ITERATIVE IMAGE INPAINTING WITH STRUCTURAL SIMILARITY MASK FOR ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoencoders have emerged as popular methods for unsupervised anomaly detection. Autoencoders trained on the normal data are expected to reconstruct only the normal features, allowing anomaly detection by thresholding reconstruction errors. However, in practice, autoencoders fail to model small detail and yield blurry reconstructions, which makes anomaly detection challenging. Moreover, there is objective mismatching that models are trained to minimize total reconstruction errors while we expect a small deviation on normal pixels and large deviation on anomalous pixels. To tackle these two issues, we propose the iterative image inpainting method that reconstructs partial regions in an adaptive inpainting mask matrix. This method constructs inpainting masks from the anomaly score of structural similarity. Overlaying inpainting mask on images, each pixel is bypassed or reconstructed based on the anomaly score, enhancing reconstruction quality. The iterative update of inpainted images and masks by turns purifies the anomaly score directly and follows the expected objective at test time. We evaluated the proposed method using the MVTec Anomaly Detection dataset. Our method outperforms previous state-of-the-art in several categories and showed remarkable improvement in high-frequency textures.

1 INTRODUCTION

Anomaly detection (AD) is the identification task of the rarely happened events or items that differ from the majority of the data. In the real world, there are many applications, such as the medial diagnosis (Baur et al., 2018; Zimmerer et al., 2019b;a), defect detection in the factories (Matsubara et al., 2018; Bergmann et al., 2018; 2019), early detection of plant disease (Wang et al., 2019), and X-Ray security detection in public space (Griffin et al., 2018). Because manual inspection by humans is slow, expensive, and error-prone, automating visual inspection is the popular application of artificial intelligence. To transfer knowledge from humans to machines, there is a lack of anomalous samples by their low event rate and difficulty annotating and categorizing various anomalous defects beforehand. Therefore, AD methods typically take unsupervised approaches that try to learn compact features of data from normal samples and detect anomalies by thresholding anomaly score to measure the deviation from learned features. To deal with high-dimensional images and learn their features, it is popular to use deep neural networks (Goodfellow et al., 2016).

In this work, we focus on the reconstruction-based unsupervised AD. This attempts to reconstruct only the normal dataset and classify the normal or anomalous data on thresholding reconstruction errors (Bergmann et al., 2018; 2019). The architectures are based on deep neural networks such as deep autoencoders (Goodfellow et al., 2016), variational autoencoders (VAEs) (Kingma & Welling, 2013; Rezende et al., 2014), or autoencoders with generative adversarial networks (GANs) (Goodfellow et al., 2014; Schlegl et al., 2017; 2019). These models compress the high-dimensional information into the data manifold in lower-dimensional latent space by reconstructing input data under certain constraints for latent space, such as a prior distribution or an information bottleneck (Alemi et al., 2016).

The issue in the reconstruction-based approach is that autoencoders fail to model small details and yield blurry image reconstruction, especially for the high-frequency textures, such as carpet, leather, and tile (Bergmann et al., 2019). Dehaene et al. (2020) pointed out that there is no guarantee of the generalization of their behavior for out-of-samples and local defects added to normal images can

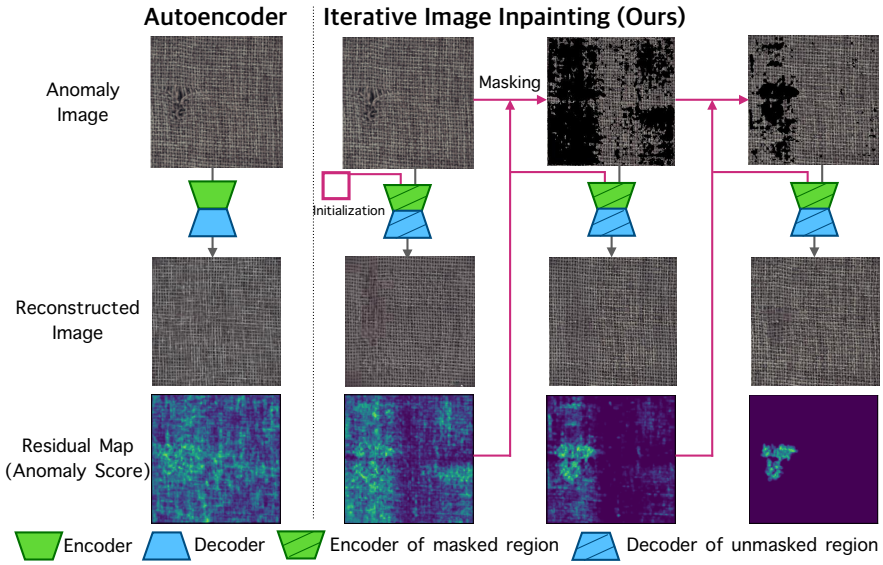

Figure 1: Autoencoder vs. I3AD method (Ours). Autoencoder fails to reconstruct the high-frequency texture of "carpet" and yields a large residual map. Ours utilizes residual map as an inpainting mask and iteratively updates the reconstructed image and the residual map.

also deteriorate whole images. In the viewpoint of the signal-to-noise ratio (SNR), it is interpreted that blurry reconstruction makes anomaly signals (reconstruction errors in anomaly pixels) unclear and increases normal noises (reconstruction errors in normal pixels). The SNR explains the feasibility of AD by thresholding a sample-wise reconstruction errors. Therefore, low SNR leads to AD challenging.

We point out an additional issue about the gap between optimized function at training and evaluated function at testing. Rethinking our goal in unsupervised AD, it is concluded not to minimize reconstruction errors merely but to maximize the SNR. Although models are trained to minimize a sample-wise reconstruction error, they are expected to have a large deviation of anomaly pixels and a small deviation of normal pixels at testing.

In this paper, we propose *I3AD* (Iterative Image Inpainting for Anomaly Detection). As Figure 1, our method utilizes inpainting models that only encode unmasked regions and reconstruct masked regions, instead of vanilla autoencoders. Once computed anomaly score, it is recycled as an inpainting mask for the next iteration. We show that the iterative update enhances the reconstruction quality and satisfies the expected objective to maximize the expected SNR at testing. Through experiments and analysis on the MVTec AD dataset shows that our I3AD outperforms existing methods on nine categories and has ave. $+11.6\%$ improvement on texture category.

2 Methodology

2.1 High-level idea

We think of unsupervised AD using autoencoders. Here, we implicitly assume anomalies show up in partial regions, and pixels in surrounding regions obey by the distribution of normal datasets. Therefore, the sample-wise anomaly score based on reconstruction errors is the summation on two types of pixels: (1) pixels of normal regions (normal noises) and (2) pixels of anomalous regions (anomaly signals). We expect an ideal model with zero errors on normal regions and distinguishable per-pixel scores on anomalous regions, leading to the high SNR computed as anomaly signals divided by normal noises.

Inheriting vanilla autoencoders' architecture does not help to resolve the low SNR issue mentioned by Bergmann et al. (2019) and Dehaene et al. (2020). As basic architecture, autoencoders are forced to encode whole images including local defects and decode whole images including fine structures

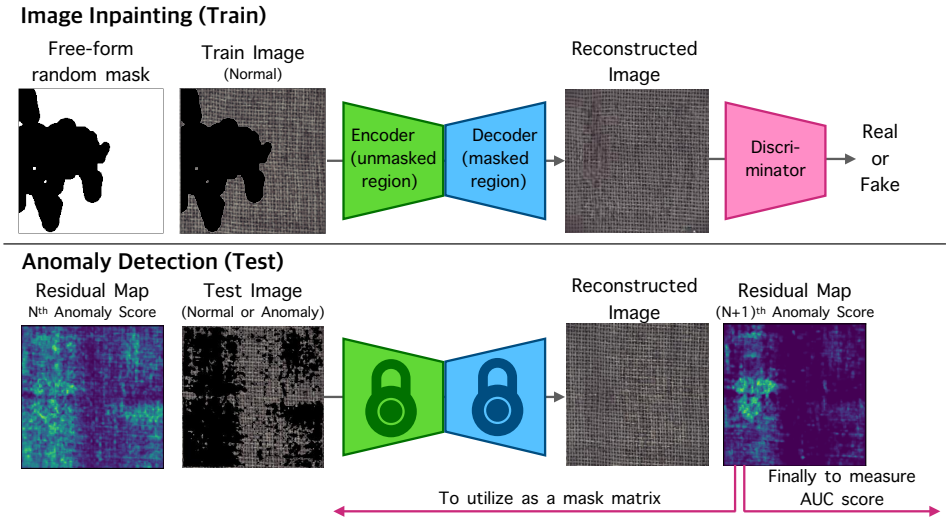

Figure 2: Model overview of the iterative image inpainting method for AD (I3AD). At the training step, we solve a general image inpainting task in normal datasets. At the testing step, a generator receives test images with an adaptive mask and only reconstructs masked regions. This adaptive mask is updated from previous reconstruction results during iterations.

of background pixels. They are unable to learn their behavior against inputting anomalous pixels, while the local defects can affect decoding whole reconstruction.

One approach to realize the ideal model and resolve the issue is a combination of per-pixel identity function and conditional autoencoder. Compared to vanilla autoencoders, conditional autoencoders can encode only normal regions and decode only anomalous regions. The per-pixel identity function could copy the remaining unreconstructed regions. This model architecture is the same as an image inpainting model. Actually, deep inpainting models are designed by conditional autoencoders that encode unmasked regions and fill in masked regions under a certain mask matrix (Yu et al., 2019). However, the image inpainting method for AD falls into the tautology trap that we do not know a perfect inpainting mask matrix in advance, and detecting anomalous regions is the main goal to achieve.

The key ideas to disentangle this tautology are mask generation by the anomaly score and iterative update of a mask matrix. Updating the inpainting mask matrix dynamically controls encoding and decoding information with a pixel-wise confidence level for AD. A generator gradually receives more information on potentially normal pixels and focuses on the suspected pixels during iterations. Furthermore, this process not only reduces background noises but also improves the SNR directly.

2.2 Iterative Image Inpainting for Anomaly Detection

Our model overview is depicted in Figure 2. Following the above discussion, we construct our I3AD method by an inpainting generator using a conditional generative adversarial network (cGAN) (Isola et al., 2017) and a mask generator utilizing anomaly score based on structural similarity measure (SSIM) index (Wang et al., 2004). As a mask generator, we explain the detail in the next subsection.

At first, we solve a general image inpainting task using under normal dataset to train a generative network. Normal images are partially hidden by randomly generated masks. A generator network is trained to decode the masked regions from the unmasked regions and the corresponding bool mask matrix. A discriminator network distinguishes generated images from normal images. A generator is rewarded for fooling a discriminator, while a discriminator is rewarded for detecting the generated images, respectively. This training can be considered as the two-player min-max game that a generator and a discriminator compete. As a result, the inpainting model tries to find the

optimal point on the loss function as below:

$$\min_{G}\max_{D} \; E_{x\sim P_{\mathrm{data}}(x)}[\log D(x)]+E_{x\sim P_{\mathrm{data}}(x),M\sim P(M)}[\log(1-D(G(\hat{x}=M\odot x,M)))],$$

where x and $\hat{x}$ are real samples from data distribution $P_{\mathrm{data}}(x)$ and their masked real samples. $\odot$ is an element-wise product. M is the corresponding bool mask matrices generated from the random distribution $P(M)$. $G(\hat{x},M)$ is an image inpainting network that takes an incomplete image and masked matrix. $D(x)$ denotes a binary classifier whether an image is generated or real.

We borrow and customize Spectral-Normalized Markovian GAN (SN-PatchGAN) architecture following by Yu et al. (2019). The network consists of two networks: a coarse-to-fine generator network with attention module and gated convolution, and a spectral normalized Markovian (Patch) discriminator network. Our I3AD expects to handle more fine structured masks than usual free-form masks. Therefore, to better handle more irregular masks, we apply the self-attention module (Zhang et al., 2018) instead of the contextual attention module originally designed for large rectangular masks as described in Yu et al. (2018; 2019).

To stabilize the training of GAN, we adapted proposed spectral normalization (Miyato et al., 2018) for discriminator's layers. As approximation of objective min-max loss function (Miyato et al., 2018), we also followed loss functions respectively for generator L_G and discriminator L_D below:

$$\begin{aligned}
L_G &= -E_{x\sim P_{\mathrm{data}}(x),M\sim P(M)}[D^{\mathrm{sn}}(G(\hat{x}=M\odot x,M))] \\
L_D &= E_{x\sim P_{\mathrm{data}}(x)}[\mathrm{ReLU}(1-D^{\mathrm{sn}}(x))] \\
&\quad + E_{x\sim P_{\mathrm{data}}(x),M\sim P(M)}[\mathrm{ReLU}(1+D^{\mathrm{sn}}(G(\hat{x}=M\odot x,M)))],
\end{aligned}$$

where $D^{\mathrm{sn}}(x)$ denotes spectral-normalized discriminator. ReLU is the abbreviation of Rectified Linear Units activation function, defined by $\mathrm{ReLU}(x)=\max(0,x)$. For a generator network, we use a spatially discounted l_1 reconstruction loss (Yu et al., 2018).

At the test step, a generator receives test images that are normal or anomalous images and an adaptive mask matrix. This mask matrix is constructed from the target image and reconstruction image at the previous iteration step and dynamically updated during iterations. Here, the generator reconstructs targeted masked regions that are dynamically changed, utilizing the gradually increasing information of surrounding regions in iteration steps.

2.3 Inpainting mask of Structural similarity (SSIM Mask)

To mask anomalous pixels during test iterations, we propose the structural similarity mask (SSIM-Mask). SSIM-Mask M_i is a binary mask thresholding the pixel-wise SSIM Index between an input image x_0 and the reconstructed image $\tilde{x}_i$ at i^{th} iteration step, defined as

$$\begin{aligned}
\tilde{x}_i &= G(\hat{x}_{i-1},M_{i-1}) \\
a_i(x) &= \mathrm{SSIM}(x_0,\tilde{x}_i) \\
M_i &= \begin{cases} 1 & \text{if} \quad a_i(x)\geq u \\ 0 & \text{otherwise,} \end{cases}
\end{aligned}$$

where u denotes the threshold level for binary classification. Details of SSIM calculation is in Appendix A. After N iterations, we use the Nth SSIM anomaly score $a_N(x)$ for AD evaluation.

3 Results

3.1 Experiment Configuration

We performed experiments with the industrial dataset named MVTec Anomaly Detection (MVTec AD)(Bergmann et al., 2019). Please see Appendix B for further details. We resize all images to 256×256 pixels in 15 categories for training and testing.

I3AD We train an I3AD generator by a total of 500,000 iterations with 10 images per batch. The optimizers for both the generator and the discriminator are Adam (Kingma & Ba, 2014) with learning rates of 0.0001 for the generator and 0.0004 for the discriminator, respectively. We generate

Table 1: Results for AD on the MVTec AD dataset, expressed in the AUROC for different autoencoders and datasets. We compare the vanilla L^2 and SSIM autoencoders(Bergmann et al., 2018) and their iterative projection method (Dehaene et al., 2020) as baselines. Bold font is the best AUC in each category and lightblue background indicates the best method measured by the average AUC on L^2 and SSIM anomaly score.

		Bergmann et al. (2019) autoencoders				Dehaene et al. (2020) iterative projection				Ours Inpainting	
	Model	L^2AE		DSAE		L^2AE		DSAE		I3AD	
	Score	L^2	SSIM	L^2	SSIM	L^2	SSIM	L^2	SSIM	L^2	SSIM
Textures	**carpet**	0.476	0.531	0.401	0.538	0.447	0.513	0.402	0.449	0.602	**0.605**
	grid	0.947	0.764	0.849	0.665	0.947	0.936	0.891	0.840	**0.998**	0.947
	leather	0.778	0.694	0.720	0.382	0.826	0.831	0.772	0.752	0.823	**0.659**
	tile	0.763	0.706	0.533	0.380	0.810	0.815	0.684	0.662	0.978	**0.983**
	wood	0.939	0.861	0.894	0.875	**0.952**	0.935	0.915	0.896	0.938	0.936
Objects	**bottle**	0.978	0.921	0.966	0.921	0.958	0.980	0.967	**0.983**	0.966	0.905
	cable	0.758	0.827	0.751	0.807	0.564	0.725	0.630	0.717	0.767	**0.773**
	capsule	0.728	0.689	0.738	0.699	0.792	0.803	0.819	**0.813**	0.708	0.731
	hazelnut	0.902	0.841	0.898	0.845	0.956	0.942	**0.957**	0.912	0.930	0.895
	metal nut	0.613	0.569	0.597	0.557	0.583	0.623	0.547	0.590	**0.658**	0.605
	pill	0.825	**0.838**	0.780	0.818	0.741	0.746	0.721	0.725	0.783	0.780
	screw	0.015	0.670	0.264	0.659	0.768	0.773	0.772	0.718	**0.980**	0.842
	toothbrush	0.981	0.886	0.972	0.875	**0.989**	0.978	0.983	0.975	0.958	0.903
	transistor	0.820	0.881	0.818	0.879	0.746	0.813	0.728	0.792	0.864	**0.876**
	zipper	0.873	0.785	0.833	0.755	0.818	0.864	0.772	0.810	0.994	**0.998**
Ave.	**Texture**	0.781	0.711	0.680	0.568	0.796	0.806	0.733	0.720	**0.868**	0.826
	Object	0.749	0.791	0.762	0.782	0.792	0.825	0.790	0.804	**0.861**	0.831
	Total	0.760	0.764	0.734	0.710	0.793	0.818	0.771	0.776	**0.863**	0.829

randomly blushing stroke masks using the algorithm proposed by Yu et al. (2019). Regarding mask generation, we apply 6 by 6 patched four checkerboard matrices masks initialization. To find the threshold parameter u, we compute the L_1 reconstruction errors on training datasets in each category from 0.10 to 0.50 by 0.05. Then we find a minimal threshold to achieve small errors less than the order of 10.

Baselines As baseline models, we compare two autoencoders that are trained to minimize different reconstruction loss of L^2 and SSIM. Similar to Bergmann et al. (2018), both autoencoder models are parameterized by the same architecture with latent space dimensionality (set to 100) and optimized by Adam with the same learning rate (set to 0.0001). These models are trained by 20,000 iterations on 10 images per batch. Both L^2 and SSIM autoencoders are applied by the iterative projection method. have 50 iterations with step size $\alpha = 0.5$ and L^1 regulation weight $\lambda = 0.05$, same parameter setting with Dehaene et al. (2020).

For evaluation, we compute the Area Under the Receiver Operating Characteristics (AUROC) to get a performance independent of the determined threshold. The AUROC scores of L^2 reconstruction error and SSIM dissimilarity on whole images are computed for each model. For the SSIM Index, their hyperparameters are selected as window size 11, $\alpha = \beta = \gamma = 1$, same with Bergmann et al. (2018). We find the threshold hyperparameter u for a generator to achieve low l_1 reconstruction errors on training normal images.

All experiments were performed on a graphics processing unit NVIDIA Quadro RTX 8000 based on a system running Python 3.7, PyTorch library version 1.5, and CUDA 10.1.

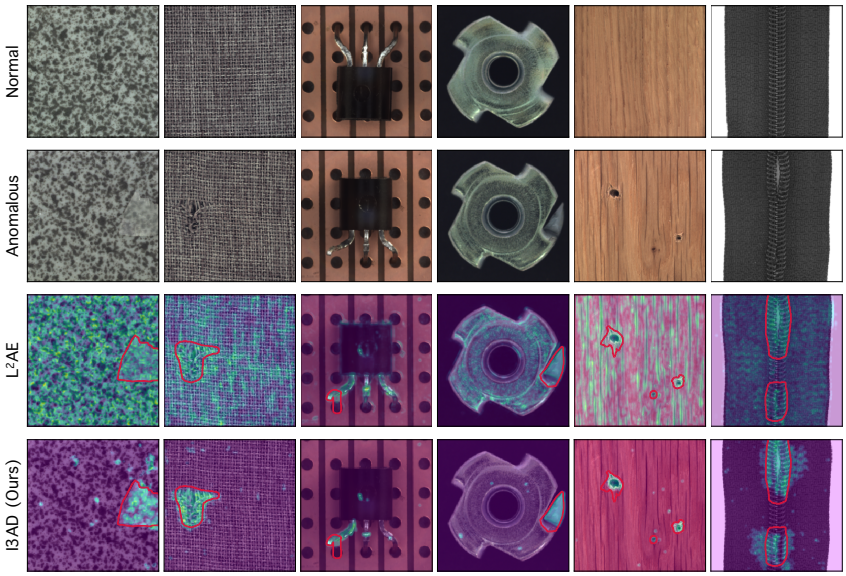

Figure 3: First row: Normal samples of tile, carpet, transistor, screw, wood, and zipper in MVTec-AD dataset; Second row: anomalous samples from the aforementioned dataset categories; Third row: Anomaly map by L^2 autoencoder (Bergmann et al., 2019); Fourth row: our proposed anomaly map by our I3AD method. Ground truth is represented by red contour, and each estimated anomaly score is highlighted by green.

3.2 Qualitative and Quantitative Results

Table 1 shows the AUROC for different autoencoders and datasets. The baselines of vanilla L^2 and SSIM autoencoders perform high AUC results on high-resolution images for object datasets, whereas the high-frequency texture data such as carpet, leather, or tile are challenging for them. I3AD measures SSIM anomaly scores due to utilizing the SSIM-Mask update, which minimizes reconstruction errors under the SSIM Index measure. I3AD iteration reduces background noise irreversibly, and the AUC score measured by L^2 distance got the almost same score with SSIM distance after the iteration finished. Our I3AD outperforms baselines on nine categories and has ave. I3AD improves ave. $+11.1\%$ on textures, ave. $+8.85\%$ on objects and ave. $+13.01\%$ in total from vanilla autoencoders.

Figure 3 shows the qualitative results of a baseline L^2AE and our I3AD method. SSIM anomaly score is used for highlighting. Our I3AD has large improvement in high-frequency categories such as tile, carpet, and zipper. Appendix E shows the visual results of the remaining categories of the MVTecAD dataset.

3.3 Hyperparameter Sensitivity

In this section, we investigate the hyperparameter sensitivity for the AD results. Compared to typical inpainting hard tasks to fill various types of images, usage for AD relax model to learn every single category. Thus, instead of hyperparameters of the inpainting, we focus on the hyperparameters regarding mask generation.

Mask initialization We use four checkerboard matrices to initialize inpainting masks at testing. Appendix 6 shows examples of checkerboard matrices on different patch sizes. In Figure 4, we found that different initialization has performance differences in the first few steps while they achieve similar results after enough iterations.

Mask generation threshold We determine mask generation threshold u naively from the L_1 reconstruction levels on training datasets. Figure 5 shows that the best parameters of each category are unknown and inaccessible from training datasets. However, we could find a threshold param-

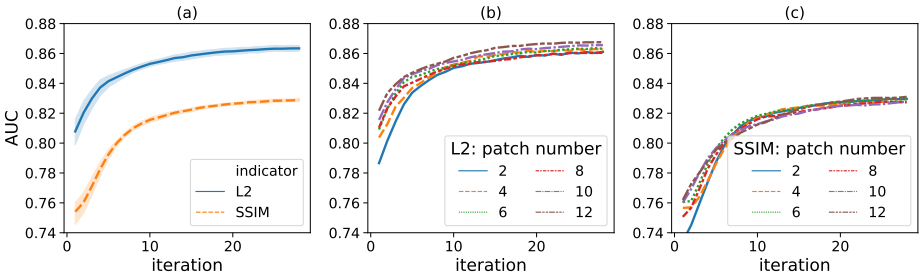

Figure 4: AUC change of different patch-size mask initialization during iteration. Both L^2 and SSIM anomaly scores are computed. (a) average AUC result of each anomaly score, (b) AUC result measured by L^2 (c) AUC result measured by SSIM

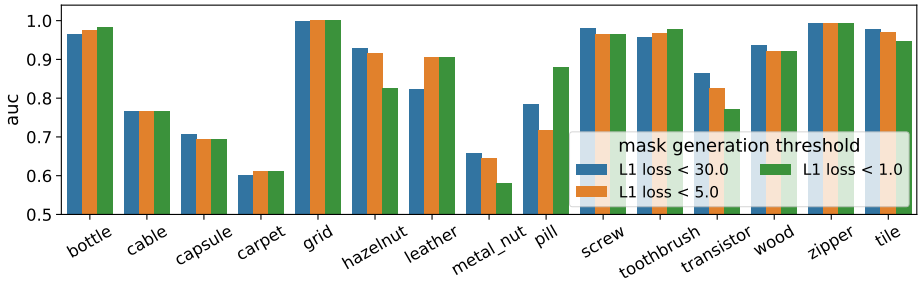

Figure 5: AUC change of different threshold u for mask generation

eter achieving similar total AUC score from enough cutoff bounds naively and have large room to optimize this parameter in semi-supervised approach.

4 Related Works

Image inpainting Image inpainting (a.k.a. image completion or image filling) is one of the Image-to-Image translation tasks to cover damaged, deteriorating, or missing parts from remaining parts or alternative contents guided by users. In computer vision, recently, there is much research about the approach based on feed-forward generative models with convolutional networks, especially using conditional generative adversarial networks (cGANs)(Isola et al., 2017).

Iizuka et al. (2017) proposed the model with global and local consistency to handle high-resolution images. To handle irregular masks, Liu et al. (2018) proposed a partial convolution layer that the weights in the convolutional layers are re-normalized by the number of valid pixels from the mask matrix. To produce higher-quality image inpainting, Yu et al. (2018) proposed two-stage architecture with coarse and refinement networks and a contextual attention module in a refinement network. The contextual attention module captures long-range spatial dependencies between masked regions and their surroundings, allowing models to fill large rectangular masks. To handle free-form masks, (Yu et al., 2019) proposed the gated convolution that extends the partial convolution and works pixel normalization and mask updates as trainable weights in neural network layers.

Anomaly detection Bergmann et al. (2019) proposed the MV-TecAD dataset and reviewed a brief overview of current state-of-the-art methods.

Schlegl et al. (2017) utilized GAN to model the manifold of the training data. A generator generates various anomalous-free images from a random noise sample. However, it could not generate an anomalous-free image close to a targeted image. To generate the closest sample, their algorithm iteratively finds a latent noise sample within the reconstruction errors and the similarity of outputs in a discriminator's first layer. Since there was a drawback of long runtimes from iterative computation,

several studies proposed encoder networks to efficiently find an expected random noise. (Zenati et al., 2018; Akcay et al., 2018; Schlegl et al., 2019) These model architectures are basically close to autoencoders with an adversarial term, whereas they train a generator network and an encoder network differently with additional loss terms.

Dehaene et al. (2020) proposed the iterative projection method on trained autoencoders to make blurred images clear at testing time. It iteratively minimizes the energy function measuring reconstruction errors and projects an input sample to the normal manifold of trained autoencoders. They similarly leverage the localization mask $\Omega_t = \|x_t - f_{\mathrm{VAE}}(x_t)\|^2$ at iteration step and show the improvement to reduce the number of iterations. On the other hand, this method still reconstructs the whole image. They compute the backpropagation with enough iterations for the energy minimization at test time, whereas our model only has forward calculation. This technique largely improves the reconstruction of existing broader autoencoders under 128×128 pixels patches cropped from 512×512 pixels in textures and 256×256 pixels in objects. On the other hand, we found their improvement is limited under 256×256 pixels in high-frequency textures, such as carpet, leather, and tile.

5 Discussions and future works

The recent autoencoder-based model improves to minimize reconstruction errors utilizing additional modules or loss terms, as shown in the previous section. To the best of my knowledge, these studies ignore the fundamental phenomenon that autoencoders might learn the trivial identity function if models are allowed too much capacity (Goodfellow et al., 2016). In order to extract a useful representation of data distribution, many studies are proposing additional regularized terms such as denoising, contrastive, sparse, stacked autoencoders. However, this focuses on the representation of latent space and does not ensure the prevention of learning trivial identity function.

Indeed, we have a simple experiment about several models based on U-Net architectures (Ronneberger et al., 2015) with switching their four skip-connections. We faced on learning of trivial identity function and reconstruction of anomalous pixels themselves if models have at least one skip-connection from three connections sorted by upstream layers. Skip connections will not only unexpectedly changes model trivial identity function but also be highly dependent on model architecture and their depth. Our I3AD utilizes GAN image inpainting network and attempts to encode partial regions and decode selected image that is not in encoded regions.

As future works, our I3AD should be tested on different inpainting models going forward. The coarse-to-fine network with an attention module has a remarkable performance, requiring a large network structure. To reduce the model parameters for speeding up the inference and saving hardware costs, Sagong et al. (2019) proposes weight sharing on coarse and fine generator networks while maintaining performance.

6 Conclusion

In high-resolution images, autoencoders fail to model small details, which yields blurry image reconstructions. This is especially the case for high-frequency textures, such as carpet, leather, and tile. To tackle this issue, we propose the iterative image inpainting method to reconstruct partial regions in an adaptive manner. Our method utilizes the structural similarity measure for inpainting regions, which modifies their structural difference and enhances their reconstruction quality at a few iteration steps. Our method achieves state-of-the-art results on various AD datasets under a high-resolution condition and shows especially large improvement in the texture categories.

References

Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian conference on computer vision*, pp. 622–637. Springer, 2018.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *International MICCAI Brainlesion Workshop*, pp. 161–169. Springer, 2018.

Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv preprint arXiv:1807.02011*, 2018.

Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9592–9600, 2019.

David Dehaene, Oriel Frigo, Sébastien Combrexelle, and Pierre Eline. Iterative energy-based projection on a normal data manifold for anomaly localization. *arXiv preprint arXiv:2002.03734*, 2020.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Lewis D Griffin, Matthew Caldwell, Jerone TA Andrews, and Helene Bohler. "unexpected item in the bagging area": Anomaly detection in x-ray security images. *IEEE Transactions on Information Forensics and Security*, 14(6):1539–1553, 2018.

Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 85–100, 2018.

Takashi Matsubara, Ryosuke Tachibana, and Kuniaki Uehara. Anomaly machine component detection by deep generative model with unregularized score. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2018.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Min-cheol Sagong, Yong-goo Shin, Seung-wook Kim, Seung Park, and Sung-jea Ko. Pepsi: Fast image inpainting with parallel decoding network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11360–11368, 2019.

Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pp. 146–157. Springer, 2017.

Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical image analysis*, 54:30–44, 2019.

Dongyi Wang, Robert Vinson, Maxwell Holmes, Gary Seibel, Avital Bechar, Shimon Nof, and Yang Tao. Early detection of tomato spotted wilt virus by hyperspectral imaging and outlier removal auxiliary classifier generative adversarial nets (or-ac-gan). *Scientific reports*, 9(1):1–14, 2019.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5505–5514, 2018.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4471–4480, 2019.

Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

David Zimmerer, Fabian Isensee, Jens Petersen, Simon Kohl, and Klaus Maier-Hein. Unsupervised anomaly localization using variational auto-encoders. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 289–297. Springer, 2019a.

David Zimmerer, Jens Petersen, Simon AA Kohl, and Klaus H Maier-Hein. A case for the score: Identifying image anomalies using variational autoencoder gradients. *arXiv preprint arXiv:1912.00003*, 2019b.

A Anomaly Score of Structural similarity (SSIM Index)

For AD, some per-pixel error measure such as L^p-distance is utilized as anomaly score. The structural similarity (SSIM) metric (Wang et al., 2004) could be also employed to capture perceptual similarity(Bergmann et al., 2018; 2019). The SSIM Index defines a structural similarity 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 $s(p, q)$:

$$\mathrm{SSIM}(p, q) = l(p, q)^{\alpha} c(p, q)^{\beta} s(p, q)^{\gamma},$$

where $\alpha, \beta, \gamma \in R$ are hyperparameters. From the patches' mean intensities μ_p, μ_q, the patches' variances σ_p, σ_q, and their covariance σ_{pq}, the three measures are defined as

$$l(p, q) = \frac{2\mu_p\mu_q + c_1}{\mu_p^2 + \mu_q^2 + c_1}, \quad c(p, q) = \frac{2\sigma_p\sigma_q + c_2}{\sigma_p^2 + \sigma_q^2 + c_2}, \quad s(p, q) = \frac{2\sigma_{pq} + c_2}{2\sigma_p\sigma_q + c_2}.$$

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 three measures under equally weighting $\alpha = \beta = \gamma = 1$, the SSIM Index is derived by

$$\mathrm{SSIM}(p, q) = \frac{(2\mu_p\mu_q + c_1)(2\sigma_{pq} + c_2)}{(\mu_p^2 + \mu_q^2 + c_1)(\sigma_p^2 + \sigma_q^2 + c_2)}.$$

We have that $\mathrm{SSIM}(p, q) \in [-1, 1]$, and $\mathrm{SSIM}(p, q) = 1$ if and only if p and q are identical.

B MVTEC ANOMALY DETECTION DATASET

We performed experiments with the industrial dataset named MVTec Anomaly Detection (MVTec AD)(Bergmann et al., 2019). It contains the 5,354 high-resolution color images of ten different objects and five textures categories. It splits 3,629 images for training and validation and 1,725 images for testing. At test time, there are 467 defect-free images and 1,258 defect images. This dataset in total includes 73 different defect types, such as defects on the object's surface, structural defects, or the absence of certain parts. These anomalies are manually generated with the aim to produce realistic anomalies as they would occur in real-world industrial inspection scenarios. All image resolutions are set in the range between 700×700 and 1024×1024 pixels. We resize all images to 256×256 pixels for training and testing. This resolution is same with (Bergmann et al., 2019) for objects and different with previous studies (Bergmann et al., 2019; Dehaene et al., 2020) for textures. They subsampled the original images to 512×512 and crop random patches of size 128×128 for the textures.

C ANALOGY OF SSIM-MASK WITH PARTIAL CONVOLUTION LAYER

In inpainting task, the convolutional layer in a generator network assigns the weight balanced on both valid pixels in unmasked region and invalid pixels in masked region. (Liu et al., 2018) proposed partial convolution layer to adapt an irregular shaped mask and re-normalize the dependence of valid pixels. It is computed by the following equations:

$$O = \begin{cases} W * (X \odot \frac{M}{\mathrm{sum}(M)}) & \text{if} \quad \mathrm{sum}(M) > 0 \\ 0 & \text{otherwise,} \end{cases}$$

where let W be the convolution filter weights for the convolution layer, X be the feature values for the sliding window, M be the corresponding binary mask, and $O_{x,y}$ be the output features. The corresponding binary mask M has also the rule-based update in each layer following by

$$M^{l+1} = \begin{cases} 1 & \text{if} \quad \mathrm{sum}(M^l) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Our I3AD passes through the forward model iteratively, which is interpreted as a very deep generative model. Our SSIM-Mask corresponds to a kind of partial convolutions, which extends to non-linear structural similarity rule.

D MASK INITIALIZATION

During test iterations, the generative model needs the initialized mask at the first iteration step. To cover whole images and segment a few batches, we split images into X by X patches and aggregated them into four masks.

E QUALITATIVE RESULT

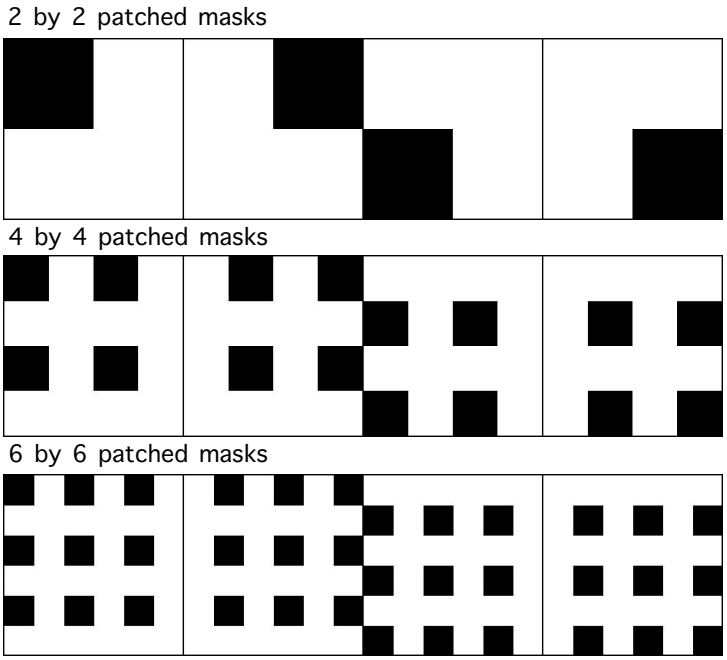

Figure 6: Mask initialization in test iteration step. Images are masked by four masks that includes X by X patched boxes.

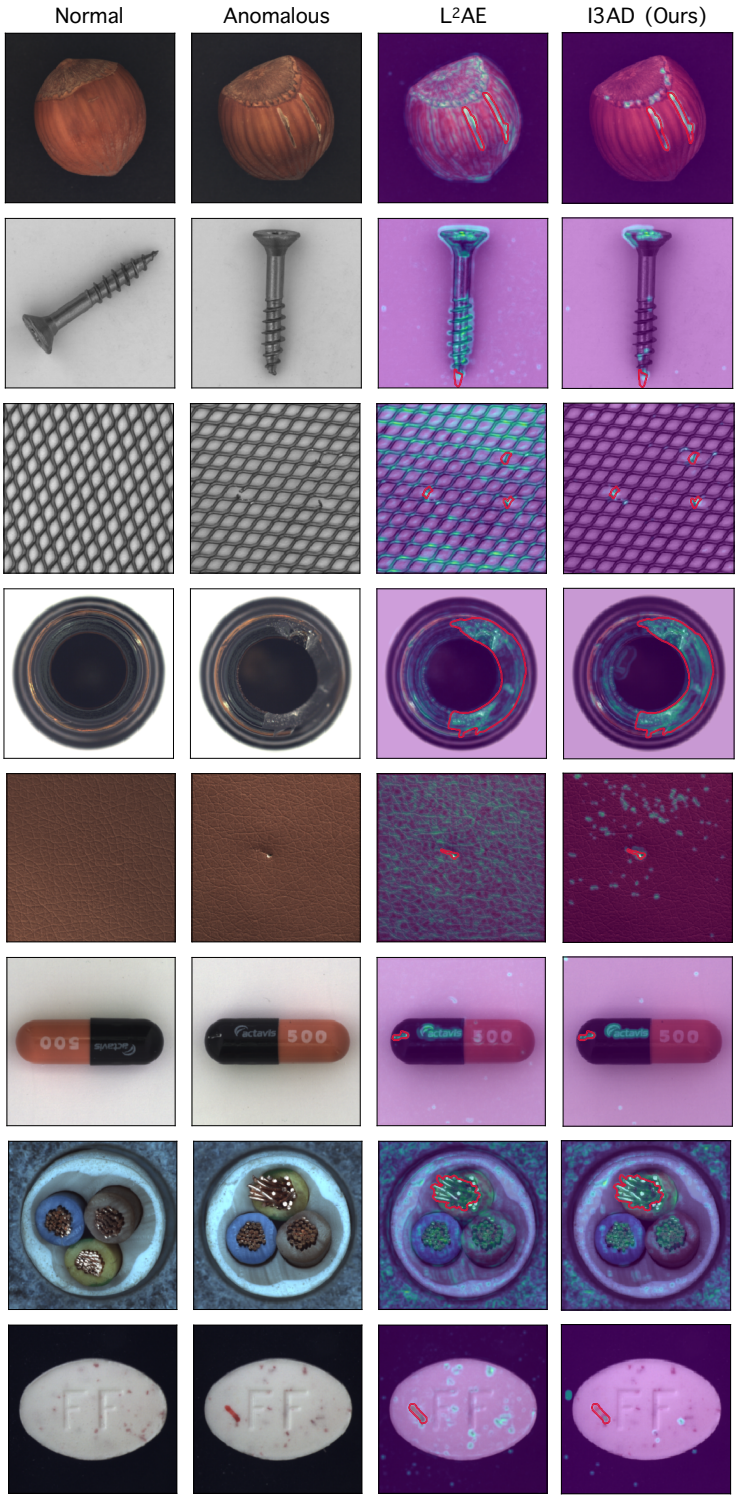

Figure 7: Other qualitative results