Automated detection of micrometer-cracks and delamination in CT volumes of previously stressed CFRP pressure rods

Thomas Schromm$^{1,2}$, Jonas Holtmann$^{1,2}$, Michael Koch$^2$, Christian Große$^1$

$^1$Chair of Non-Destructive Testing, Technical University of Munich, Baumbachstr. 7, 81245 Munich, Germany, e-mail: zfp@cbm.bgu.tum.de
$^2$Technology Material and Process Analytics, BMW AG, Hufelandstraße 5, 80788 Munich, Germany, e-mail: thomas.schromm@bmw.de

Abstract

Industrial tasks that produce hundreds of terabytes of data per year require efficient evaluation tools for the purpose of saving valuable resources such as time or highly trained personnel. Therefore, processes with high automation potential need to be identified and put into practice. Before deploying a new tool, however, a thorough investigation is required. It needs to be tested whether the achievable degree of automation of such a new tool yields results which are on one hand reliable and on the other hand of sufficient quality. This work evaluates the applicability of artificial neural networks (ANNs) to detecting μm-cracks and delaminations in reconstructed computed tomography (CT) volumes of previously stressed carbon fibre reinforced polymer (CFRP) pressure rods. Common network architectures, varying network parameters and different training sets have been investigated and compared in order to determine the combination that performs best. The constellation (network, hyperparameters, training data) that performed best reached an average precision (AP) of 0.87. Based on the rather small data set of $\approx 1.6 \times 10^5$ images and the unstructured nature and great diversity of the investigated features, this result can be regarded as very good. The detected feature sizes ranged from $\approx 100 \mu m$ to a centimeter in length and from tens of microns to a few hundred microns in width. The results suggest that artificial neural networks have the potential to be used reliably for the automatic detection of μm-cracks and delaminations in CT volumes of CFRP pressure rods, provided the data set used for training is large and diverse enough and the network is being updated when new data is available.

Keywords: Automation, machine learning, neural networks, non-destructive testing, carbon fibre reinforced polymers, computed tomography, feature detection

1 Introduction

Vehicle weight reduction is an imperative aspect, especially when aiming for an energy and cost efficient electromobility service. This is due to the mass related energy density$^1$ of Li-ion batteries$^2$ being considerably lower than the one of diesel fuel$^3$ or gasoline$^4$. In order to increase the cruising range, the battery’s storage capacity must be increased. This implies, however, an increase in total vehicle weight which in turn reduces the possible cruising range. To compensate the increase in weight heavy metallic load-bearing components, which amount for up to 40%$^1$ of the vehicle’s mass, can be replaced with strong yet light components made of composite materials such as carbon fibre reinforced polymers (CFRPs). Despite its lower density, CFRP outperforms steel in terms of tensile elastic modulus and tensile strength$^2$. Nevertheless, these hybrid structures must be thoroughly tested before incorporating them in load-bearing components. In the scope of a former project$^3$ CFRP pressure rods (see Figure 1) were produced, stressed and scanned with Computed Tomography (CT) in order to analyze their volumetric deformation behaviour under pressure. The project focused on different failure modes like forming of cracks and delamination. Finding these features in CT volumes can be very time consuming, since they are often very small (micrometers) compared to the size of the the sample (millimeters to centimeters).

Figure 1: Top: Exemplary CFRP pressure rod with dimensions $140 \, mm \times 11 \, mm \times 4 \, mm$. Adapted from [31].

In the last decade, machine learning (ML) algorithms have been used for automating various 2D analysis tasks ranging from feature detection and style transfer to image reconstruction (see for example [26]). Especially artificial neural networks (ANNs) have proven to be very effective for image related tasks like feature detection and classification (see for example [25]). In this case, the respective energy storage technologies are compared only qualitatively on the basis of mass related energy densities.

$^1$This is not the full story when comparing energy storage technologies quantitatively, according to [24]. This work, however, does not focus on this discussion.

$^2$0.32 – 0.85 MJ kg$^{-1}$ according to [27] and 0.36 – 0.95 MJ kg$^{-1}$ according to [28]

$^3$43 MJ kg$^{-1}$ according to [29]

$^4$40 – 42 MJ kg$^{-1}$ according to [30]
work we compare the performance of differently set up ANNs regarding their ability to detect \( \mu m \)-cracks and delamination in CT volumes of previously stressed carbon fibre reinforced polymer pressure rods.

## 2 Methods

### 2.1 Computed Tomography

Although we tested our algorithms exclusively on CT data it is in general possible to train a network on data from other non-destructive testing (NDT) imaging methods, such as eddy current, ultrasound or thermography (e.g. [5], [6], [7], respectively). These imaging methods, however, cannot always be applied successfully due to their small penetration depth or incapability to provide volumetric information. X-rays on the other hand, with energies still well below the relevant photodisintegration\(^5\) thresholds (e.g. 1.67 MeV for the stable isotope beryllium-9 [33]) can penetrate even larger car structures made of either CFRP or metal well enough to reconstruct highly resolved volumes — without inducing radioactive reactions. Exemplary CT cross sections from an analyzed CFRP sample can be seen in Figure 2. The applied scan parameters are summarized in Table 1.

![Image](image-url)

**Figure 2:** Exemplary CT cross sections from a previously stressed CFRP pressure rod. Subtle cracks and delaminations can be seen in the following cross sections: top-center, top-right and bottom-left.

<table>
<thead>
<tr>
<th>( I ) (( \mu A ))</th>
<th>( U ) (kV)</th>
<th>( SDD ) (mm)</th>
<th>( SOD ) (mm)</th>
<th>( M_g )</th>
<th>( S_{VOX} ) (( \mu m^3 ))</th>
<th># Images over 360(^\circ)</th>
<th># Avrgd.</th>
<th>Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>140</td>
<td>816</td>
<td>58</td>
<td>14</td>
<td>14</td>
<td>2000</td>
<td>3</td>
<td>0.25 mm of Cu</td>
</tr>
</tbody>
</table>

### 2.2 Convolutional Neural Networks

Machine Learning algorithms can be any kind of algorithms that provide the ability to learn without being explicitly programmed [23]. This holds true for artificial neural networks (ANNs) like convolutional neural networks (CNNs) or fully connected networks (FCNs). The aim of ANNs is to find a function \( f(\cdot) \) that maps a set \( \Gamma \) onto another set \( \Omega \) (see Figure 3b). The applicability of ANNs to images and other forms of data, is based on the assumption that multiple non-linear layers can approximate complicated functions [10]. A neural network is called deep neural network when it has multiple hidden layers between the input and output layer. The cornerstones of each network are nodes, biases, weights and activation functions. Nodes and weights can be seen in Figure 3b. Activation functions and biases are not shown for matters of readability.

A common type of learning is supervised learning. Here the algorithm is fed with a large number of input pairs, each of which consist of raw data like an image or the selected region of an image and an associated label. The pairs are subsequently passed through the network and are used to minimize a cost function. The cost function quantifies the error between predicted labels (output) and expected labels (input) and presents it in the form of a single real number [22]. The error is then propagated backwards through the network. This backpropagation, also known as learning phase, adapts the weights via gradient descent or stochastic gradient descent until the error is minimized. [15]

\(^5\)Nuclear process in which an atomic nucleus absorbs a high-energy gamma ray, and decays emitting a subatomic particle.
The computational cost of fully connected networks (Figure 3b) increases rapidly with each added layer and input dimension. As a consequence the size of the input images needs to be successively reduced before reaching the fully connected layers. This can be achieved by means of convolution \(^6\) or pooling. In Figure 3a the operation in Layer 1 illustrates a convolution and the operation in Layer 2 exemplifies max-pooling in which only the maximum of three pixels will be passed to the next layer. The initial number of \(12 \times 9\) pixels is thereby reduced to only \(4 \times 1\). This means automatically that the contained information of the initial 108 pixels was condensed to the 4 remaining pixel values. The resulting number of input nodes is now small enough to pass their content on to a fully connected layer as can be seen in Figure 3b.

Figure 3: (a) Schematic illustration of a CNN. In this example a \(3 \times 3\) Kernel averages 9 pixels from Layer 1 and passes the result to Layer 2. There, a pooling operation takes place. (b) Fully connected Layers. Each node is connected with every node of the following layer. For matters of readability activation functions and biases are not shown, as well as only the top three connections with their respective weights of 0.3, 0.8 and 0.2.

2.3 Training

All 1621 cross sections (examples shown in Figure 2) were divided into three sub-sets: Training (60%), validation (20%) and test (20%). The performance of the networks is evaluated with the independent test data set, as soon as the validation cost function settled down or reached a small enough value. This usually happened around \(10^2 - 10^3\) steps (\(10 - 10^2\) epochs\(^7\)). This way networks that were trained on the same data but used different hyperparameters can be compared.

Training a network from scratch, however, is computationally expensive since all weights are randomly distributed at first. Reaching a certain level of generalization from this starting point therefore takes time. This is why transfer learning \([20, 21]\) is applied in many cases: An already trained network with weights forming certain abstract filters (edges, lines, etc.) is used as the foundation (feature extractor). Only the weights in some layers at the end of this network are replaced by randomly distributed weights. This allows for a case-specific fine-tuning based on the new data set from which the network is supposed to learn. All used networks are based exclusively on the COCO (common objects in context) data set, which was chosen over ImageNet due to the fact that COCO has more instances per category. According to \([17]\), this can aid in learning detailed object models which are capable of precise 2D localization. However, this was not further investigated in the scope of this work. Networks which are trained on the COCO \([16, 17]\) data set and the ImageNet \([18, 19]\) data set are publicly available.

The performance of different network architectures as well as different training sets was compared. ResNet \([14]\), MobileNet \([13]\) and Inception ResNet v2 \([12]\), which are pre-trained feature extractors, were utilized for this comparison as the networks’ backbones. As object detectors we used RetinaNet and Faster R-CNN (region-based convolutional neural network) \([11]\).

Learning rates which are too high can possibly cause spikes in the loss curves\(^8\) \([9]\). Learning rates which are too low, on the other hand, can slow down the learning process and/or can cause the cost function to get stuck at a local minimum. Most modern deep learning libraries, like Keras, adapt the learning rate in order to prevent both \([8]\). It was therefore investigated whether or not a decrease of the learning rates by a factor of 0.01 would impact the network’s performance. In order to minimize the influence of the arrangement of the training sets A,B and E, their content was shuffled for each epoch.

We used the average precision (AP) measure to compare the different network constellations. The AP is defined as the area under

\(^6\)There are also convolutional operations which do not reduce the input layer’s size. In this case they act merely as a filter which preserves dimensionality.

\(^7\)One epoch is define as one full training cycle, i.e. the network has seen the whole training set one time.

\(^8\)The loss curves represent the chronological sequence of the cost function’s output over all epochs.
a precision-recall-curve (PR-curve, see Figure 4a). The PR-curve in turn indicates how accurate the performance is depending on how many positives there are. Hence it is a measure of correctness versus completeness:

\[
\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}}
\]

\[
\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}
\]

To investigate the distribution of features among all images, we shuffled the sets five times, as can be seen in Figure 4b, and trained a network. This is called k-fold cross-validation and is used to discover biased distributions of features contained in the data set.

Figure 4: (a) Exemplary precision-recall-curve (PR-curve). It tells you how often the network is correct depending on the number of all relevant features that were found. A recall of “1” means all relevant features were found. (b) Principle of k-fold cross validation: Each row represents one training. The constellations that make up training, validation, and test set are never the same.

There are countless combinations of parameter settings one can use for a CT scan. It would be too time and cost intensive to scan samples over a wide range of parameters and orientations. Instead, to quantify their influence on the network’s performance and to increase the data set, the concept of data augmentation was employed. The different constraints can be seen in Table 2.

<table>
<thead>
<tr>
<th>Brightness</th>
<th>Contrast</th>
<th>Rotation</th>
<th>Translation in [%]</th>
<th>Shear in [%]</th>
<th>Scaling in [%]</th>
<th>Prob. to flip</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b \in [-0.2, 0.2] )</td>
<td>( c \in [0.8, 1.2] )</td>
<td>( \alpha \in [-\frac{\pi}{10}, \frac{\pi}{10}] )</td>
<td>( tx,y \in [-10, 10] )</td>
<td>( p \in [-10, 10] )</td>
<td>( sx,y \in [-10, 10] )</td>
<td>( fx,y = 0.5 )</td>
</tr>
</tbody>
</table>

3 Results

3.1 Decreased learning rates

We found that in our case a decrease in learning rate does reduce the number and intensity of spikes in the loss functions but does not improve the overall performance of the network. The subsiding of spikes, can be explained with the fact that smaller steps on the surface of the cost function are less prone to large and abrupt local changes. The average precisions were 0.68 and 0.75 for the lower training rate and the original rate, respectively.

3.2 Cross-validation

As described above in subsection 2.3, we performed a five-fold cross validation in order to ensure a homogeneous distribution of features in the complete data set (Subsets A - E). Table 3 shows the results of the trainings as well as the number of epochs after which the trainings were considered most successful. The similarity of the average precisions (APs) shows that there are no strong outliers which could disrupt a steady performance of the network.
Table 3: Results of the cross validation. Average precision AP, classification and localization loss, \( L_c \) and \( L_l \). Adapted from [3].

<table>
<thead>
<tr>
<th>Subsets used for training</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>ABE</th>
<th>ADE</th>
<th>CDE</th>
<th>BCD</th>
<th>mean ± std</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>0.77</td>
<td>0.75</td>
<td>0.80</td>
<td>0.75</td>
<td>0.84</td>
<td>0.78 ± 0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( L_c )</td>
<td>0.97</td>
<td>0.92</td>
<td>1.00</td>
<td>1.09</td>
<td>0.86</td>
<td>0.97 ± 0.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( L_l )</td>
<td>0.19</td>
<td>0.17</td>
<td>0.17</td>
<td>0.17</td>
<td>0.18</td>
<td>0.18 ± 0.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steps</td>
<td>933</td>
<td>1302</td>
<td>1509</td>
<td>1308</td>
<td>2673</td>
<td>1545 ± 664.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Epochs</td>
<td>38</td>
<td>53</td>
<td>61</td>
<td>53</td>
<td>107</td>
<td>62.4 ± 26.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.3 Impact of data set size
As stated in subsection 3.2, the influence of the selected original training data on the final network performance is marginal (Table 3 and AD1 in Table 4). In addition, we investigated the impact of the data set’s size on the network performance. The networks performance increased when only 770 images were used, compared to 1621 (AD1 and AD2, respectively). This slight improvement might stem from the fact that fewer, yet similar data requires less features to learn, or respectively, less abstractions need to be made. It suggests, that the smaller data set suffices for training. However, since this only holds true for this rather small and exclusive data set, one cannot consider it to be representative for all possible features that can occur. What’s more is that this slight increase in average precision almost vanishes when additional geometric augmentation (AD6 and AD7) is applied. This shows, that the data augmentation has a larger impact on the performance than the reduction of data size. All in all, the observed results in Table 4 (AD1-10) can predominantly be attributed to the selected network architectures and whether the original data was additionally augmented or not. Now that the influence of the data set selection has proven to be marginal on the networks performance we continue to compare the different network architectures mentioned above. For the remaining comparisons the subsets A,B and E were randomly chosen as training data and C and D as validation and test data, respectively.

3.4 Impact of different architectures and data sets
The ResNet50 and ResNet101 backbones yielded similar results. However, in this case ResNet101 is considered inferior to ResNet50 since ResNet101 has twice as many hidden layers than ResNet50 and is therefore computationally more expensive. The MobileNet backbone (AD9) performed poor when compared to the other architectures. It took the networks with a one-stage RetinaNet as object detector approximately 50 seconds to evaluate 600 images. The two-stage approach with the object detector Faster R-CNN on the other hand needed approximately 65 seconds. The slightly better AP value of 0.87 (AD10) however compensates for the slightly longer evaluation times. The two-stage approach (AD10) therefore outperformed the other one-stage ensembles (AD1-9).

Table 4: Overview of the investigated network architecture and data set combinations (AD1-10). Adapted from [3].

<table>
<thead>
<tr>
<th>Object Detector</th>
<th>Feature Extractor (backbone)</th>
<th>AP</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet50 (original 1621 images)</td>
<td>0.78</td>
<td>AD1</td>
<td></td>
</tr>
<tr>
<td>ResNet50 (restricted to 770)</td>
<td>0.79</td>
<td>AD2</td>
<td></td>
</tr>
<tr>
<td>ResNet50 (augmented images)</td>
<td>0.81</td>
<td>AD3</td>
<td></td>
</tr>
<tr>
<td>- Contrast</td>
<td>0.84</td>
<td>AD4</td>
<td></td>
</tr>
<tr>
<td>- Brightness</td>
<td>0.83</td>
<td>AD5</td>
<td></td>
</tr>
<tr>
<td>- Geometry</td>
<td>0.83</td>
<td>AD6</td>
<td></td>
</tr>
<tr>
<td>ResNet50 (compl. &amp; augm.)</td>
<td>0.83</td>
<td>AD7</td>
<td></td>
</tr>
<tr>
<td>ResNet50 (restr. &amp; augm.)</td>
<td>0.82</td>
<td>AD8</td>
<td></td>
</tr>
<tr>
<td>ResNet101</td>
<td>0.79</td>
<td>AD9</td>
<td></td>
</tr>
<tr>
<td>MobileNet</td>
<td>0.68</td>
<td>AD10</td>
<td></td>
</tr>
</tbody>
</table>
3.5 Test on further samples

After the best performing network (Faster R-CNN with an Inception ResNet v2 backbone, see AD10 in Table 4) was identified, it was qualitatively tested on three new pressure rods. The reconstructed volumes of the three samples with predicted bounding boxes are illustrated in Figure 5. Cross sections of the new samples with predicted bounding boxes are shown in Figure 6. As can be seen in Sample 2 (Figure 6b), some features could not be identified successfully. The network might have failed to perform well in this case for two reasons: 1) The predefined aspect ratios of the bounding boxes of 0.1, 0.5, 1.0, 2 and 5 used by the network. Using a larger set of possible aspect ratios might improve the performance. Especially long and thin bounding boxes with an aspect ratio smaller than 0.1 might achieve the required results. However, this will increase the computational time. 2) The right cross section of Sample 2 in Figure 6b contains a rather broad feature, which appear rather infrequently compared to the other formations. Here the features’ stratification [4] in the training data needs to be analysed in order to gather more such examples on which the network can be trained.

(a) Sample 1 – Top view  
(b) Sample 1 – Top view  
(c) Sample 2 – Top view  
(d) Sample 3 – Side view

Figure 5: The trained network was applied to data sets of three new samples (Sample 1-3). Cracks and/or delaminations are emphasized. This can be seen clearly in (a) and (b) where the same reconstructed volume of Sample 1 is shown with and without bounding boxes, respectively. The progression of the cracks and/or delamination is clearly visible in all three samples (b-d). The two colors (blue, red) of the bounding boxes have no special meaning in this case.

(a) Sample 1  
(b) Sample 2  
(c) Sample 3

Figure 6: CT cross sections from three new pressure rods. (a) Almost all features were detected. (b) Here only parts of the features of interest were detected. (c) Almost all features were detected.

A simple graphical user interface (GUI) was developed in order to quantify the network’s performance on the new data set obtained from Sample 1. In total 1017 cross sections from Sample 1 with predicted bounding boxes were analyzed by three examiners individually. It took them approximately 1.5 hours each to evaluate the 1017 examples. In the GUI the examiners had the choice to mark entire cross sections either as “correct” or “incorrect”. Their choice depended only on their subjective judgement on whether the network detected all features in a cross section or not. 149 (15%) cross sections were marked as “incorrect” by all three examiners. In 241 (24%) cases they did not reach the same conclusion, meaning two examiners marked a cross section as “correct” while one of them marked it as “incorrect” or vice versa. The amount of correctly identified cross sections is therefore 627 (61%).


4 Conclusion and outlook

In order to increase the efficiency of object detection in reconstructed CT volumes of stressed CFRP pressure rods, machine algorithms were employed and their feasibility for an automated evaluation was quantified. The performance of three feature extractors (ResNet, MobileNet and Inception ResNet v2) and two object detectors on top (RetinaNet and Faster R-CNNs) was compared. In addition, the influence of the data (distribution, size) was analyzed as well as the impact of the learning rate. The following conclusions, which only hold true for the presented data and network constellations, were reached:

• Using machine learning algorithms for object detection in CT volumes can increase the efficiency of the evaluation process considerably.

• The best performing network (Faster R-CNN object detector with a Inception ResNet v2 backbone as feature extractor) managed to evaluate approximately 600 images/min with an AP of 0.87. This is considered very good, however, there is still room for improvement for which much more diverse data is required.

• The 1621 cross sections that were used for training, evaluation and testing, present a rather homogeneously distributed data set regarding the structural behaviour of features. However, as was found in subsection 3.5, the presence of infrequently appearing examples will degrade the network’s performance. Therefore the stratification of characteristics needs to be investigated and the data set needs to be extended accordingly. The 149 cross sections with features that were not found by the network in subsection 3.5 need to be added to the training data set for that purpose.

The 2D bounding boxes can be superimposed on the original reconstructed CT volume (Figure 5). This way the progression and distribution of cracks and delaminations can be analyzed a lot easier. However, there is potential to optimize the evaluation process even further:

• Two class differentiation. This work used only one class for both cracks and delamination. In a next step, it would be desirable to have a network which can discern cracks from delaminations. This would allow for a more complete analysis tool.

• Pixel-wise segmentation. Instead of highlighting the features with bounding boxes, a network could be trained on pixel-wise labeled data. This way, only the cracks and delaminations would be highlighted.

These improvements require much more data from which the network can learn. This in turn increases the computational cost as well as the amount of time spent on the labeling process. Despite these challenges and by taking into account that they pose as a one-time effort, we are optimistic about the magnitude of the resulting benefits.

References


