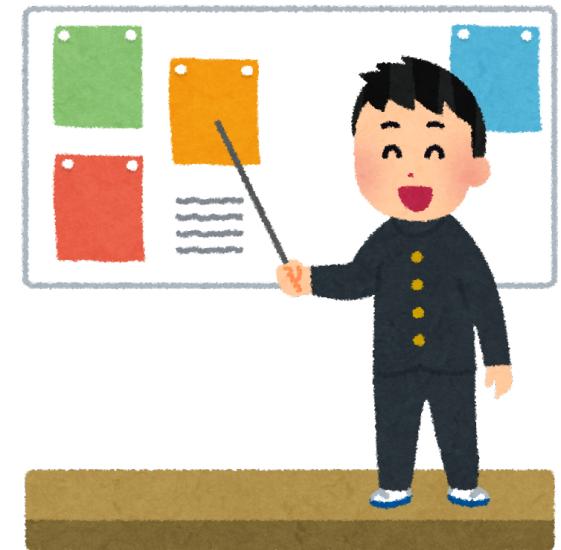


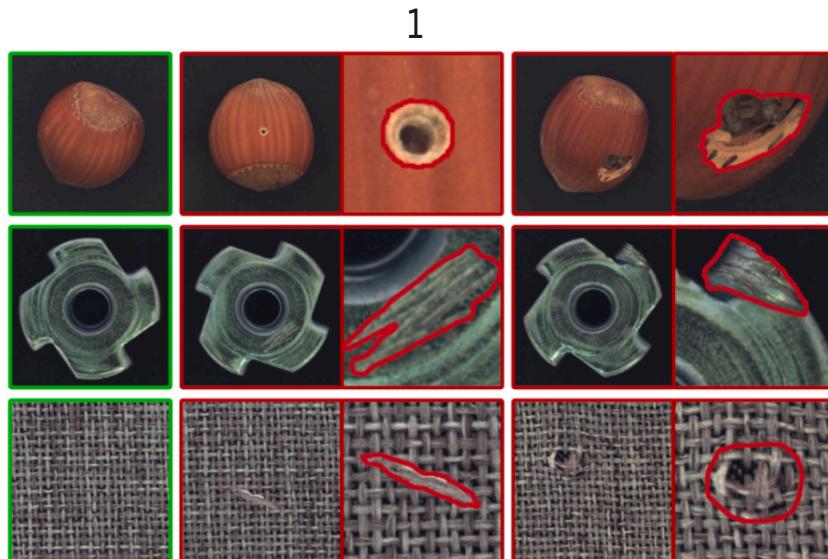
ICLR2020の異常検知論文を紹介 (2019/11/23)

ぱんさん@カーネル



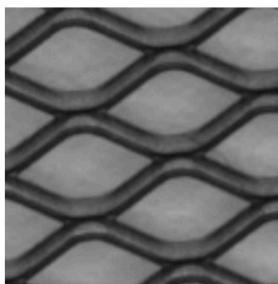
異常検知とは？

- 1: ある画像のよごれや傷などの異常を検知する (一般的な異常検知)
 - 今回紹介する論文はこちらです
- 2: 訓練分布のカテゴリ以外のカテゴリを検知する (Out-of-distribution 検知)
 - 似た問題設定
 - Open set recognition: In-distributionの分類 + OOD検知
 - Generalized zero-shot learning: In-distributionの分類 + OODの分類

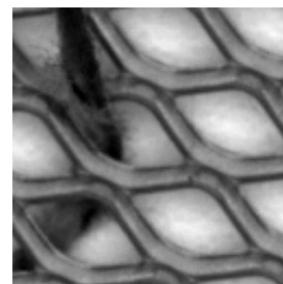


今回紹介する論文の概要

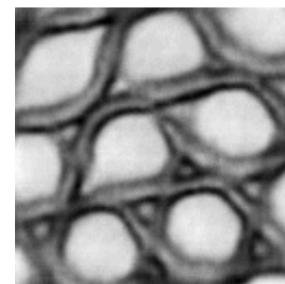
- タイトル: Iterative energy-based projection on a normal data manifold for anomaly localization
- 学会: ICLR 2020 (under review)
 - 点数: 8,6,3
- 一言で言うと:
 - 正常データのみを使ってAE(オートエンコーダ)ベースのモデルの学習を行ったあと、推論時に損失関数の勾配を利用することで、異常データがAEによって得られた正常データの多様体の最も近いところにマッピングされるように再構成し、その差をとることで異常箇所の特定を行う



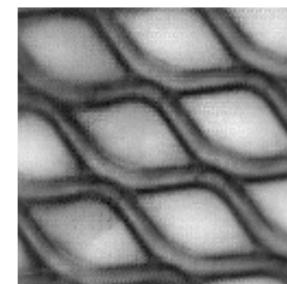
(a) Training sample



(b) Anomalous sample



(c) VAE reconstruction

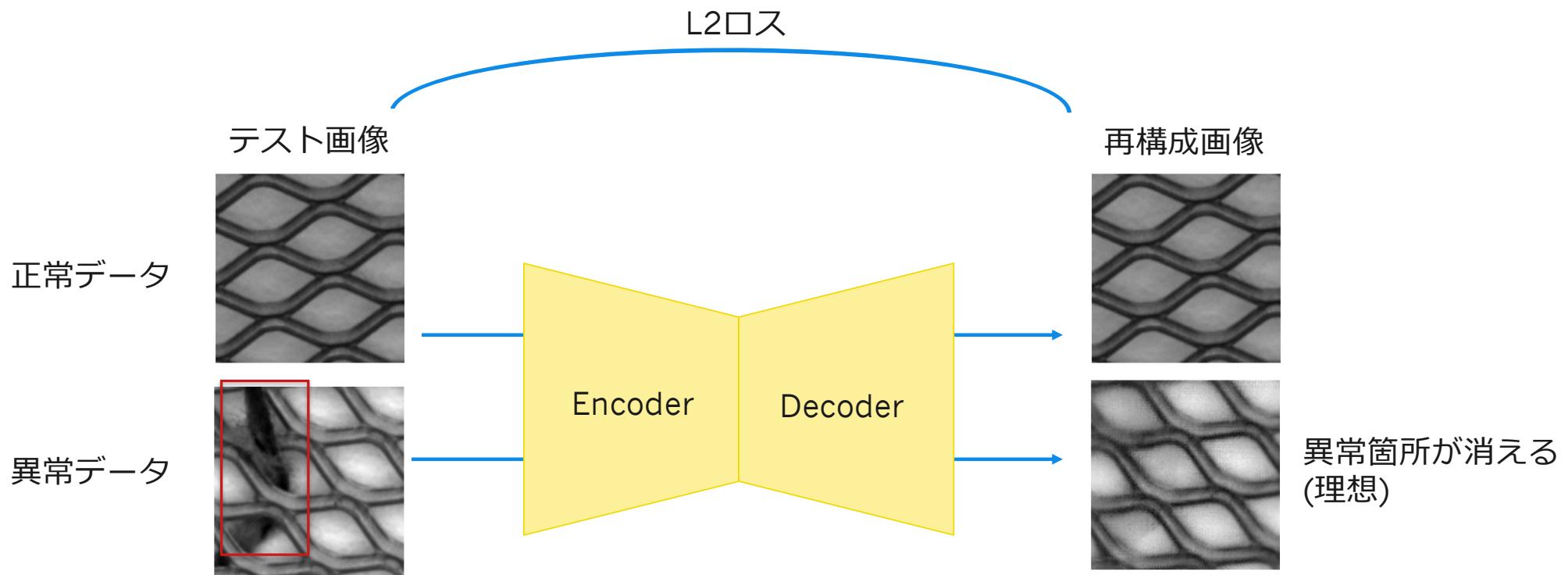


(d) Gradient-based projection

提案手法(d)は
異常箇所だけを
きれいに消している

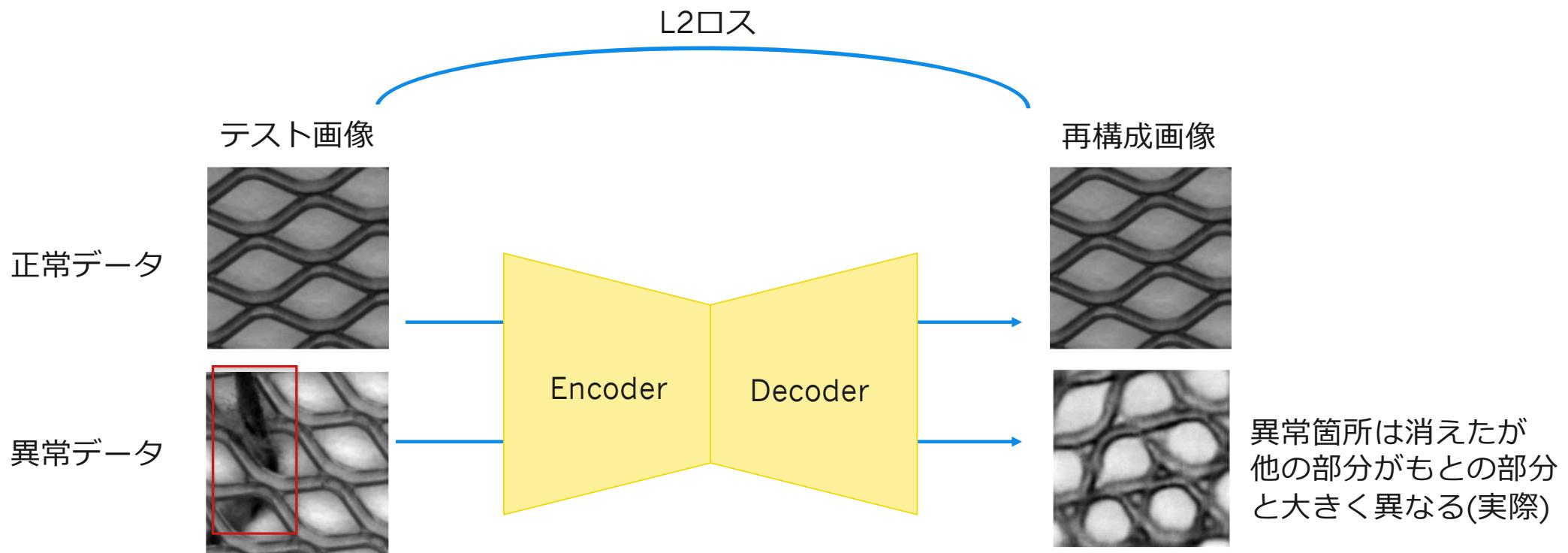
背景知識: 異常部位特定の一般的な手法

- オートエンコーダベースの生成モデルは正常データの多様体を学習すると仮定
- テスト時に異常データが入力されると、異常データは訓練時にはないので、元画像と最も近くなる正常データを出力する
- 元画像と再構成画像の差分を利用して異常部位を特定する



背景知識: 異常部位特定の一般的な手法

- しかし単純に再構成すると、異常データは異常箇所が原因で、異常箇所以外の部分でも元画像と異なる画像が出力されてしまう問題がある(もしくはぼやける)
- この場合、差分をとって異常箇所の検知を行うことができなくなる



関連手法: VAEの下限を使った異常検知

- ある閾値Tを決めて、VAEの下限を上回れば異常とする方法

$$\begin{aligned}\log p(\mathbf{x}) &= \log \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \\ &\geq \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}) - D_{\text{KL}}(q(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})) = -\mathcal{L}(\mathbf{x})\end{aligned}$$

- しかし、[Matsubara+ 2018]によれば、KL項は良くない影響を及ぼすため、再構成項だけを利用することを考えている

$$\mathcal{L}_r(\mathbf{x}) = -\mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z})$$

- まあ結局、VAEの再構成もぼやけるため、単純にVAEを使うのも微妙
 - もちろん再構成がぼやけないUnetのようなオートエンコーダを使っても、多様体学習しないからダメ(傷も復元されてしまう)

関連手法: GANを使った異常検知

- それでは、鮮明な画像を出力するためにGANを使えばよいのでは?
→ AnoGAN [Schlegl+ 2017]
 - 正常画像で学習しているため、異常画像は生成できないという考えに基づく
 - しかし、GANは画像から潜在変数 z を求めることはできない
 - そのため、異常画像から以下の式を使うことにより、異常画像と最も近い正常画像を出力するような z を推定する(これがそのまま異常スコアとなる)
 - f_D は識別器の中間層の出力

$$E_{AnoGAN} = \|\mathbf{x} - G(\mathbf{z})\|_1 + \lambda \cdot \|f_D(\mathbf{x}) - f_D(G(\mathbf{z}))\|_1$$

どの z がgeneratorに
近い元画像を出力させるか この z がdiscriminatorに
近い元画像に対する判定を行わせるか

- 今回の提案手法は、オートエンコーダベースのモデルを使い、 z の空間で近い正常画像を探すのではなく、 x の空間で探す

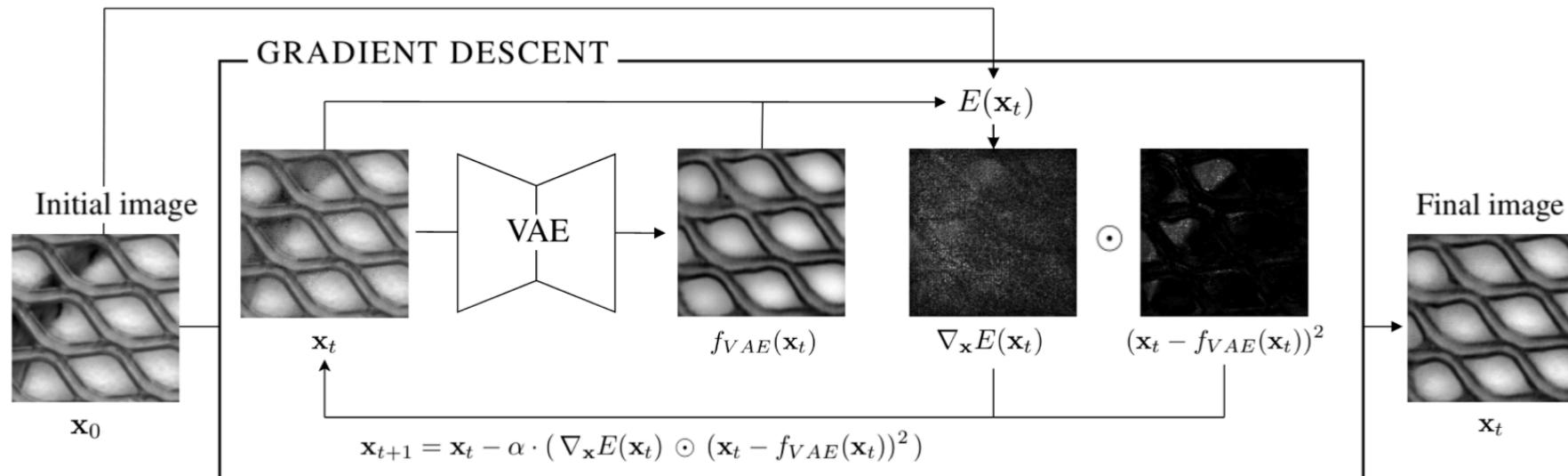
提案手法: エネルギー関数を利用した正常データの多様体への射影

- エネルギー関数の勾配を利用して、元画像を作り変えていく

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha \cdot \nabla_{\mathbf{x}} E(\mathbf{x}_t),$$
 - 敵対的サンプルを作る方法と近い(敵対的ではないが)
- エネルギー関数 = 再構成項 + 正則化項

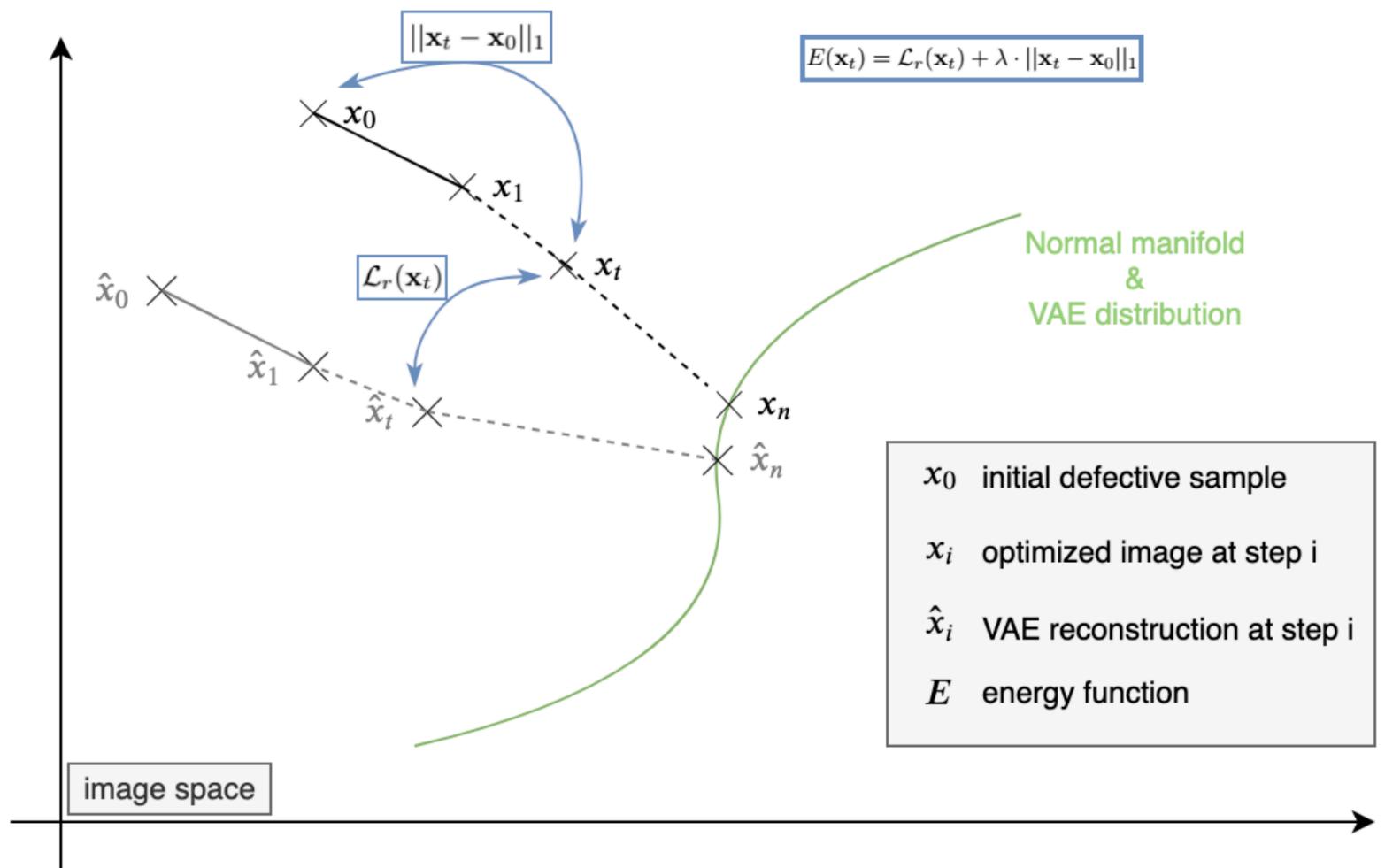
$$E(\mathbf{x}_t) = \mathcal{L}_r(\mathbf{x}_t) + \lambda \cdot \|\mathbf{x}_t - \mathbf{x}_0\|_1$$
 - 再構成項: 傷がない正常画像に近づける
 - 正則化項: もとの画像から離れすぎないようにする
- 特に誤差が大きいところを更新するようにすると、収束が早くなる

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha \cdot (\nabla_{\mathbf{x}} E(\mathbf{x}_t) \odot (\underline{\mathbf{x}_t - f_{VAE}(\mathbf{x}_t)})^2)$$



提案手法: イメージ

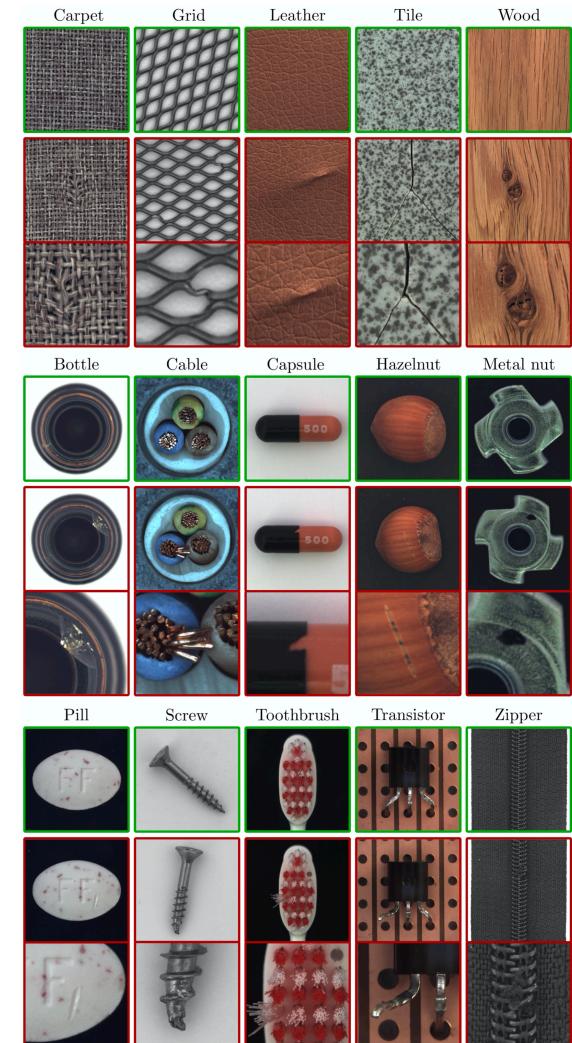
- 元画像から離れすぎない程度に勾配更新を繰り返すことで、正常画像の多様体に近づけていく



データセット: MVTec AD (CVPR2019)

- 15種類のカテゴリーの異常検知用データセット
 - 訓練データ: 正常データのみ
 - テストデータ: 正常データ + 異常データ

Category	# Train	# Test (good)	# Test (defective)	# Defect groups	# Defect regions	Image side length
Textures	Carpet	280	28	5	97	1024
	Grid	264	21	5	170	1024
	Leather	245	32	5	99	1024
	Tile	230	33	5	86	840
	Wood	247	19	5	168	1024
Objects	Bottle	209	20	3	68	900
	Cable	224	58	8	151	1024
	Capsule	219	23	5	114	1000
	Hazelnut	391	40	4	136	1024
	Metal Nut	220	22	4	132	700
	Pill	267	26	7	245	800
	Screw	320	41	5	135	1024
	Toothbrush	60	12	1	66	1024
	Transistor	213	60	4	44	1024
	Zipper	240	32	7	177	1024
Total		3629	467	73	1888	-

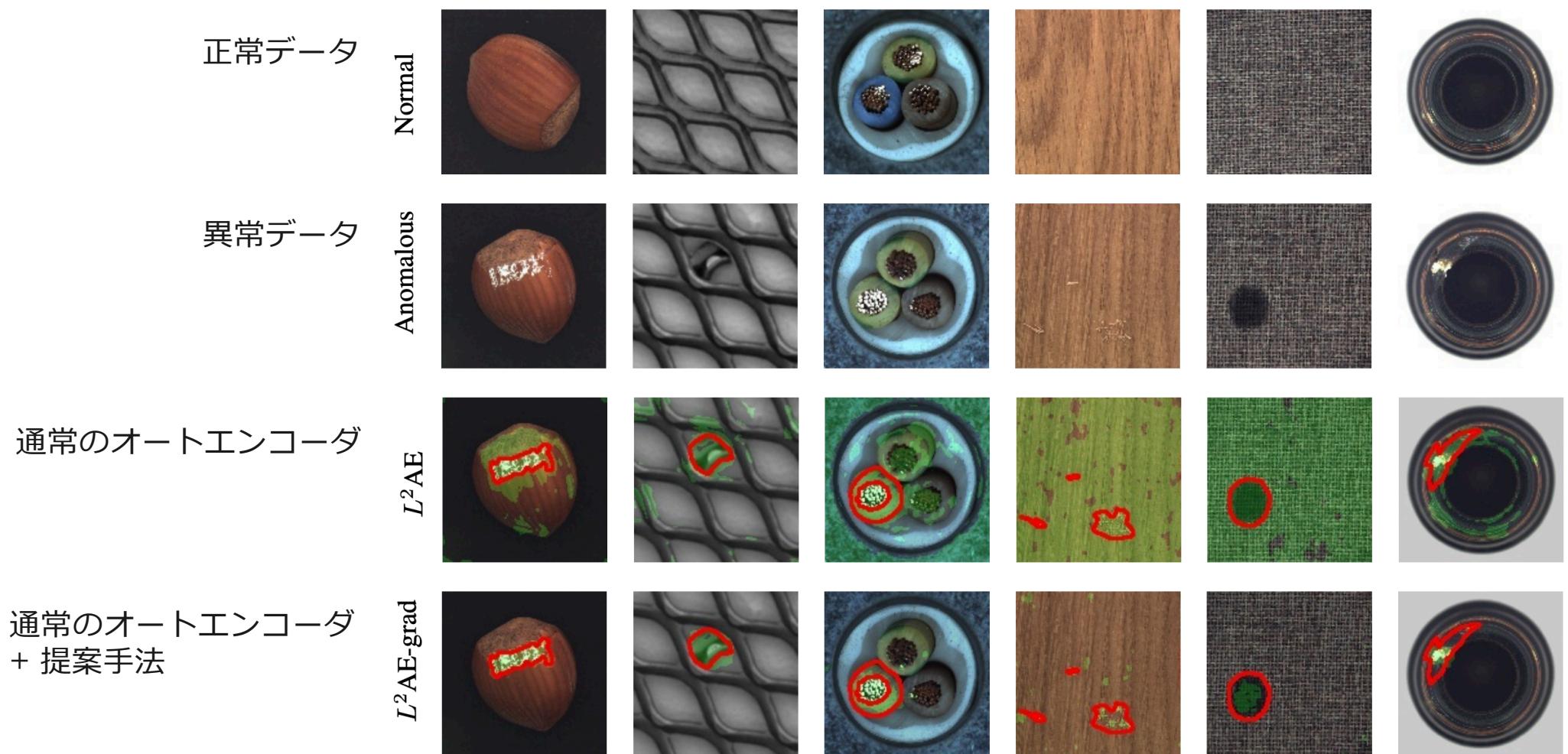


定量的結果: AUROCによる比較

- ベースラインに追加した提案手法(grad)によるプロセスが効果的であることを示している
 - 緑: 元手法に比べて良くなる, 赤: 悪くなる, 太字: 全ての手法で最もAUROCが高い

Category	L^2 AE	L^2 AE-grad	DSAE	DSAE-grad	VAE	VAE-grad	γ -VAE	γ -VAE-grad
Textures	carpet	0.539	0.734	0.545	0.774	0.580	0.735	0.648
	grid	0.960	0.981	0.960	0.980	0.888	0.961	0.950
	leather	0.751	0.921	0.710	0.602	0.834	0.925	0.818
	tile	0.476	0.575	0.496	0.626	0.465	0.654	0.491
	wood	0.630	0.805	0.641	0.738	0.695	0.838	0.665
Objects	bottle	0.909	0.916	0.933	0.951	0.902	0.922	0.913
	cable	0.732	0.864	0.790	0.859	0.828	0.910	0.777
	capsule	0.786	0.952	0.769	0.884	0.862	0.917	0.814
	hazelnut	0.976	0.984	0.966	0.966	0.977	0.976	0.977
	metalfnut	0.880	0.899	0.881	0.920	0.881	0.907	0.883
	pill	0.885	0.912	0.895	0.927	0.888	0.930	0.897
	screw	0.979	0.980	0.983	0.925	0.958	0.945	0.976
	toothbrush	0.971	0.983	0.973	0.984	0.971	0.985	0.971
	transistor	0.906	0.921	0.904	0.934	0.894	0.919	0.896
	zipper	0.680	0.889	0.828	0.887	0.814	0.869	0.706

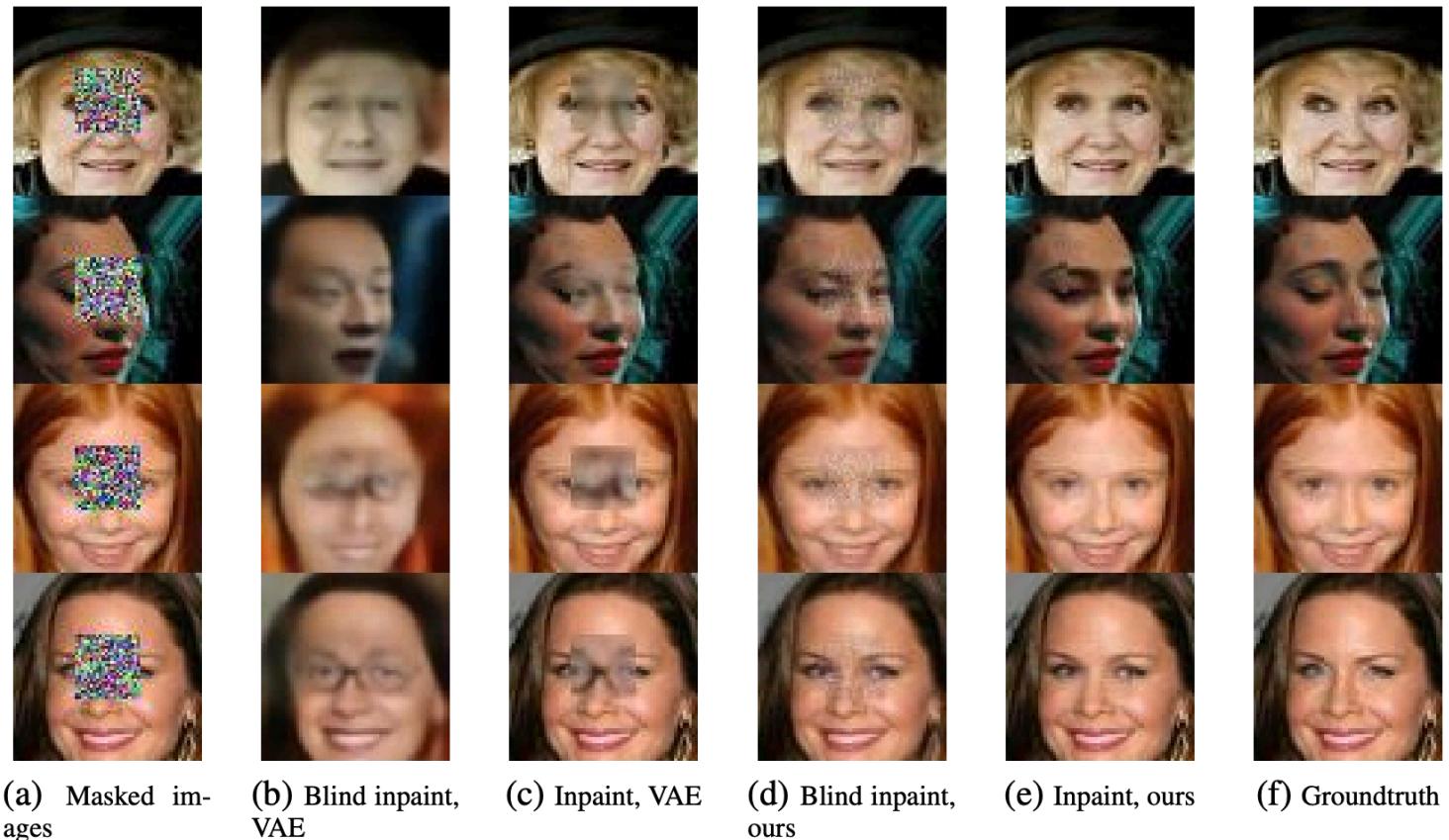
定性的結果: 異常箇所特定



緑は予測された異常箇所・赤は実際の異常箇所を示す

定性的結果: 画像補完

- 実はmask画像復元にも使える(こちらのほうが異常箇所がわかっている分簡単な問題設定)
 - mask箇所のみを更新する $\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha \cdot (\nabla_{\mathbf{x}} E(\mathbf{x}_t) \odot \Omega)$



実際に実装してみた

- 提案手法は、単純にVAEで再構成するよりも、ぼやけるのを防ぎつつ異常箇所を消しているのがわかる
- 実装記事: <https://qiita.com/kogepan102/items/122b2862ad5a51180656>

