a good state. Thus, the mean function is found to be superior.

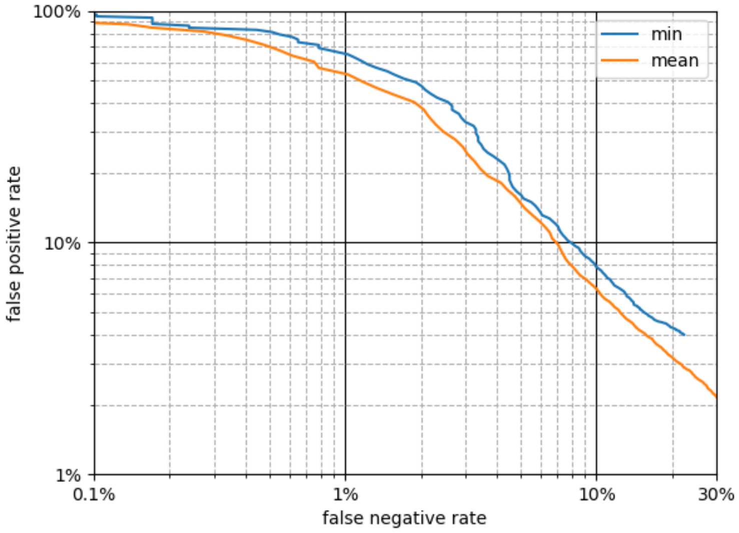

Figure 7. Comparison of tile score merge mode performance.

4.2. *Evaluation Method*

The size of a defect in our training data is typically larger than the size of one grid tile. In other words, a defect usually spreads over several grid tiles. It is important for our application that we can find all the defects, but we do not need to know the exact extent of each defect. Because of this, it is sufficient if only one (or more) of the tiles that a defect covers is classified as defective. Thus, for evaluating the usefulness of our algorithms regarding their intended application, we consider a defect as detected even if not all the tiles that it covers were classified as defective. Additionally, we ignore the border region of a labelled defect.

4.3. *Threshold Selection*

After merging the tile values, a grid of outlier values remains. The threshold required to detect these values into the Boolean labels *normal* and *abnormal* is not known a priori. It can be chosen arbitrarily to achieve a required maximum false negative rate. Figure 8 shows how the commonly used classification evaluation metrics are dependent on the threshold for the LOF approach. Outliers are assigned smaller numbers. Thus, with a larger threshold, more tiles will be classified abnormal. The curve for the recall shows that more actual defects can be detected. On the other hand, raising the threshold will decrease the precision, which means that many of the tiles that are classified defective are, in fact, in a good state. The graph for the accuracy is also worth noting. The dataset contains far more normal than abnormal examples, therefore, the accuracy can be close to 100% even when the recall is not that good. This metric is, thus, not suited to evaluating the performance of anomaly detection. Figure 9 shows the same metrics for the transfer learning approach. The graphs are generally lower, which shows that our transfer learning discriminator is inferior to our LOF discriminator.

Abnormal tiles have lower (more negative) scores than normal tiles. Thus, a larger threshold leads to a decrease in the recall (more defective examples found). However, at the same time, more false positive results are generated, which decreases the precision.

The primary goal with our application is to find most defects, i.e., a large recall. However, at the same time, the precision must be low enough to reduce the work that is required afterwards. Labelling everything as defective is useless. The threshold has to be selected to fulfil both conditions.

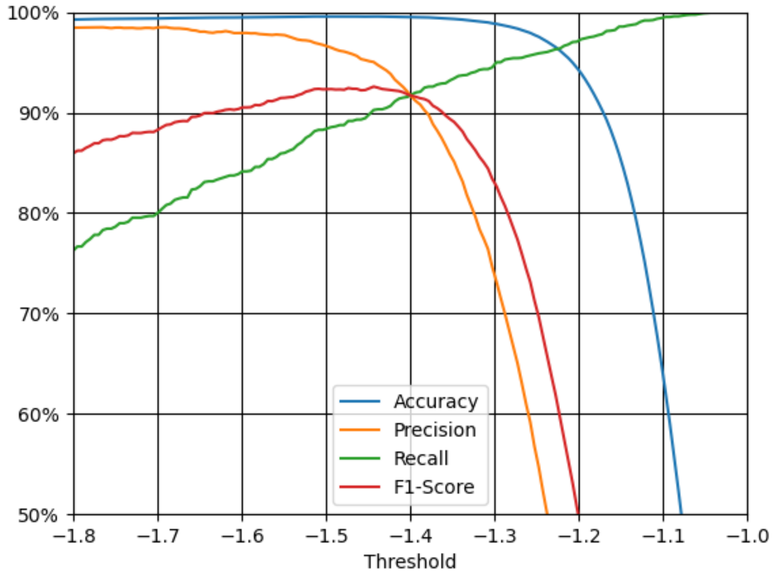

Figure 8. Dependence of evaluation metrics on threshold, LOF.

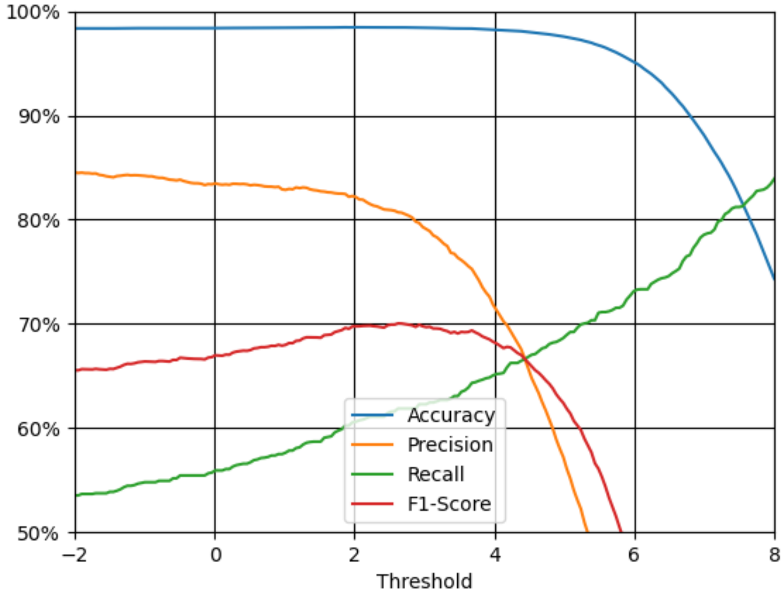

Figure 9. Dependence of evaluation metrics on threshold, Transfer Learning.

We were able to achieve much better results using the LOF approach, as can be seen when comparing the graphs of Figures 8 and 9. Thus, we will focus only on that from now on. If we chose a desired recall of 95%, we would achieve the following average results from our data as shown in the confusion matrix in Table 1 and evaluation metrics in Table 2:

Table 1. Confusion matrix for recall = 95% (threshold = −1.298).

		Actual Label	
		abnormal	normal
Predicted label	abnormal	2781	1017
	normal	146	95,820

Table 2. Evaluation metrics for recall = 95% (threshold = −1.298).

Accuracy	98.8%
Precision	73.2%
Recall	95.0%
F1-Score	82.7%

Figure 10 shows a few examples from our dataset with overlaid classification results. Correct classification of damage is coloured in green, yellow are false positives (classified damage where there is none) and red are false negatives (did not detect the damage). The magenta colouring shows the border areas that were excluded from our analysis.

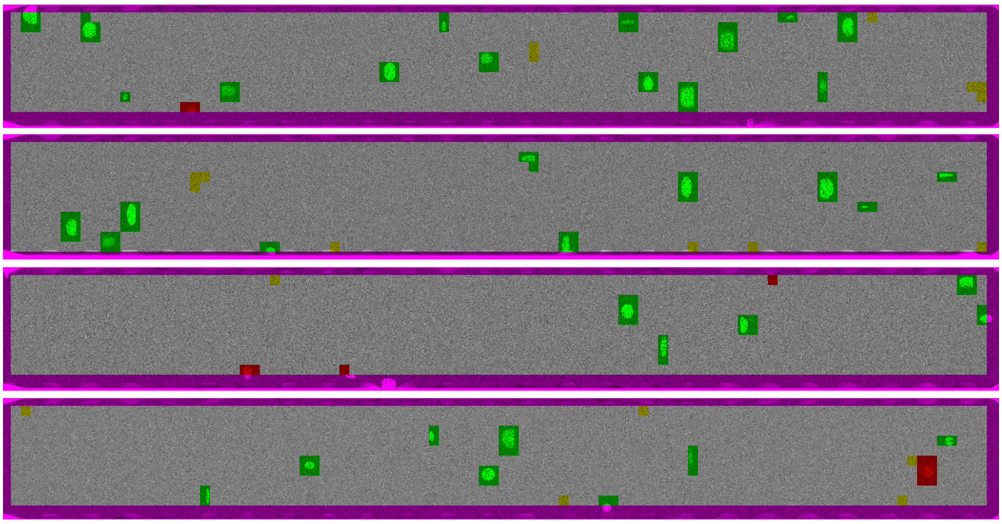

Figure 10. Examples of classification results on test data. Correct classification of damage is coloured in green, yellow are false positives (classified damage where there is none) and red are false negatives (did not detect the damage). The magenta colouring shows the border areas that were excluded from our analysis.

As can be seen from the examples, most damage that was not detected is close to the borders of the scan strips. They are covered by fewer tiles and, thus, have a lesser chance of being detected.

The reason for the rather worse results in transfer learning is possibly that the trained three-layer network is too small to detect all randomly sized damages.

5. Application to Real Data

Since the method appears fundamentally suitable for identifying potential damage areas when using simulated data, the next step is to analyse its application to the real dataset. The real data represents a quay wall above water. It is surveyed with a terrestrial laser scanner of type Z + F Imager 5016 in Lübeck city port, Germany on 13 sensor positions. We fused the 13 sensor positions on point clouds to achieve a small spacing between the points. The point spacing within the point clouds varies due to different scanning positions but is around 1 cm. The noise reaches 1–2 mm. Figure 11 shows a photo of the quay wall and the corresponding point cloud. There is obviously spalling in the upper part of the quay wall between the fenders. Two examples of damaged areas are shown in Figure 12. Both are spalling in the concrete. They are up to 1.5 m wide and reach a height of 50 cm.

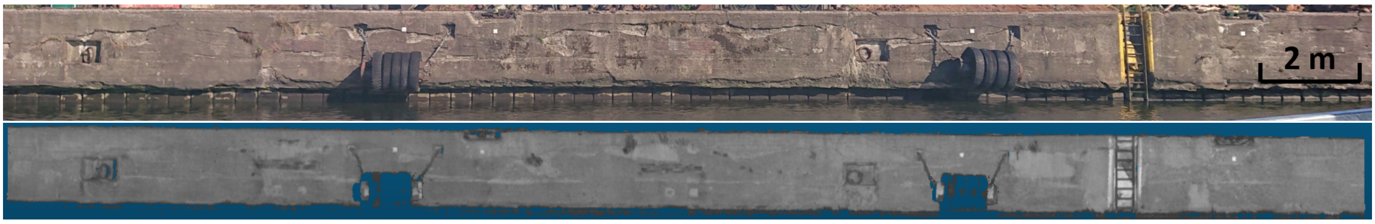

Figure 11. Real dataset in Lübeck city port. The upper row shows a photo of the quay wall and the lower row the corresponding point cloud. Spalling is in the upper part of the quay wall between the fenders.

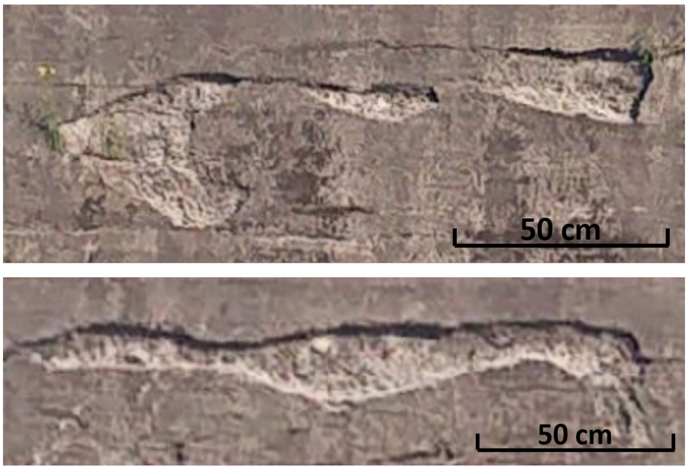

Figure 12. Two examples of damaged areas in the real dataset. Both pictures show the concrete spalling damage. The **upper** image shows three smaller damages with a size of up to 50 cm × 50 cm. The **lower** image show larger spalling damage, which is 1.5 m in width and 30 cm in height.

The point clouds were manually divided into three categories to generate ground truth: quay wall (blue), concrete spalling (green), and additional objects (red). This classification and corresponding depth and label image are shown in the top row of Figure 13.

Since the methodology performs much better with LOF, as can be seen in the comparison of Figures 8 and 9, we use only LOF for the real data. The average result can be seen in Figure 14. When comparing the average results from the simulated dataset (Figure 8) with the real dataset (Figure 14), it can be seen that the curves for accuracy, precision, recall, and F1 score show similar behaviour with a lower overall accuracy.

Again, a threshold is chosen, where precision and recall are essentially equal, which is a good compromise between true and false positives in an economic sense. The corresponding confusion matrix for the threshold of -1.55 selected can be seen in Table 3. As can be seen from the table, there is again a strong imbalance between the two classes. The number of false positives and false negatives is essentially the same.

Table 3. Confusion matrix for recall = 72.6% (threshold = -1.55).

		Actual Label	
		abnormal	normal
Predicted label	abnormal	9128	3437
	normal	3506	57,225

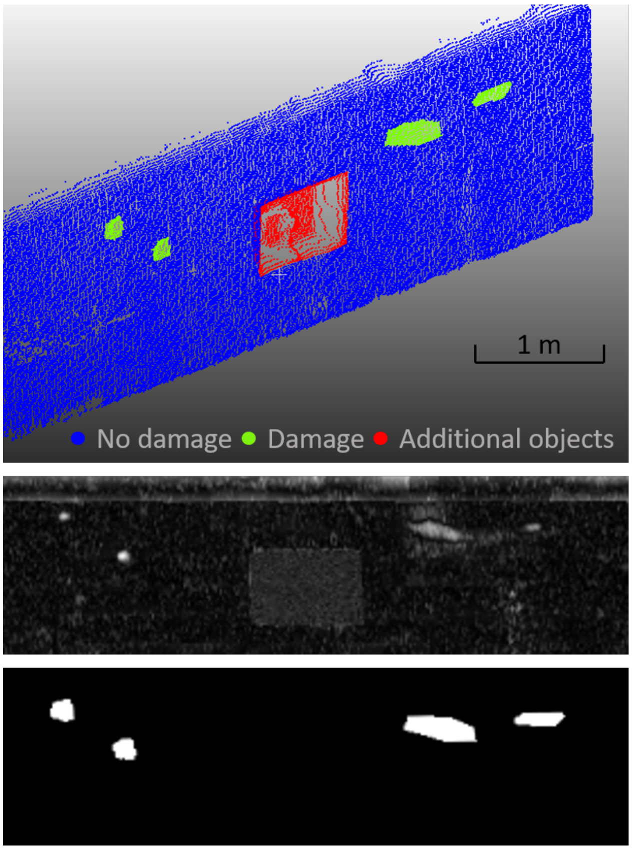

Figure 13. Real dataset with manually classified ground truth in the top row. No damage is in blue, concrete spalling is in green and additional objects, such as ladders or fenders, are in red. The middle row shows the corresponding height field in greyscale, with lighter areas indicating greater distances between the point cloud and the model. The bottom row shows the binary label image, with white indicating damage and black indicating no damage.

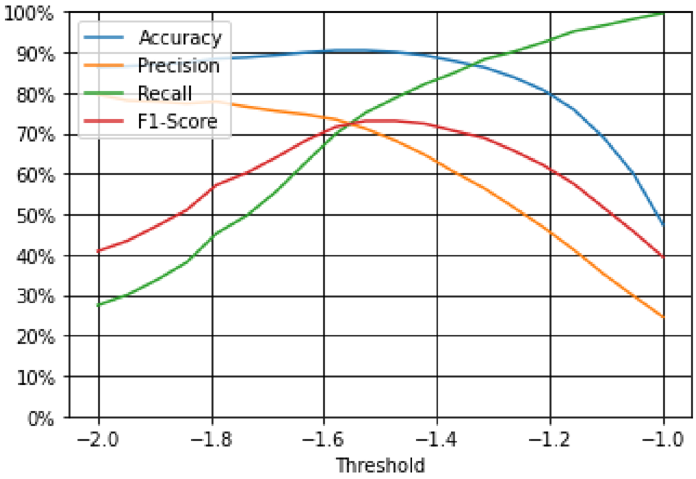

Figure 14. Dependence of evaluation metrics on the threshold, LOF for real data.

The evaluation metrics for the threshold of −1.55 selected are shown in Table 4. Accuracy reaches 90.5%. Precision and recall are with 72.2% and 72.6% mainly at the same level. This results in an average F1 score of 72.4%. This is still an indication of a good classification, but somewhat worse than in the simulated data.

Table 4. Evaluation metrics for recall = 72.6% (threshold = −1.55).

Accuracy	90.5%
Precision	72.2%
Recall	72.6%
F1-Score	72.4%

The classification result for two exemplary images is shown in Figure 15. Green, red and yellow indicate true positives, false negatives and false positives, respectively. Here, the original height field is shown with a higher contrast in the middle to make grey value differences in the height field more visible to the human eye. It can be seen in the top row of Figure 15 that all damages are detected and classified correctly. There are no false positives or false negatives. Only the two small damages at the bottom edge are not recognised, but this is because the edge areas are cut off during classification. The example in the bottom row shows a weaker classification result. Two damaged areas are not detected and, furthermore, two areas are falsely detected as damage.

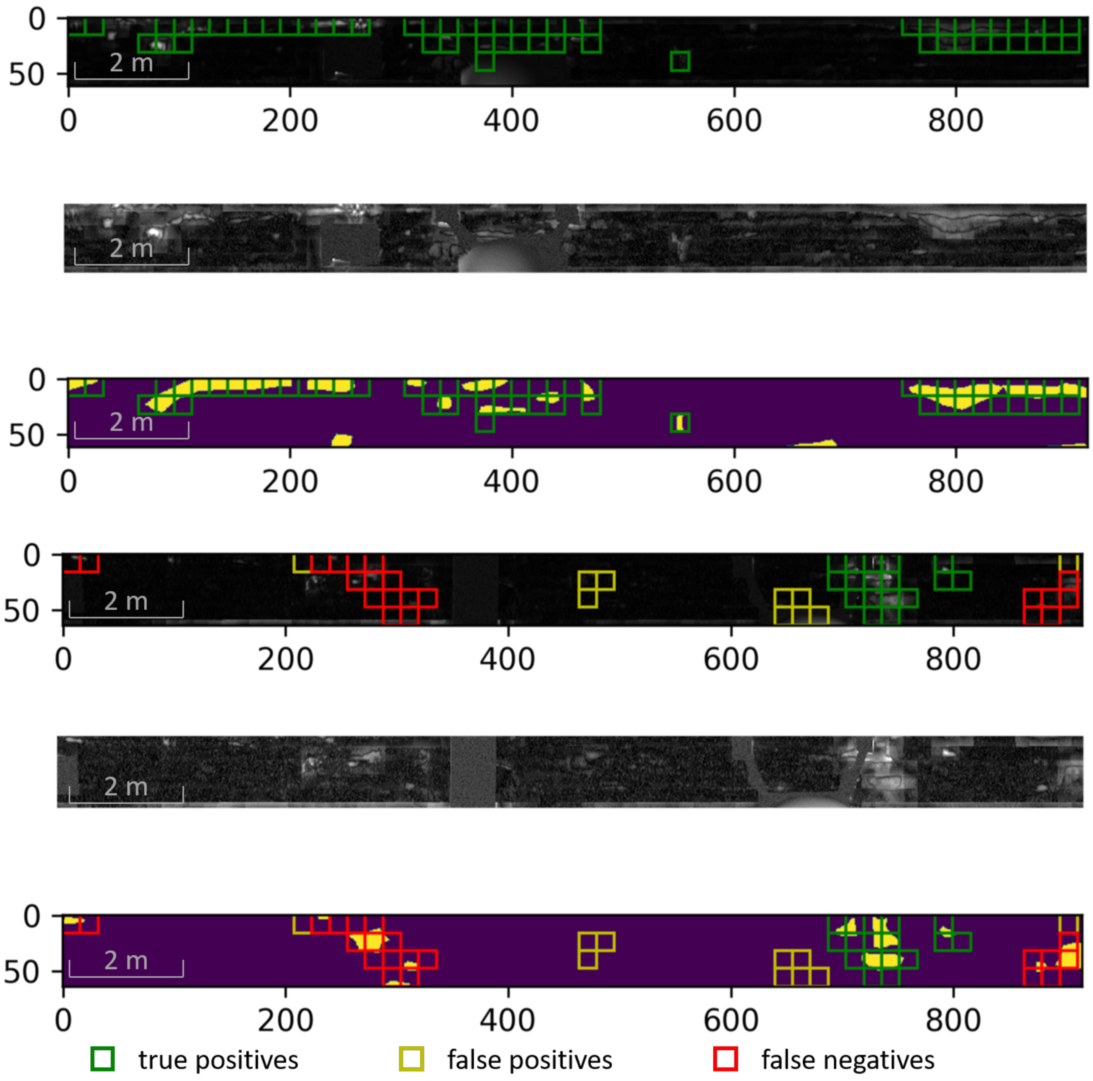

Figure 15. Two examples of damage detection using the real dataset from Lübeck city port. The **top** row in each case is the point cloud with overlaid damage detection. Green, red and yellow indicate true positives, false negatives and false positives, respectively. The **middle** row in each case is the height field and the **bottom** row is the label image with overlaid detected areas.

Therefore, the result is worse than for the simulated data (cf. Table 2). Nevertheless, the method seems to give good results when applied to real data.

6. Discussion

We assume several reasons for the different results. The first and probably the most important point is that there are other disturbing objects in the data, such as ladders, fenders, plants and ropes. The objects also lead to a higher distance in the height fields, which are currently not separable from higher distances based on real damage. The impact, particularly of plants, can be reduced by measurements in seasons with little vegetation, such as winter. The second point is that we do not clean or filter the data at the beginning. Only a rough manual cutting into the area of interest is carried out. The dataset still contains outliers and sensor artefacts that lead to false measurements. These artefacts are particularly strong where the structure comes into contact with the water. Therefore, the optimal time for the measurement is when the water level is as low as possible. In addition, the noise in the real data may not be normally distributed and still contain systematic components, unlike the simulated data. The threshold value for the separation into damaged and undamaged zones is chosen in the present study in such a way that a good trade-off between detected damage and actually correct classifications is achieved. The threshold may differ from another data set and has to be chosen again. In a very sensible and non-economic approach, one would choose a different threshold value that would give a higher recall.

The method presented is currently limited to geometrical damages, such as spalling and large cracking. The reason is that only 3D point clouds are used, and no colour information, which would be necessary to also detect small cracks and sintering. The method cannot detect damages smaller than the decimetre range because the point spacing and noise level underwater from a multibeam echo-sounder is much higher than a laser-scanned point cloud above water. Damages above water can be detected from the centimetre range due to the higher accuracy of the laser scanner and the smaller point spacing. Nevertheless, the results still contribute to automated damage detection and a digitally guided building inspections process.

The presented method gives similar results to other studies. Point CNN gives a mean interval above the unity score of 74.68% for bridge inspections with point cloud classification [37]. A combination of images and point clouds based on Otsu's algorithm for automatic concrete crack-detection achieves an average F1 score of 86.7% [38].

7. Conclusions and Outlook

The point clouds are converted into depth images and processed in a pretrained CNN with two extensions. Regarding the classification, firstly, an NN is attached to the CNN and, secondly, the LOF is calculated. Building inspection can be digitalised and taken to a completely new level with the method presented. We achieve a significantly higher completeness of the infrastructure inspections with the k-MSS used compared to the manual method with divers. We obtain a quality-controlled and reproducible mapping of the infrastructure by using laser scanners and hydrographic measurements. Suspected damage can be reliably detected and verified through the area-based measurement of the component surfaces above and below water. A comparison of different measurement epochs—as they have to be carried out every six years within the framework of the building inspection—is, thus, also possible for structures below water, so that the damage development and the service life of these economically important structures for our national economy can be better observed and evaluated in the future.

The procedure of transferring the features from natural images to point clouds and then performing a novel detection is totally new in the context of structural health monitoring systems. It is now possible for the first time to detect damage automatically.

The methodology presented is intended to automatically create a suspicion plan with suspected damage regions from point clouds. To be able to apply the methodology in reality, all damaged regions must be found as far as possible. Furthermore, only the damaged regions should be recognised as such. This means that the accuracy and the recall value together should be as high as possible. The methodology was first tested on simulated data and then applied to real data.

The analysis of the simulated data resulted in a very good classification with an F1 score of 96.3%. Concerning the requirements mentioned above, the method is suitable for creating suspicion plans of damage regions on quay walls. The result is slightly less effective for the real data. The F1 score is 72.4%. When looking at examples of damage in the data that has not been detected by our algorithm, it can be seen that most of them are at the border of the scanned depth map.

The results could be further improved by handling the edges separately, as those typically show a significantly different distribution compared to the rest of the scan. So far, we have only been able to test our algorithm on simulated data and one real dataset of a concrete quay wall. 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.

Author Contributions: Conceptualization, F.H., L.G. and H.A.; methodology, F.H., L.G. and H.A.; software, F.H. and L.G.; formal analysis, F.H., L.G. and H.A.; investigation, F.H., L.G. and H.A.; writing—original draft preparation, F.H., L.G. and H.A.; writing—review and editing, F.H., L.G., I.N. and H.A.; visualisation, F.H. and L.G.; supervision, H.A. and I.N. all authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Federal Ministry of Transport and Digital Infrastructure grant number 19H18011C. The publication of this article was funded by the Open Access Fund of the Leibniz University Hannover.

Supported by:

IHATEC
Innovative
Hafentechnologien

Federal Ministry
of Transport and
Digital Infrastructure

on the basis of a decision
by the German Bundestag

Acknowledgments: This work was carried out as part of the joint research project "3DHydroMapper–Bestandsdatenerfassung und modellgestützte Prüfung von Verkehrswasserbauwerken". It consists of five partners and one associated partner: Hesse und Partner Ingenieure (multisensor system and kinematic laser scanning), WK Consult (structural inspection, BIM and maintenance planning), Niedersachsen Ports (sea and inland port operation), Fraunhofer IGP (automatic modeling and BIM), Leibniz University Hannover (route planning and damage detection) and Wasserstraßen-und Schifffahrtsverwaltung des Bundes (management of federal waterways).

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

References

1. Hesse, C.; Holste, K.; Neumann, I.; Hake, F.; Alkhatib, H.; Geist, M.; Knaack, L.; Scharr, C. 3D HydroMapper: Automatisierte 3D-Bauwerksaufnahme und Schadens-erkennung unter Wasser für die Bauwerksinspektion und das Building Information Modelling. *Hydrogr. Nachr.-J. Appl. Hydrogr.* **2019**, *113*, 26–29.
2. Moisan, E.; Charbonnier, P.; Foucher, P.; Grussenmeyer, P.; Guillemin, S.; Samat, O.; Pages, C. Assessment of a static multibeam sonar scanner for 3D surveying in confined suqaquatic environments. In Proceedings of the XXIII ISPRS Congress, ISPRS, Prague, Czech Republic, 12–19 July 2016; Volume 41, pp. 541–548.
3. Robert, K.; Huvenne, V.A.; Georgiopoulou, A.; Jones, D.O.; Marsh, L.; DO Carter, G.; Chaumillon, L. New approaches to high-resolution mapping of marine vertical structures. *Sci. Rep.* **2017**, *7*, 1–14. [CrossRef] [PubMed]
4. Hadavandsiri, Z.; Lichti, D.D.; Jahraus, A.; Jarron, D. Concrete Preliminary Damage Inspection by Classification of Terrestrial Laser Scanner Point Clouds through Systematic Threshold Definition. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 585. [CrossRef]

5. Janowski, L.; Wroblewski, R.; Rucinska, M.; Kubowicz-Grajewska, A.; Tysiac, P. Automatic classification and mapping of the seabed using airborne LiDAR bathymetry. *Eng. Geol.* **2022**, *301*, 106615. [CrossRef]
6. Seo, H. Long-term Monitoring of zigzag-shaped concrete panel in retaining structure using laser scanning and analysis of influencing factors. *Opt. Lasers Eng.* **2021**, *139*, 106498. [CrossRef]
7. Aldosari, M.; Al-Rawabdeh, A.; Bullock, D.; Habib, A. A Mobile LiDAR for Monitoring Mechanically Stabilized Earth Walls with Textured Precast Concrete Panels. *Remote Sens.* **2020**, *12*, 306. [CrossRef]
8. O'Byrne, M.; Schoefs, F.; Ghosh, B.; Pakrashi, V. Texture analysis based damage detection of ageing infrastructural elements. *Comput.-Aided Civ. Infrastruct. Eng.* **2013**, *28*, 162–177. [CrossRef]
9. Gatys, L.; Ecker, A.S.; Bethge, M. Texture synthesis using convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 262–270.
10. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems 27*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., Eds.; Curran Associates, Inc.: San Francisco, CA, USA, 2014; pp. 3320–3328.
11. Carvalho, T.; de Rezende, E.R.S.; Alves, M.T.P.; Balieiro, F.K.C.; Sovat, R.B. Exposing computer generated images by eye's region classification via transfer learning of VGG19 CNN. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017; pp. 866–870.
12. Abati, D.; Porrello, A.; Calderara, S.; Cucchiara, R. 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.
13. Li, G.; Deng, C.; Wu, J.; Xu, X.; Shao, X.; Wang, Y. Sensor Data-Driven Bearing Fault Diagnosis Based on Deep Convolutional Neural Networks and S-Transform. *Sensors* **2019**, *19*, 2750. [CrossRef] [PubMed]
14. Grubbs, F.E. Procedures for Detecting Outlying Observations in Samples. *Technometrics* **1969**, *11*, 9005. [CrossRef]
15. Goldstein, M.; Uchida, S. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PLoS ONE* **2016**, *11*, e0152173. [CrossRef] [PubMed]
16. Andrews, J.; Tanay, T.; Morton, E.J.; Griffin, L.D. Transfer representation-learning for anomaly detection. In Proceedings of the 33rd International Conference on Machine Learning New York, NY, USA, 19–24 June 2016; JMLR: New York, NY, USA, 2016.
17. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 15–18 May 2000; pp. 93–104.
18. García-Teodoro, P.; Díaz-Verdejo, J.; Maciá-Fernández, G.; Vázquez, E. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Comput. Secur.* **2009**, *28*, 18–28. [CrossRef]
19. Li, S.H.; Yen, D.C.; Lu, W.H.; Wang, C. Identifying the signs of fraudulent accounts using data mining techniques. *Comput. Hum. Behav.* **2012**, *28*, 1002–1013. [CrossRef]
20. Lin, J.; Keogh, E.; Ada Fu.; Van Herle, H. Approximations to magic: Finding unusual medical time series. In Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05), Dublin, Ireland, 23–24 June 2005; pp. 329–334. [CrossRef]
21. Sigholm, J.; Raciti, M. Best-effort Data Leakage Prevention in inter-organizational tactical MANETs. In Proceedings of the MILCOM 2012–2012 IEEE Military Communications Conference, Orlando, FL, USA, 29 October–1 November 2012; pp. 1–7. [CrossRef]
22. Basharat, A.; Gritai, A.; Shah, M. Learning object motion patterns for anomaly detection and improved object detection. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8. [CrossRef]
23. Gebhardt, J.; Goldstein, M.; Shafait, F.; Dengel, A. Document Authentication Using Printing Technique Features and Unsupervised Anomaly Detection. In Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013; pp. 479–483. [CrossRef]
24. Skidmore, A.K. A comparison of techniques for calculating gradient and aspect from a gridded digital elevation model. *Int. J. Geogr. Inf. Syst.* **1989**, *3*, 323–334. [CrossRef]
25. Simony, M.; Milzy, S.; Amendey, K.; Gross, H.M. Complex-YOLO: An Euler-Region-Proposal for Real-time 3D Object Detection on Point Clouds. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
26. Haas, T.C. Kriging and automated variogram modeling within a moving window. *Atmos. Environ. Part A Gen. Top.* **1990**, *24*, 1759–1769. [CrossRef]
27. Drixler, E. Analyse der Form und Lage von Objekten im Raum. München, DGK Reihe C, Heft Nr. 409. Ph.D. Thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany 1993.
28. Sibson, R. A brief description of natural neighbour interpolation. In*Interpreting Multivariate Data*; Barnett, V., Ed.; John Wiley & Sons: Chichester, UK, 1981; pp. 21–36.
29. Wan, X.; Liu, L.; Wang, S.; Wang, Y. A Transfer Learning Strip Steel Surface Defect Recognition Network Based on VGG19. In *Advanced Manufacturing and Automation IX*; Lecture Notes in Electrical Engineering; Wang, Y., Martinsen, K., Yu, T., Wang, K., Eds.; Springer: Singapore, 2020; Volume 634, pp. 333–341.
30. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
31. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.

32. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In *Computer Vision–ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 818–833.
33. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the International Conference on Machine Learning (ICML), Haifa, Israel, 21–24 June 2010 .
34. Serra, J. *Image Analysis and Mathematical Morphology*; Academic Press, Inc.: Cambridge, MA, USA, 1983.
35. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
36. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
37. Kim, H.; Kim, C. Deep-Learning-Based Classification of Point Clouds for Bridge Inspection. *Remote Sens.* **2020**, *12*, 3757. [CrossRef]
38. Chen, X.; Li, J.; Huang, S.; Cui, H.; Liu, P.; Sun, Q. An Automatic Concrete Crack-Detection Method Fusing Point Clouds and Images Based on Improved Otsu's Algorithm. *Sensors* **2021**, *21*, 1581. [CrossRef] [PubMed]