Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders

Paul Bergmann\(^1\), Sindy Löwe\(^1,2\), Michael Fauser\(^1\), David Sattlegger\(^1\) and Carsten Steger\(^1\)

\(^1\)MVTec Software GmbH, Germany
\(^2\)University of Amsterdam, The Netherlands
{bergmannp, fauser, sattlegger, steger}@mvtec.com, sindy.lowe@student.uva.nl

Keywords: Unsupervised Learning, Anomaly Detection, Defect Segmentation.

Abstract: Convolutional autoencoders have emerged as popular methods for unsupervised defect segmentation on image data. Most commonly, this task is performed by thresholding a per-pixel reconstruction error based on an \(\ell^p\)-distance. This procedure, however, leads to large residuals whenever the reconstruction includes slight localization inaccuracies around edges. It also fails to reveal defective regions that have been visually altered when intensity values stay roughly consistent. We show that these problems prevent these approaches from being applied to complex real-world scenarios and that they cannot be easily avoided by employing more elaborate architectures such as variational or feature matching autoencoders. We propose to use a perceptual loss function based on structural similarity that examines inter-dependencies between local image regions, taking into account luminance, contrast, and structural information, instead of simply comparing single pixel values. It achieves significant performance gains on a challenging real-world dataset of nanofibrous materials and a novel dataset of two woven fabrics over state-of-the-art approaches for unsupervised defect segmentation that use per-pixel reconstruction error metrics.

1 INTRODUCTION

Visual inspection is essential in industrial manufacturing to ensure high production quality and high cost efficiency by quickly discarding defective parts. Since manual inspection by humans is slow, expensive, and error-prone, the use of fully automated computer vision systems is becoming increasingly popular. Supervised methods, where the system learns how to segment defective regions by training on both defective and non-defective samples, are commonly used. However, they involve a large effort to annotate data and all possible defect types need to be known beforehand. Furthermore, in some production processes, the scrap rate might be too small to produce a sufficient number of defective samples for training, especially for data-hungry deep learning models.

In this work, we focus on unsupervised defect segmentation for visual inspection. The goal is to segment defective regions in images after having trained exclusively on non-defective samples. It has been shown that architectures based on convolutional neural networks (CNNs) such as autoencoders (Goodfellow et al., 2016) or generative adversarial networks (GANs) (Goodfellow et al., 2014) can be used for this task. We provide a brief overview of such methods in Section 2. These models try to reconstruct their inputs in the presence of certain constraints such as a bottleneck and thereby manage to capture the essence of high-dimensional data (e.g., images) in a lower-dimensional space. It is assumed that anomalies in the test data deviate from the training data manifold and the model is unable to reproduce them. As a result, large reconstruction errors indicate defects. Typically, the error measure that is employed is a per-pixel \(\ell^p\)-distance, which is an ad-hoc choice made for the sake of simplicity and speed. However, these measures yield high residuals in locations where the reconstruction is only slightly inaccurate, e.g., due to small localization imprecisions of edges. They also fail to detect structural differences between the input and reconstructed images when the respective pixels’ color values are roughly consistent. We show that this limits the usefulness of such methods when employed in complex real-world scenarios.

To alleviate the aforementioned problems, we propose to measure reconstruction accuracy using the structural similarity (SSIM) metric (Wang et al., 2004). SSIM is a distance measure designed to capture perceptual similarity that is less sensitive to edge...
alignment and gives importance to salient differences between input and reconstruction. It captures interdependencies between local pixel regions that are disregarded by the current state-of-the-art unsupervised defect segmentation methods based on autoencoders with per-pixel losses. We evaluate the performance gains obtained by employing SSIM as a loss function on two real-world industrial inspection datasets and demonstrate significant performance gains over per-pixel approaches. Figure 1 demonstrates the advantage of perceptual loss functions over a per-pixel $\ell^2$-loss on the NanoTWICE dataset of nanofibrous materials (Carrera et al., 2017). While both autoencoders alter the reconstruction in defective regions, only the residual map of the SSIM autoencoder allows a segmentation of these areas. By changing the loss function and otherwise keeping the same autoencoding architecture, we reach a performance that is on par with other state-of-the-art unsupervised defect segmentation approaches that rely on additional model priors such as handcrafted features or pretrained networks.

2 RELATED WORK

Detecting anomalies that deviate from the training data has been a long-standing problem in machine learning. Pimentel et al. (Pimentel et al., 2014) give a comprehensive overview of the field. In computer vision, one needs to distinguish between two variants of this task. First, there is the classification scenario, where novel samples appear as entirely different object classes that should be predicted as outliers. Second, there is a scenario where anomalies manifest themselves in subtle deviations from otherwise known structures and a segmentation of these deviations is desired. For the classification problem, a number of approaches have been proposed (Perera and Patel, 2018; Sabokrou et al., 2018). Here, we limit ourselves to an overview of methods that attempt to tackle the latter problem.

(Napoletano et al., 2018) extract features from a CNN that has been pretrained on a classification task. The features are clustered in a dictionary during training and anomalous structures are identified when the extracted features strongly deviate from the learned cluster centers. General applicability of this approach is not guaranteed since the pretrained network might not extract useful features for the new task at hand and it is unclear which features of the network should be selected for clustering. The results achieved with this method are the current state-of-the-art on the NanoTWICE dataset, which we also use in our experiments. They improve upon previous results by (Carrera et al., 2017), who build a dictionary that yields a sparse representation of the normal data. Similar approaches using sparse representations for novelty detection are (Boracchi et al., 2014; Carrera et al., 2015; Carrera et al., 2016).

(Schlegl et al., 2017) train a GAN on optical coherence tomography images of the retina and detect anomalies such as retinal fluid by searching for a latent sample that minimizes the per-pixel $\ell^2$-reconstruction error as well as a discriminator loss. The large number of optimization steps that must be performed to find a suitable latent sample makes this approach very slow. Therefore, it is only useful in applications that are not time-critical. Recently, (Zenati et al., 2018) proposed to use bidirectional GANs (Donahue et al., 2017) to add the missing encoder network for faster inference.
However, GANs are prone to run into mode collapse, i.e., there is no guarantee that all modes of the distribution of non-defective images are captured by the model. Furthermore, they are more difficult to train than autoencoders since the loss function of the adversarial training typically cannot be trained to convergence (Arjovsky and Bottou, 2017). Instead, the training results must be judged manually after regular optimization intervals.

(Baur et al., 2018) propose a framework for defect segmentation using autoencoding architectures and a per-pixel error metric based on the $\ell^1$-distance. To prevent the disadvantages of their loss function, they improve the reconstruction quality by aligning input data and adding an adversarial loss to enhance the visual quality of the reconstructed images. However, for many applications that work on unstructured data, prior alignment is impossible. Furthermore, optimizing for an additional adversarial loss during training but simply segmenting defects based on per-pixel comparisons during evaluation might lead to worse results since it is unclear how the adversarial training influences the reconstruction.

Other approaches take into account the structure of the latent space of variational autoencoders (VAEs) (Kingma and Welling, 2014) in order to detect outlier defects. (An and Cho, 2015) define a reconstruction probability for every image pixel by drawing multiple samples from the estimated encoding distribution and measuring the variability of the decoded outputs. (Soukup and Pinetz, 2018) disregard the decoder output entirely and instead compute the KL divergence as a novelty measure between the prior and the encoder distribution. This is based on the assumption that defective inputs will manifest themselves in mean and variance values that are very different from those of the prior. Similarly, (Vasilev et al., 2018) define multiple novelty measures, either by purely considering latent space behavior or by combining these measures with per-pixel reconstruction losses. They obtain a single scalar value that indicates an anomaly, which can quickly become a performance bottleneck in a segmentation scenario where a separate forward pass would be required for each image pixel to obtain an accurate segmentation result. We show that per-pixel reconstruction probabilities obtained from VAEs suffer from the same problems as per-pixel deterministic losses (cf. Section 4).

All the aforementioned works that use autoencoders for unsupervised defect segmentation have shown that autoencoders reliably reconstruct non-defective images while visually altering defective regions to keep the reconstruction close to the learned manifold of the training data. However, they rely on per-pixel loss functions that make the unrealistic assumption that neighboring pixel values are mutually independent. We show that this prevents these approaches from segmenting anomalies that differ predominantly in structure rather than pixel intensity. Instead, we propose to use SSIM (Wang et al., 2004) as the loss function and measure of anomaly by comparing input and reconstruction. SSIM takes interdependencies of local patch regions into account and evaluates their first and second order moments to model differences in luminance, contrast, and structure. (Ridgeway et al., 2015) show that SSIM and the closely related multi-scale version MS-SSIM (Wang et al., 2003) can be used as differentiable loss functions to generate more realistic images in deep architectures for tasks such as superresolution, but do not examine its usefulness for defect segmentation in an autoencoding framework. In all our experiments, switching from per-pixel to perceptual losses yields significant gains in performance, sometimes enhancing the method from a complete failure to a satisfactory defect segmentation result.

3 METHODOLOGY

3.1 Autoencoders for Unsupervised Defect Segmentation

Autoencoders attempt to reconstruct an input image $\mathbf{x} \in \mathbb{R}^{k \times h \times w}$ through a bottleneck, effectively projecting the input image into a lower-dimensional space, called latent space. An autoencoder consists of an encoder function $E: \mathbb{R}^{k \times h \times w} \rightarrow \mathbb{R}^d$ and a decoder function $D: \mathbb{R}^d \rightarrow \mathbb{R}^{k \times h \times w}$, where $d$ denotes the dimensionality of the latent space and $k, h, w$ denote the number of channels, height, and width of the input image, respectively. Choosing $d \ll k \times h \times w$ prevents the architecture from simply copying its input and forces the encoder to extract meaningful features from the input patches that facilitate accurate reconstruction by the decoder. The overall process can be summarized as

$$\hat{\mathbf{x}} = D(E(\mathbf{x})) = D(\mathbf{z}) \ ,$$

where $\mathbf{z}$ is the latent vector and $\hat{\mathbf{x}}$ the reconstruction of the input. In our experiments, the functions $E$ and $D$ are parameterized by CNNs. Strided convolutions are used to down-sample the input feature maps in the encoder and to up-sample them in the decoder. Autoencoders can be employed for unsupervised defect segmentation by training them purely on defect-free image data. During testing, the autoencoder will fail...
The approximate posterior distribution $Q(z|x)$ obtained by encoding an input image can be used to define further anomaly measures. One option is to compute a distance between the two distributions, such as the KL-divergence $\mathbb{KL}(Q(z|x)\|P(z))$, and indicate defects for large deviations from the prior $P(z)$ (Soukup and Pinetz, 2018). However, to use this approach for the pixel-accurate segmentation of anomalies, a separate forward pass for each pixel of the input image would have to be performed. A second approach for utilizing the posterior $Q(z|x)$ that yields a spatial residual map is to decode $N$ latent samples $z_1, z_2, \ldots, z_N$ drawn from $Q(z|x)$ and to evaluate the per-pixel reconstruction probability $R_{VAE} = P(x|z_1, z_2, \ldots, z_N)$ as described by (An and Cho, 2015).

### 3.1.3 Feature Matching Autoencoder

Another extension to standard autoencoders was proposed by (Dosovitskiy and Brox, 2016). It increases the quality of the produced reconstructions by extracting features from both the input image $x$ and its reconstruction $\hat{x}$ and enforcing them to be equal. Consider $F : \mathbb{R}^{l \times h \times w} \rightarrow \mathbb{R}^f$ to be a feature extractor that obtains an $f$-dimensional feature vector from an input image. Then, a regularizer can be added to the loss function of the autoencoder, yielding the feature matching autoencoder (FM-AE) loss

$$L_{FM}(x, \hat{x}) = L_2(x, \hat{x}) + \lambda \| F(x) - F(\hat{x}) \|^2_2,$$

where $\lambda > 0$ denotes the weighting factor between the two loss terms. $F$ can be parameterized using the first layers of a CNN pretrained on an image classification task. During evaluation, a residual map $R_{FM}$ is obtained by comparing the per-pixel $\ell^2$-distance of $x$ and $\hat{x}$. The hope is that sharper, more realistic reconstructions will lead to better residual maps compared to a standard $\ell^2$-autoencoder.

### 3.1.4 SSIM Autoencoder

We show that employing more elaborate architectures such as VAEs or FM-AEs does not yield satisfactory improvements of the residual maps over deterministic $\ell^2$-autoencoders in the unsupervised defect segmentation task. They are all based on per-pixel evaluation metrics that assume an unrealistic independence between neighboring pixels. Therefore, they fail to detect structural differences between the inputs and their reconstructions. By adapting the loss and evaluation functions to capture local inter-dependencies between image regions, we are able to drastically improve upon all the aforementioned architectures. In Section 3.2, we specifically motivate the use of the structural similarity metric SSIM($x, \hat{x}$) as both the loss function and the evaluation metric for autoencoders to obtain a residual map $R_{SSIM}$.
has a constant \( q \) form the SSIM index. Each of the patch pairs \( p \) and \( q \) (Wang et al., 2004). Figure 2 shows the difference in the patches’ mean, variance, and covariance in cases. SSIM on the other hand is sensitive to variations in the reconstruction. In contrast to the \( \ell^2 \)-error map, SSIM gives more importance to the visually more salient disturbances than to the slight inaccuracies around reconstructed edges.

### 3.2 Structural Similarity

The SSIM index (Wang et al., 2004) defines a distance measure between two \( K \times K \) image patches \( p \) and \( q \), taking into account their similarity in luminance \( l(p, q) \), contrast \( c(p, q) \), and structure \( s(p, q) \):

\[
SSIM(p, q) = l(p, q)\alpha c(p, q)\beta s(p, q),
\]

where \( \alpha, \beta, \gamma \in \mathbb{R} \) are user-defined constants to weight the three terms. The luminance measure \( l(p, q) \) is estimated by comparing the patches’ mean intensities \( \mu_p \) and \( \mu_q \). The contrast measure \( c(p, q) \) is a function of the patch variances \( \sigma_p^2 \) and \( \sigma_q^2 \). The structure measure \( s(p, q) \) takes into account the covariance \( \sigma_{pq} \) of the two patches. The three measures are defined as:

\[
l(p, q) = \frac{2\mu_p\mu_q + c_1}{\mu_p^2 + \mu_q^2 + c_1},
\]

\[
c(p, q) = \frac{2\sigma_p\sigma_q + c_2}{\sigma_p^2 + \sigma_q^2 + c_2},
\]

\[
s(p, q) = \frac{2\sigma_{pq} + c_2}{2\sigma_p\sigma_q + c_2}.
\]

The constants \( c_1 \) and \( c_2 \) ensure numerical stability and are typically set to \( c_1 = 0.01 \) and \( c_2 = 0.03 \). By substituting (5)-(7) into (4), the SSIM is given by

\[
SSIM(p, q) = \frac{(2\mu_p\mu_q + c_1)(2\sigma_{pq} + c_2)}{(\mu_p^2 + \mu_q^2 + c_1)(\sigma_p^2 + \sigma_q^2 + c_2)}. \tag{8}
\]

It holds that \( SSIM(p, q) \in [-1, 1] \). In particular, \( SSIM(p, p) = 1 \) if and only if \( p \) and \( q \) are identical (Wang et al., 2004). Figure 2 shows the different perceptions of the three similarity functions that form the SSIM index. Each of the patch pairs \( p \) and \( q \) has a constant \( \ell^2 \)-residual of 0.25 per pixel and hence assigns low defect scores to each of the three cases. SSIM on the other hand is sensitive to variations in the patches’ mean, variance, and covariance in its respective residual map and assigns low similarity to each of the patch pairs in one of the comparison functions.

To compute the structural similarity between an entire image \( x \) and its reconstruction \( \hat{x} \), one slides a \( K \times K \) window across the image and computes a SSIM value at each pixel location. Since (8) is differentiable, it can be employed as a loss function in deep learning architectures that are optimized using gradient descent.

Figure 3 indicates the advantages SSIM has over per-pixel error functions such as \( \ell^2 \) for segmenting defects. After training an \( \ell^2 \)-autoencoder on defect-free checkerboard patterns of various scales and orientations, we apply it to an image (Figure 3(a)) that contains gray strokes and dots that simulate defects. Figure 3(b) shows the corresponding reconstruction produced by the autoencoder, which removes the defects from the input image. The two remaining subfigures display the residual maps when evaluating the reconstruction error with a per-pixel \( \ell^2 \)-comparison or SSIM. For the latter, the luminance, contrast, and structure maps are also shown. For the \( \ell^2 \)-distance, both the defects and the inaccuracies in the reconstruction are weighted equally in the error map, which makes them indistinguishable. Since SSIM computes three different statistical features for image comparison and operates on local patch regions, it is less sensitive to small localization inaccuracies in the reconstruction. In addition, it detects defects that manifest themselves in a change of structure rather than large differences in pixel intensity. For the defects added in this particular toy example, the contrast function yields the largest residuals.
Figure 4: Example images from the contributed texture dataset of two woven fabrics. (a) and (b) show examples of non-defective textures that can be used for training. (c) and (d) show exemplary defects for both datasets. See the text for details.

4 EXPERIMENTS

4.1 Datasets

Due to the lack of datasets for unsupervised defect segmentation in industrial scenarios, we contribute a novel dataset of two woven fabric textures, which is available to the public\(^1\). We provide 100 defect-free images per texture for training and validation and 50 images that contain various defects such as cuts, roughened areas, and contaminations on the fabric. Pixel-accurate ground truth annotations for all defects are also provided. All images are of size 512 × 512 pixels and were acquired as single-channel gray-scale images. Examples of defective and defect-free textures can be seen in Figure 4. We further evaluate our method on a dataset of nanofibrous materials (Carrera et al., 2017), which contains five defect-free gray-scale images of size 1024 × 700 for training and validation and 40 defective images for evaluation. A sample image of this dataset is shown in Figure 1.

4.2 Training and Evaluation Procedure

For all datasets, we train the autoencoders with their respective losses and evaluation metrics, as described in Section 3.1. Each architecture is trained on 10,000 defect-free patches of size 128 × 128, randomly cropped from the given training images. In order to capture a more global context of the textures, we down-scaled the images to size 256 × 256 before cropping. Each network is trained for 200 epochs using the ADAM (Kingma and Ba, 2015) optimizer with an initial learning rate of 2 × 10^{-4} and a weight decay set to 10^{-5}. The exact parametrization of the autoencoder network shared by all tested architectures is given in Table 1. The latent space dimension for our experiments is set to \(d = 100\) on the texture images and to \(d = 500\) for the nanofibres due to their higher structural complexity. For the VAE, we decode \(N = 6\) latent samples from the approximate posterior distribution \(Q(z|x)\) to evaluate the reconstruction probability for each pixel. The feature matching autoencoder is regularized with the first three convolutional layers of an AlexNet (Krizhevsky et al., 2012) pre-trained on ImageNet (Russakovsky et al., 2015) and a weight factor of \(\lambda = 1\). For SSIM, the window size is set to \(K = 11\) unless mentioned otherwise and its three residual maps are equally weighted by setting \(\alpha = \beta = \gamma = 1\).

The evaluation is performed by striding over the test images and reconstructing image patches of size 128 × 128 using the trained autoencoder and computing its respective residual map \(R\). In principle, it would be possible to set the horizontal and vertical stride to 128. However, at different spatial locations, the autoencoder produces slightly different reconstructions of the same data, which leads to some striding artifacts. Therefore, we decreased the stride to 30 pixels and averaged the reconstructed pixel values. The resulting residual maps are thresholded to obtain candidate regions where a defect might be present. An opening with a circular structuring element of diameter 4 is applied as a morphological post-

---

### Table 1: General outline of our autoencoder architecture.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Output Size</th>
<th>Kernel</th>
<th>Stride</th>
<th>Padding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>128x128x1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Com1</td>
<td>64x64x32</td>
<td>4x4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Com2</td>
<td>32x32x32</td>
<td>4x4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Com3</td>
<td>32x32x32</td>
<td>3x3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Com4</td>
<td>16x16x64</td>
<td>4x4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Com5</td>
<td>16x16x64</td>
<td>3x3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Com6</td>
<td>8x8x128</td>
<td>4x4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Com7</td>
<td>8x8x64</td>
<td>3x3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Com8</td>
<td>8x8x32</td>
<td>3x3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Com9</td>
<td>1x1x(d)</td>
<td>8x8</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\(^1\)The dataset will be made available at http://www.mvtex.com/company/research/publications.
processing to delete outlier regions that are only a few pixels wide (Steger et al., 2018). We compute the receiver operating characteristic (ROC) as the evaluation metric. The true positive rate is defined as the ratio of pixels correctly classified as defect across the entire dataset. The false positive rate is the ratio of pixels misclassified as defect.

4.3 Results

Figure 5 shows a qualitative comparison between the performance of the $\ell^2$-autoencoder and the SSIM autoencoder on images of the two texture datasets. Although both architectures remove the defect in the reconstruction, only the SSIM residual map reveals the defects and provides an accurate segmentation result. The same can be observed for the NanoTWICE dataset, as shown in Figure 1.

We confirm this qualitative behavior by numerical results. Figure 6 compares the ROC curves and their respective AUC values of our approach using SSIM to the per-pixel architectures. The performance of the latter is often only marginally better than classifying each pixel randomly. For the VAE, we found that the reconstructions obtained by different latent samples from the posterior does not vary greatly. Thus, it could not improve on the deterministic framework. Employing feature matching only improved the segmentation result for the dataset of nanofibrous materials, while not yielding a benefit for the two texture datasets. Using SSIM as the loss and evaluation metric outperforms all other tested architectures significantly. By merely changing the loss function, the achieved AUC improves from 0.688 to 0.966 on the dataset of nanofibrous materials, which is comparable to the state-of-the-art given in (Napoletano et al., 2018), where values of up to 0.974 are reported. In contrast to this method, autoencoders do not rely on any model priors such as handcrafted features or pre-trained networks. For the two texture datasets, similar leaps in performance are observed.

Since the dataset of nanofibrous materials contains defects of various sizes and smaller sized defects contribute less to the overall true positive rate when weighting all pixel equally, we further evaluated the overlap of each detected anomaly region with the ground truth for this dataset and report the $p$-quantiles for $p \in \{25\%, 50\%, 75\%\}$ in Figure 7. For false positive rates as low as 5%, more than 50% of the defects have an overlap with the ground truth that is larger than...
91%. This outperforms the results achieved by (Napoletano et al., 2018), who report a minimal overlap of 85% in this setting.

We further tested the sensitivity of the SSIM autoencoder to different hyperparameter settings. We varied the latent space dimension $d$, SSIM window size $k$, and the size of the patches that the autoencoder was trained on. Table 2 shows that SSIM is insensitive to different hyperparameter settings once the latent space dimension is chosen to be sufficiently large. Using the optimal setup of $d = 500$, $k = 11$, and patch size $128 \times 128$, a forward pass through our architecture takes $2.23$ ms on a Tesla V100 GPU. Patch-by-patch evaluation of an entire image of the NanoTWICE dataset takes $3.61$ s on average, which is significantly faster than the runtimes reported by (Napoletano et al., 2018). Their approach requires between $15$ s and $55$ s to process a single input image.

Figure 8 depicts qualitative advantages that employing a perceptual error metric has over per-pixel distances such as $\ell^2$. It displays two defective images from one of the texture datasets, where the top image contains a high-contrast defect of metal pins which contaminate the fabric. The bottom image shows a low-contrast structural defect where the fabric was cut open. While the $\ell^2$-norm has problems to detect the low-contrast defect, it easily segments the metal pins due to their large absolute distance in gray values with respect to the background. However, misalignments in edge regions still lead to large residuals in non-defective regions as well, which would make these thin defects hard to segment in practice. SSIM robustly segments both defect types due to its simultaneous focus on luminance, contrast, and structural information and insensitivity to edge alignment due to its patch-by-patch comparisons.

5 CONCLUSION

We demonstrate the advantage of perceptual loss functions over commonly used per-pixel residuals in autoencoding architectures when used for unsupervised defect segmentation tasks. Per-pixel losses fail to capture inter-dependencies between local image regions and therefore are of limited use when defects manifest themselves in structural alterations of the defect-free material where pixel intensity values stay roughly consistent. We further show that employing probabilistic per-pixel error metrics obtained by VAEs or sharpening reconstructions by feature matching regularization techniques do not improve the segmentation result since they do not address the problems that arise from treating pixels as mutually independent.

SSIM, on the other hand, is less sensitive to small inaccuracies of edge locations due to its comparison of local patch regions and takes into account three different statistical measures: luminance, contrast, and structure. We demonstrate that switching from per-pixel loss functions to an error metric based on structural similarity yields significant improvements by evaluating on a challenging real-world dataset of nanofibrous materials and a contributed dataset of two woven fabric materials which we make publicly available. Employing SSIM often achieves an enhancement from almost unusable segmentations to results that are on par with other state of the art approaches for unsupervised defect segmentation which additi-
onally rely on image priors such as pre-trained networks.

REFERENCES


