

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.

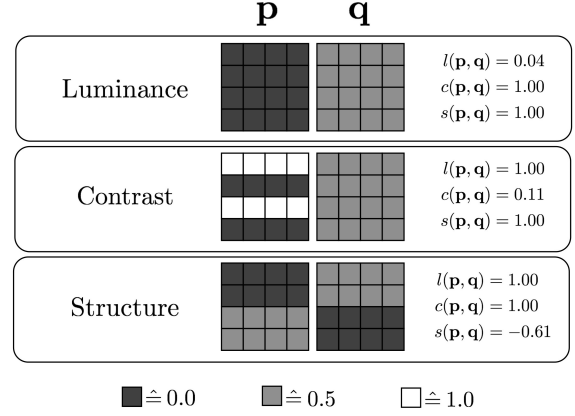
### 3. METHODOLOGY

#### 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

$$\hat{\mathbf{x}} = D(E(\mathbf{x})) = D(\mathbf{z}) , \quad (1)$$

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



**Figure 2:** Different responsibilities of the three similarity functions employed by SSIM. Example patches  $\mathbf{p}$  and  $\mathbf{q}$  differ in either luminance, contrast, or structure. SSIM is able to distinguish between these three cases, assigning close to minimum similarity values to one of the comparison functions  $l(\mathbf{p}, \mathbf{q})$ ,  $c(\mathbf{p}, \mathbf{q})$ , or  $s(\mathbf{p}, \mathbf{q})$ , respectively. An  $\ell^2$ -comparison of these patches would yield a constant per-pixel residual value of 0.25 for each of the three cases.

training them purely on defect-free image data. During testing, the autoencoder will fail to reconstruct defects that have not been observed during training, which can thus be segmented by comparing the original input to the reconstruction and computing a residual map  $R(\mathbf{x}, \hat{\mathbf{x}}) \in \mathbb{R}^{w \times h}$ .

**3.1.1.  $\ell^2$ -Autoencoder.** To force the autoencoder to reconstruct its input, a loss function must be defined that guides it towards this behavior. For simplicity and computational speed, one often chooses a per-pixel error measure, such as the  $L_2$  loss

$$L_2(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{r=0}^{h-1} \sum_{c=0}^{w-1} (\mathbf{x}(r, c) - \hat{\mathbf{x}}(r, c))^2 , \quad (2)$$

where  $\mathbf{x}(r, c)$  denotes the intensity value of image  $\mathbf{x}$  at the pixel  $(r, c)$ . To obtain a residual map  $R_{\ell^2}(\mathbf{x}, \hat{\mathbf{x}})$  during evaluation, the per-pixel  $\ell^2$ -distance of  $\mathbf{x}$  and  $\hat{\mathbf{x}}$  is computed.

**3.1.2. Variational Autoencoder.** Various extensions to the deterministic autoencoder framework exist. VAEs (Kingma and Welling, 2014) impose constraints on the latent variables to follow a certain distribution  $\mathbf{z} \sim P(\mathbf{z})$ . For simplicity, the distribution is typically chosen to be a unit-variance Gaussian. This turns the entire framework into a probabilistic model that enables efficient posterior inference and allows to generate new data from the training manifold by sampling from the latent distribution. The approximate posterior distribution  $Q(\mathbf{z}|\mathbf{x})$  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  $\mathcal{KL}(Q(\mathbf{z}|\mathbf{x})||P(\mathbf{z}))$ , and indicate defects for large deviations from the prior  $P(\mathbf{z})$  (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