これらの手法をピクセル単位の再構築損失と組み合わせる。これにより、異常を示す単一のスカラ値が得られます。これは、各画像ピクセルに対して正確なセグメンテーション結果を得るために別途フォワードパスが必要となるセグメンテーションシナリオにおいて、迅速にパフォーマンスのボトルネックとなる可能性があります。私たちは、VAEから得られるピクセル単位の再構築確率が、ピクセル単位の確定的損失と同じ問題を抱えていることを示します（参照：第4節）。

自動エンコーダーを無監督欠陥セグメンテーションに用いた上記のすべての研究は、自動エンコーダーが非欠陥画像を信頼性高く再構築しつつ、欠陥領域を視覚的に変更して再構築を訓練データの学習されたマニフォールドに近づけることを示しています。しかし、これらの研究は、隣接するピクセル値が相互に独立しているという現実的でない仮定に基づくピクセル単位の損失関数に依存しています。私たちは、この仮定が構造的な違いが主でピクセル強度ではない異常のセグメンテーションを妨げることを示します。代わりに、入力と再構築を比較して異常を測定する損失関数としてSSIM（Wang et al., 2004）を使用することを提案します。SSIMは局所パッチ領域の相互依存性を考慮し、輝度、コントラスト、構造の違いをモデル化するために、その1次および2次モーメントを評価します。Ridgeway et al.（2015）は、SSIMと密接に関連するマルチスケール版MS-SSIM（Wang et al., 2003）が、超解像度などのタスクにおいて深層アーキテクチャでより現実的な画像を生成するための微分可能な損失関数として使用できることを示していますが、オートエンコーダーフレームワークにおける欠陥セグメンテーションへの有用性は検討していません。すべての実験において、ピクセル単位の損失関数から知覚的損失関数への切り替えは性能に顕著な向上がもたらされ、場合によっては完全な失敗から満足のいく欠陥セグメンテーション結果へと改善されます。

ここで、x（r, c）は画像 x のピクセル（r, c）における輝度値を表します。評価時に残差マップ R_{^} 2（x, x^）を取得するため、x と x^ のピクセル単位の `^2 -距離が計算されます。

3.1. 自動エンコーダーを用いた非監督型欠陥セグメンテーション

オートエンコーダーは、入力画像x $\in_R^{k\times h\times w}$をボトルネックを通じ再構築し、入力画像を低次元空間（潜在空間）に投影する。オートエンコーダーは、エンコーダー関数 E $:_R^{k\times h\times w} \to_R^d$ とデコーダー関数 D $:_R^d \to_R^{k\times h\times w}$ から構成され、d は潜在空間の次元数を、k、h、w は入力画像のチャネル数、高さ、幅をそれぞれ表します。d k × h × wを選択することで、アーキテクチャが入力を単純にコピーするのを防ぎ、エンコーダーが入力パッチから意味のある特徴を抽出するように強制し、デコーダーによる正確な再構築を可能にします。全体的なプロセスは次のように要約できます

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

ここで、z は潜在ベクトル、x^ は入力の再構築です。当社の実験では、関数 E および D は CNN によってパラメータ化されています。ストライド畳み込みは、エンコーダーで入力特徴マップをダウンサンプリングし、デコーダーでアップサンプリングするために使用されます。オートエンコーダは、欠陥のない画像データのみを用いてトレーニングを行うことにより、教師なし欠陥セグメンテーションに活用することができます。

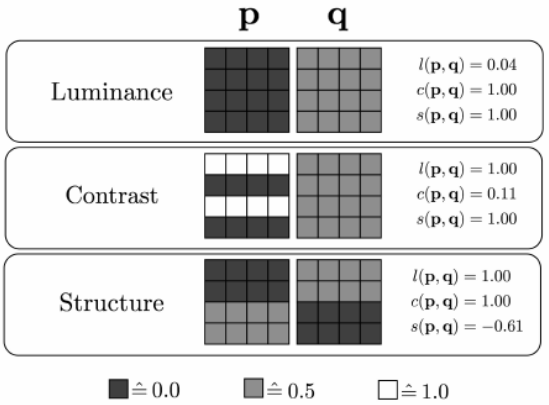

図2：SSIMで用いられる3つの類似性関数の異なる役割。例として、パッチpとqは輝度、コントラスト、または構造のいずれかで異なります。SSIMはこれらの3つのケースを区別し、比較関数l(p, q)、c(p, q)、またはs(p, q)のいずれかに最小の類似性値を割り当てます。これらのパッチの`2 -比較では、3つのケースそれぞれに対して、各ピクセルごとの残差値が0.25の定数値となります。

テスト中、オートエンコーダはトレーニング中に観察されなかった欠陥を再構築できないため、元の入力と再構築結果を比較し、残差マップ R(x , x^) $\in_R^{w\times h}$ を計算することで、その欠陥をセグメンテーションすることができます。

3.1.1. `2 -オートエンコーダー。

オートエンコーダーが入力を再構築するように強制するには、この動作を誘導する損失関数を定義する必要があります。

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

簡素化と計算速度の向上のため、通常はピクセル単位の誤差指標を選択します。例えば、L_2 損失

3.1.2. 変分オートエンコーダー。

確定的オートエンコーダーのフレームワークにはさまざまな拡張が存在します。VAE（Kingma and Welling, 2014）は、潜在変数が特定の分布 z ¤ P（ z ）に従うように制約を課します。簡素化のため、分布は通常、単位分散ガウス分布が選択されます。これにより、全体的なフレームワークは確率的モデルとなり、効率的な事後推論を可能にし、訓練マニフォールドから潜在分布をサンプリングすることで新しいデータを生成できます。入力画像をエンコードして得られる近似事後分布Q(z|x)は、さらに異常測定を定義するために使用できます。1 つの選択肢は、2 つの分布間の距離、たとえば KL 発散 KL(Q(z | x)||P（ z ））を計算し、事前分布 P（ z ）から大きく逸脱している部分を欠陥として示すことです（Soukup and Pinetz, 2018）。しかし、このアプローチをピクセル精度の異常のセグメンテーションに使用するには、入力画像の各ピクセルについて個別のフォワードパスを実行する必要があります。事後分布を活用する 2 つ目のアプローチ

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

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

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 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 $R(\mathbf{x}, \hat{\mathbf{x}}) \in \mathbb{R}^{w \times h}$.

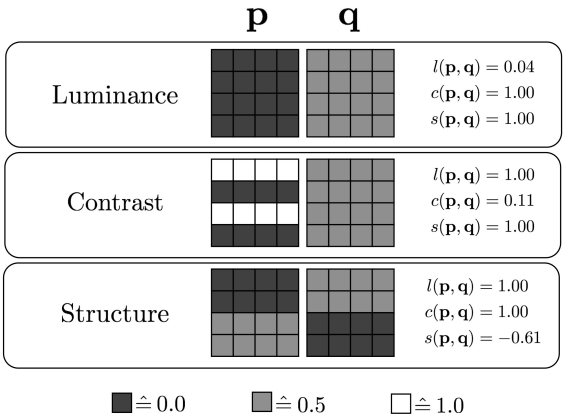

Figure 2: Different responsibilities of the three similarity functions employed by SSIM. Example patches $\mathbf{p}$ and $\mathbf{q}$ differ in either luminance, contrast, or structure. SSIM is able to distinguish between these three cases, assigning close to minimum similarity values to one of the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

3.1.1. ℓ^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

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

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 ℓ^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