

産業用異常検知における全回収に向けて

Karsten Roth^{1,*}、Latha Pemula²、Joaquin Zepeda²、Bernhard Schölkopf²、Thomas Brox²、Peter Gehler²。¹ユービンゲン大学²アマゾンAWS

要旨

Being able to spot defective parts is a critical component in large-scale industrial manufacturing. A particular challenge that we address in this work is the cold-start problem: fit a model using nominal (non-defective) example images only. While handcrafted solutions per class are possible, the goal is to build systems that work well simultaneously on many different tasks automatically. The best performing approaches combine embeddings from ImageNet models with an outlier detection model. In this paper, we extend on this line of work and propose **PatchCore**, which uses a maximally representative memory bank of nominal patch-features. PatchCore offers competitive inference times while achieving state-of-the-art performance for both detection and localization. On the challenging, widely used MVTEC AD benchmark PatchCore achieves an image-level anomaly detection AUROC score of up to 99.6% , more than halving the error compared to the next best competitor. We further report competitive results on two additional datasets and also find competitive results in the few samples regime. Code: github.com/amazon-research/patchcore-inspection.

1.はじめに

画像中の異常なパターンを検出する能力は、人間の認知に深く刻み込まれた特徴である。人間は、少数の正常な事例を見ただけで、データ中の予想される分散と異常値を区別することができる。本研究では、この問題の計算バージョンであるcold-start¹工業用画像データの視覚検査における異常検出を扱う。この問題は、正常な例の画像を取得するのは簡単だが、予想される欠陥のばらつきを完全に指定するのはコストがかかり複雑であるという、多くの産業シナリオで発生する。このタスクは、モデルが学習データ分布から抽出されたサンプルと、そのサポート外のサンプルを区別する必要がある、分布外検出問題として自然に扱われる。工業的な目視による欠陥分類は、特に困難である。

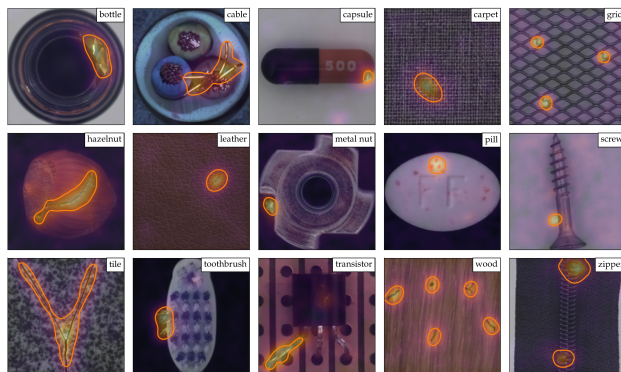


図1.MVTECベンチマークデータセットの例。画像にスーパーインポーズされたのは、Patch-のセグメンテーション結果である。Core.オレンジ色の境界線は、実際のセグメンテーション・マップの異常輪郭を示し、ガラスの破損、傷、火傷、青とオレンジのカラー・グラデーションの構造変化などの異常を示す。

薄い傷のような微妙な変化から、部品の欠落のような大きな構造的欠陥まで様々です[5]。MVTEC ADベンチマークの例と、提案手法の結果を図1に示す。コールドスタートの産業用視覚異常検出に関する既存の研究は、自動エンコード法[12, 36, 44]、GAN[2, 39, 43]、または他の教師なし適応法[42, 56]を用いた公称分布のモデルの学習に再依存する。最近、[4, 10]は、ImageNetの分類から、ターゲット分布に適応することなく、一般的な深層表現を活用することを提案した。適応が欠落しているにもかかわらず、これらのモデルは強力な異常検出性能と、欠陥の確かな空間的局在化さえ実現する。これらの手法の背後にある重要な原理は、深い特徴表現のマルチスケルな性質を利用しながら、テストサンプルとノミナルサンプルの特徴マッチングを行うことである。微妙で粒度の細かい欠陥のセグメンテーションは高解像度特徴によってカバーされるが、構造的な偏差や完全な画像レベルの異常検出は、より抽象度の高い特徴によってカバーされる。このアプローチは非適応的であるため、高い抽象度での一致度が限定的であるという欠点がある。

* Work done during a research internship at Amazon AWS.

¹Commonly also dubbed one-class classification (OCC).

には、産業環境で必要とされる抽象的な特徴がない。さらに、テスト時にこれらの手法で使用できる名目的なコンテキストは、処理可能な高レベルの特徴表現の数が少ないため、事実上制限されています。

本論文では、(1)テスト時に利用可能な名目情報を最大化し、(2)ImageNetクラスへのバイアスを低減し、(3)高速推論を維持することで、PatchCoreを効果的な救済策として提示する。画像は1つのパッチが異常である時点で既に異常であると分類できるという事実[14, 56]に依拠し、Patch-Coreはこれを実現します。Coreは局所的に集約された中間レベルの特徴パッチを利用することでこれを達成する。中レベルのネットワークパッチ特徴の利用により、PatchCoreは高解像度のImageNetクラスに対するバイアスを最小限に抑えて動作することができ、一方で局所近傍の特徴集約は十分な空間的コンテキストの保持を保証する。この結果、{v4}はテスト時に利用可能な公称コンテキストを最適に活用することができる、拡張メモリバンクとなる。最後に、PatchCoreは実用性のために、抽出されたパッチレベルのメモリバンクの冗長性を減らすと同時に、記憶メモリと推論時間を大幅に削減するための重要な要素として、名目特徴バンクに貪欲なコアセットサブサンプリング[1]を導入しており、Patch-Coreを非常に魅力的なものにしています。v6}Coreは現実的な産業ユースケースにとって非常に魅力的です。

多様なMVTec AD [5]と特殊な磁気タイル欠陥 (MTD) [26]の産業用異常検出ベンチマークでの徹底的な実験により、産業用異常検出のためのPatchCoreの威力を示す。v8}はMVTec ADとMTDで最先端の画像レベル検出スコアを達成し、MVTec ADではほぼ完璧なスコア (最高 AUROC 99.6%) を達成し、従来の手法の検出エラーを半分以下に低減し、さらに産業用異常のローカライズ性能も最先端です。Patch-Coreは、手元のデータセットでの学習を必要とせず、高速な推論時間を維持しながらこれを達成します。このため、PatchCoreは産業用異常検出の実用化において非常に魅力的である。さらに、更なる実験により、PatchCoreの高いサンプル効率が示され、公称学習データのほんの一部しか使用しないにもかかわらず、性能において他の異常検出手法に匹敵する。

2. 関連作品

ほとんどの異常検知モデルは、名目データに固有の表現を学習する能力に依存している。これは例えば、自動符号化モデル[44]を使用することで実現できる。名目特徴分布のより良い推定を促すために、ガウス混合モデル[60]、生成的敵対的学習目標[2,39,43]、事前定義された物理的補強に対する不変性[25]、再構成の再導入に対する隠れ特徴の頑健性[29]、原型的メモリバンク[21]、注意誘導[52]、構造目標[7,59]、または制約付き表現空間[38]に基づく拡張が提案されている。

ポーズをとる。GAN[13]、幾何学的変換の予測学習[20]、正規化フロー[42]など、他の教師なし表現学習法も同様に利用できる。それぞれの公称表現と新しいテスト表現が与えられれば、異常検出は単純に再構成誤差[44]、 k 最近傍との距離[18]、またはOC-SVM[46]やSVDD[50, 56]のような1クラス分類モデルをこれらの特徴の上でネットチューニングすることができる。これらのアプローチの大部分では、異常のローカライゼーションはピクセル単位の再構成誤差に基づいて自然に行われるが、GradCAM [47]やXRAI [28]のような顕著性ベースのアプローチも異常のセグメンテーションに使用できる[42, 45, 52]。

産業用異常検知。学習された公称表現による一般的な異常検知に関する文献は膨大であるが、産業用画像データには独自の課題があり [5]、[4]を始めとする最近の研究では、手元のデータに適應することなく、ImageNet [16]のような大規模な外部自然画像データセットで事前学習されたモデルを使用して、最先端の検知性能を示している。これは、finegrained, k NN-based [18] anomaly segmentation and image-level anomaly detectionのために様々な特徴階層からなるメモリバンクを利用するSPADE [10]のような、事前学習された特徴のより良い再利用に依存する他の産業的な異常検出方法を生み出した。同様に[14]は最近PaDiMを提案したが、これは局所的に制約されたBag-of-Featuresアプローチ[8]を利用し、パッチレベルのマハラノビス距離測定[33]に対してパッチレベルの特徴分布モーメント (平均と共分散) を推定する。このアプローチはフル画像で研究された[40]に似ている。例えば、[6, 45]のような生徒-教師知識差分[24]や、事前学習されたネットワーク特徴量[42]の上に学習されたフロー[17, 30]を正規化する。

v15}で使用されている特定のコンポーネントは、SPADEとPaDiMに最も関連している。SPADEは、画像レベルとピクセルレベルの異常検出のための別々のアプローチで、事前に訓練されたバックボーン・ネットワークから抽出された名目的な特徴のメモリバンクを利用する。PatchCoreも同様にメモリバンクを使用するが、より多くの名義コンテキストが保持され、より優れた誘導バイアスが組み込まれるため、より高いパフォーマンスを達成するためには、近隣を意識したパッチレベルの特徴が重要である。さらに、メモリバンクはコアセット・サブサンプリングされ、より高いパフォーマンスで低い推論コストを保証する。コアセットは、基本的なkNNやkMeansアプローチ[22]、あるいは混合モデル[19]において、利用可能な集合の構造に最も近似する部分集合を見つけ、顕著に削減されたコストで近似解を見つけることを可能にすることにより、長年使用されてきた[1, 9]。さらに最近では、コアセットに基づく手法は、ネットワークの刈り込み[34]、能動学習[48]、効果的なデータの増加など、深層学習アプローチにも応用されている。

GAN学習[49]や表現学習[41]を改善するためのミニパッチのカバレッジ。後者の3つは、貪欲なコアセット選択メカニズムを利用して成功を収めている。我々はメモリバンクの特徴空間のカバレッジを近似することを目的としているので、PatchCoreについても同様に貪欲なコアセットメカニズムを適応する。最後に、画像レベルの異常検出と異常セグメンテーションの両方に対する我々のパッチレベルアプローチは、より高い異常検出感度を奨励する目的でPaDiMに関連している。PaDiMがパッチレベルの異常検出を各パッチに特有なマハラノビス距離測定に限定するのに対して、我々はテスト時に評価される全てのパッチに等しくアクセス可能な効率的なパッチ特徴メモリバンクを利用する。そうすることで、PatchCoreは画像のアライメントに依存しなくなる一方で、より大きな名目的なコンテキストを用いて異常を推定する。さらに、PaDiMとは異なり、入力画像は学習時とテスト時に同じ形状を必要としません。最後に、PatchCoreは局所的な空間分散を考慮し、ImageNetクラスへの偏りを減らすために、緩やかに認識されたパッチ特徴スコアを利用します。

3.方法

v3}メソッドはいくつかの部分から構成されており、順を追って説明する：メモリバンクに集約されたローカルパッチ特徴 (§3.1)、効率性を高めるためのコアセット削減法 (§3.2)、そして最後に検出とローカライゼーションの決定に至る完全なアルゴリズム (§3.3)。

3.1.ローカル・アウェア・パッチの特徴

v4}を用いて、学習時に利用可能な全ての公称画像 ($\forall x \in \mathcal{X}_N: y_x = 0$) の集合を表し、 $y_x \in \{0, 1\}$ は、画像 x が公称 (0) か異常 (1) かを表す。1}は画像 x が公称 (0) か異常 (1) かを表す。従って、 \mathcal{X}_T をテスト時に提供されるサンプルの集合と定義し、 $\forall x \in \mathcal{X}_T: y_x \in \{0, 1\}$ とする。1}.4}、[10]、[14]に従い、PatchCoreはImageNetで事前に訓練されたネットワーク ϕ を使用する。特定のネットワーク階層における特徴が重要な役割を果たすので、データセット \mathcal{X} と事前訓練されたネットワーク ϕ の階層レベル j を持つ画像 $x_i \in \mathcal{X}$ の特徴を表すために $\phi_{i,j} = \phi_j(x_i)$ を使用する。特に断りがなければ、既存の文献と一致するように、 j は、ResNet-50やWideResnet-50 [57]などのResNet-ライクな[23]アーキテクチャからの特徴マップをインデックス化し、 $j \in \{1, 2, 3\}$ {v283.4}はそれぞれの空間分解能ブロックの最終出力を示す。

特徴表現の一つの選択肢は、ネットワークの特徴階層の最後のレベルである。これは[4]や[10]で行われているが、以下の2つの問題がある。第一に、より局所的な名目情報[14]が失われる。テスト時に遭遇する異常の種類 *a priori* は不明であるため、これは下流の異常検出性能に不利となる。第二に、ImageNetの事前学習されたネットワークに含まれる非常に深く抽象的な特徴は、自然画像の分類というタスクに向けてバイアスされており、これはコールドスタートの産業用異常検知タスクや手元の評価データとほとんど重ならない。

従って、我々は *intermediate* または *mid-level* 特徴表現からなるパッチレベル特徴のメモリバンク M を使用して、提供されたトレーニングコンテキストを利用することを提案し、一般的すぎる特徴やImageNet分類に偏りすぎた特徴を避ける。ResNetのようなアーキテクチャの場合、これは例えば $j \in \{2, 3\}$ パッチ表現を公式化するために、先に紹介した表記法を拡張する。特徴マップ $\phi_{i,j} \in \mathbb{R}^{c^* \times h^* \times w^*}$ を深さ c^* 、高さ h^* 、幅 w^* の3次元テンソルと仮定する。そして、 $\phi_{i,j}(h, w) = \phi_j(x_i, h, w) \in \mathbb{R}^{c^*}$ を用いて、位置 $h \in \{1, \dots, h^*\}$ における c^* 次元特徴スライスを表す。v43} $\{1, \dots, h^*\}$ と $w \in \{1, \dots, w^*\}$ の c^* 次元特徴スライスを表す。各 $\phi_{i,j}$ の受容fieldサイズが1より大きいと仮定すると、これは実質的に画像パッチ特徴表現に関係する。理想的には、各パッチ表現は、局所的な空間変化に対する意味のある変則的なコンテキストのローバストを説明するのに十分な大きさの受容fieldサイズで動作する。これはプーリングによって達成され、さらにネットワーク階層を下降させることができるが、それによって作成されたパッチ特徴はよりImageNet特有のものとなり、したがって目下の異常検出タスクとの関連性が低くなる。

このことは、空間分解能や特徴マップの使いやすさを失うことなく、受容閾値の大きさと小さな空間的变化に対する頑健性を向上させるために、各パッチレベル特徴表現を構成する際に局所近傍集計を行う動機付けとなる。そのために、上記の $\phi_{i,j}(h, w)$ の表記を拡張して、近傍サイズ)に対応する不均一なパッチサイズ p を考慮し、近傍からの特徴ベクトルを取り込む。

$$\mathcal{N}_p^{(h,w)} = \{(a, b) | a \in [h - \lfloor p/2 \rfloor, \dots, h + \lfloor p/2 \rfloor], b \in [w - \lfloor p/2 \rfloor, \dots, w + \lfloor p/2 \rfloor]\}, \quad (1)$$

として、位置 (h, w) の局所的な特徴を認識する。

$$\phi_{i,j}(\mathcal{N}_p^{(h,w)}) = f_{\text{agg}} \left(\{\phi_{i,j}(a, b) | (a, b) \in \mathcal{N}_p^{(h,w)}\} \right), \quad (2)$$

v54}は、近傍 $\mathcal{N}_p^{(h,w)}$ の特徴ベクトルの集約関数です。v56}については、適応平均プーリングを用いる。これは個々の特徴マップに対する局所平滑化と同様であり、事前に定義された次元 d の (h, w) における単一の再提示をもたらす。これは $h \in \{1, \dots, h^*\}$ と $w \in \{1, \dots, w^*\}$ を持つ全てのペア (h, w) に対して実行されるため、特徴マップの解像度を保持する。特徴マップテンソル $\phi_{i,j}$ に対して、その局所的に認識されたパッチ特徴コレクション $\mathcal{P}_{s,p}(\phi_{i,j})$ は

$$\mathcal{P}_{s,p}(\phi_{i,j}) = \{\phi_{i,j}(\mathcal{N}_p^{(h,w)}) | h, w \bmod s = 0, h < h^*, w < w^*, h, w \in \mathbb{N}\}, \quad (3)$$

v67}をオプションで使用し、4.4.2節で行ったアブレーション実験以外は1に設定した。経験的に、また[10]や[14]と同様に、我々は以下の凝集を発見した。

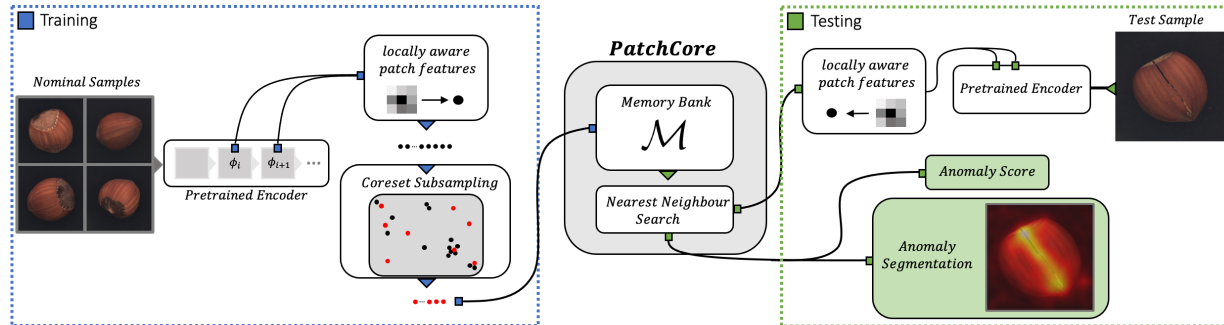


図2.v0)の概要。公称サンプルは近傍認識パッチレベル特徴のメモリバンクに分解される。冗長性と推論時間を削減するために、このメモリバンクは貪欲なコアセットサブサンプリングによってダウンサンプリングされる。テスト時には、少なくとも1つのパッチが異常であれば画像は異常と分類され、各パッチ特徴量をスコアリングすることでピクセルレベルの異常セグメンテーションが生成される。

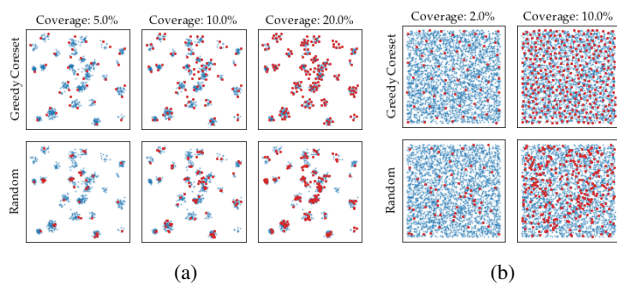


図3.比較：(a)マルチモーダル分布と(b)一様分布からサンプリングされた2Dデータ(青)のコアセット(上)とランダムサブサンプリング(下)(赤)。視覚的に、コアセットのサブサンプリングは空間サポートをより良く近似し、ランダムサブサンプリングはマルチモーダルなケースではクラスタを見逃し、(b)では均一性が低い。

複数の特徴階層を使用することで、いくつかの利点がある。しかし、使用される特徴の一般性を保持し、スパ分解能を維持するために、PatchCoreは2つの中間的な特徴階層 j と $j+$ のみを使用します。1.v7)と $|\mathcal{P}_{s,p}(\phi_{i,j})|$ が一致するように $\mathcal{P}_{s,p}(\phi_{i,j+1})$ をバイ・リニアにリスケーリングすることによって達成される。

最後に、全ての公称トレーニング・サンプル $x_i \in \mathcal{X}_N$ に対してv10)のメモリバンク M は次のように定義されます。

$$M = \bigcup_{x_i \in \mathcal{X}_N} \mathcal{P}_{s,p}(\phi_j(x_i)). \quad (4)$$

3.2. コアセットを削減したパッチ・フィーチャー・メモリー・バンク

v13)のサイズが大きくなるとv14)は非常に大きくなり、それに伴って新しいテストデータを評価する推論時間と必要なストレージの両方が増大する。この問題は、低レベル特徴マップと高レベル特徴マップの両方を利用する、異常セグメンテーションのためのSPADE [10]で既に指摘されている。SPADEでは、計算量に限界があるため、事前に特徴マップを作成する必要がある。

ピクセルレベルの異常検出のための特徴マップの編集段階は、全画像の深い特徴表現に依存する弱い画像レベルの異常検出メカニズム、すなわち最後の特徴マップの大域的平均化に基づいている。この結果、低解像度でImageNetに偏った表現が全画像から計算されることになり、検出とローカライゼーションの性能に悪影響を与える可能性がある。

これらの問題は、 M の意味を、より大きな画像サイズとカウントに対して完全に検索可能にし、異常検出とセグメンテーションの両方に有益なパッチベースの比較を可能にすることで対処できる。このためには M でエンコードされた公称特徴カバレッジを保持する必要がある。残念なことに、ランダムサブサンプリングは、特にいくつかの倍率によるサブサンプリングは、名目特徴のカバレッジにコード化された M で利用可能な有意な情報を失うことになる(4.4.2節で行われた実験も参照)。この研究では、 M を削減するためにコアセットサブサンプリングメカニズムを使用する。

概念的には、コアセットの選択は、 $S \subset A$ の部分集合を見つけることを目的とする。 A 上の問題解が S 上の問題解に最も近く、特に速く近似できるような部分集合 A を見つけることである。[1]. 特定の問題によって、関心のあるコアセットは異なります。v23)は最近傍計算を使用するため(次のセクション)、*minimax facility location*のコアセットを使用します。*facility location* コアセット選択(例えば、[48]及び[49]を参照)を用いて、パッチレベル特徴空間における M コアセット M_C の、元のメモリバンク M と比較したほぼ同様のカバレッジを確保する。

$$M_C^* = \arg \min_{M_C \subset M} \max_{m \in M} \min_{n \in M_C} \|m - n\|_2. \quad (5)$$

v30)の厳密な計算はNP-Hard [54]であるため、[48]で提案されている反復的な貪欲な近似を使用する。さらにコアセットの選択時間を短縮するために、我々は[49]に従い、Johnson-Lindenstraussの定理[11]を利用して、ランダムな線形射影 $\psi: \mathbb{R}^d \rightarrow \mathbb{R}^{d^*}$ with $d^* < d$ による要素 $m \in M$ のディメンジョン性を削減する。メモリバンクの削減はアルゴリズム1にまとめられている。表記のために

v0}%を使用して、元のメモリバンクがサブサンプリングされた n のパーセンテージを表します。Core-1%は M の100倍縮小。図3は、ランダム選択と比較した、貪欲なコアセットサブサンプリングの空間カバレッジの視覚的な印象を与える。

Algorithm 1: PatchCore memory bank.

Input: Pretrained ϕ , hierarchies j , nominal data \mathcal{X}_N , stride s , patchsize p , coresets target l , random linear projection ψ .

Output: Patch-level Memory bank \mathcal{M} .

Algorithm:

```

 $\mathcal{M} \leftarrow \{\}$ 
for  $x_i \in \mathcal{X}_N$  do
  |  $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{P}_{s,p}(\phi_j(x_i))$ 
end
/* Apply greedy coresets selection. */
 $\mathcal{M}_C \leftarrow \{\}$ 
for  $i \in [0, \dots, l-1]$  do
  |  $m_i \leftarrow \arg \max_{m \in \mathcal{M} - \mathcal{M}_C} \min_{n \in \mathcal{M}_C} \|\psi(m) - \psi(n)\|_2$ 
  |  $\mathcal{M}_C \leftarrow \mathcal{M}_C \cup \{m_i\}$ 
end
 $\mathcal{M} \leftarrow \mathcal{M}_C$ 

```

3.3.v6}による異常検知

公称パッチ特徴メモリバンク $\{v7\}$ を用いて、テスト画像 $\{v9\}$ に対する画像レベル異常スコア $\{v8\}$ を、そのパッチコレクション $\{v11\}$ 内のテストパッチ特徴と $\{v13\}$ 内のそれぞれの最近傍 $\{v12\}$ との間の最大距離スコア $\{v10\}$ によって推定する：

$$m^{\text{test},*}, m^* = \arg \max_{m^{\text{test}} \in \mathcal{P}(x^{\text{test}})} \arg \min_{m \in \mathcal{M}} \|m^{\text{test}} - m\|_2 \quad (6)$$

$$s^* = \|m^{\text{test},*} - m^*\|_2.$$

v15}を得るために、 s^* 上でスケールリング w を使用して、隣接パッチの振る舞いを考慮する：もし異常候補 $m^{\text{test},*}, m^*$ に最も近いメモリバンク特徴が、それ自体近隣のサンプルから遠く、それによって既に稀な公称出現である場合、異常スコアを増加させます。

$$s = \left(1 - \frac{\exp \|m^{\text{test},*} - m^*\|_2}{\sum_{m \in \mathcal{N}_b(m^*)} \exp \|m^{\text{test},*} - m\|_2} \right) \cdot s^*, \quad (7)$$

v21}はテストパッチ特徴 m^* に対して M で b 最も近いパッチ特徴である。この再重み付けは単なる最大パッチ距離よりも頑健であることがわかりました。v25}が与えられると、セグメンテーションが直接続く。式7 (1行目) の画像レベルの異常スコアは $\arg \max$ -操作によって各パッチの異常スコアを計算する必要がある。セグメンテーションマップは、[14]と同様に、計算されたパッチ異常スコアを再整理することにより、同じステップで計算することができる。

を、それぞれの空間的位置で計算する。元の入力解像度と一致させるために、(中間的なネットワーク特徴を使用したい場合がある) 二次線形補間によって結果をアップスケールする。さらに、カーネル幅 $\sigma=4$ のガウスで結果を平滑化した。4で平滑化した、このパラメータは最適化しなかった。

4.実験

4.1.実験の詳細

データセット産業用異常検知のパフォーマンスを研究するために、我々の実験の大部分はMVTec Anomaly Detection ベンチマーク [5] で実行されている。

MVTec ADには、合計5354画像からなる15のサブデータセットがあり、そのうち1725画像がテストセットである。各サブデータセットは、公称値のみの学習データと、様々な欠陥タイプを持つ特定の製品の公称値と異常値の両方のサンプルとそれぞれの異常値グラウンドトゥルースマスクを含むテストセットに分割される。10, 14, 56]と同様に、画像は256×256と224×224にリサイズされ、中央で切り取られる。256と224 [v28] 224にリサイズされる。クラス保持オーグメンテーションに関する事前知識が必要なため、データのオーグメンテーションは適用されない。

また、より専門的なタスクにおける産業用異常検知についても研究する。そのために、[42]で使用されたMagnetic Tile Defects (MTD) [26]データセットを活用する。facts (MTD) [26]データセットは[42]で使用されたもので、照明レベルや画像サイズが異なる925枚の欠陥のない画像と392枚の異常な磁気タイル画像を含む。42]と同様に、欠陥のない画像の20%がテスト時に評価され、残りはコールドスタート学習に使用される。

最後に、Patch-Coreの非産業画像データへの潜在的な適用可能性も強調する。v32}の非産業用画像データへの適用可能性を強調し、例えば[52]や[14]のようにMini Shanghai Tech Campus (mSTC)でコールドスタートの異常位置特定をベンチマークします。(mSTC)は[52]や[14]などで行われている。mSTCはオリジナルのSTCデータセット[32]のサブサンプリングバージョンで、1つおきのトレーニングビデオフレームとテストビデオフレームのみを使用する。このデータセットには12の異なるシーンからの歩行者映像が含まれる。トレーニングビデオには通常の歩行者の行動が含まれ、テストビデオには格闘やサイクリングなどの異なる行動が含まれる。コールドスタート実験の信頼性を高めるため、我々は確立されたmSTCプロトコル[14, 52]に従い、異常監視を利用せず、画像は256×256にリサイズした。

評価指標。画像レベルの異常検知性能は、生成された異常スコアを用いて、レシーバ・オペレータ曲線下面積 (AUROC) により測定される。先行研究に従い、MVTec 上でクラス平均 AUROC [2, 10, 14]を計算する。セグメンテーション性能を測定するために、[6]に従い、ピクセル単位の AUROC と PRO メトリックの両方を使用する。PRO スコアは、MVTec AD の様々な異常サイズをより良く考慮するために、接続された異常成分の重なりと回復を考慮する。

表1.MVTec AD [5]における異常検知性能(AUROC)。PaDiM*は[14]の問題特異的バックボーン選択の結果を示す。誤分類の総カウントは、F1-最適閾値が与えられた場合の偽陽性予測と偽陰性予測の合計として決定された。我々は競合する手法の個々の異常スコアを持っていなかったため、PatchCoreについてのみこの数を計算することができた。

Method	SPADE [10]	PatchSVDD [56]	DifferNet [42]	PaDiM [14]	Mah.AD [40]	PaDiM* [14]	PatchCore-25%	PatchCore-10%	PatchCore-1%
AUROC ↑	85.5	92.1	94.9	95.3	95.8	97.9	99.1	99.0	99.0
Error ↓	14.5	7.9	5.1	4.7	4.2	2.1	0.9	1.0	1.0
Misclassifications ↓	-	-	-	-	-	-	42	47	49

表2.MVTec AD [5]における異常セグメンテーション性能 (ピクセル単位のAUROC)。

Method	AE _{SSIM} [5]	γ -VAE + grad. [15]	CAVGA-R _w [52]	PatchSVDD [56]	SPADE [10]	PaDiM [14]	PatchCore-25%	PatchCore-10%	PatchCore-1%
AUROC ↑	87	88.8	89	95.7	96.0	97.5	98.1	98.1	98.0
Error ↓	13	11.2	11	4.3	4.0	2.5	1.9	1.9	2.0

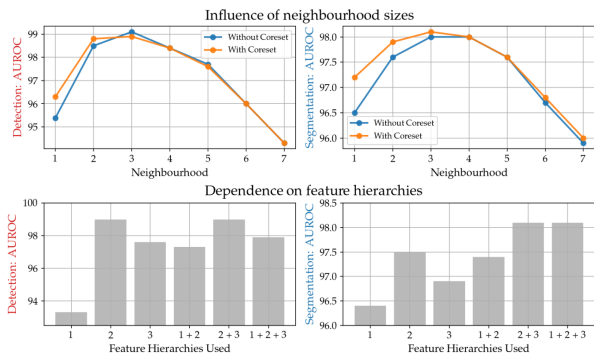


図4.ローカルウェアネスとネットワーク特徴の深さ対検出性能。PROスコアの結果は補足を参照。

4.2.MVTec ADの異常検知

MVTecにおける画像レベルの異常検知の結果を表1に示す。v4}については、メモリバンクサブサンプリングの様々なレベル (25%、10%、1%) について報告する。全ての場合において、PatchCoreは有意に高い平均画像異常検出性能を達成し、全てのサブデータセットにおいて一貫して高い性能を達成した (詳細な比較は補足Bを参照)。2.1% (PaDiM) の誤差が Patch-Core-25%で0.9%に減少したことに注意してください。Core-25%では0.9%になり、誤差が57%減少したことになります。これは、工業的な検査環境において、重要かつ有意な削減となります。最適F1閾値でのMVTecの場合、1725枚の画像のうち誤って分類されたのは42枚のみで、全クラスの3分の1が完璧に解決されている。補足資料Bでは、F1-optimal working pointとfull recallの両方で、SPADEとPaDiMの両方と比較して分類エラーも低いことも示す。v8}ではCoreでは、誤分類されたままの画像は50枚以下である。さらに、PatchCoreはピクセル単位のAUROC (表2、98.1対PaDiMの97.5)とPROメトリック (表3、93.5対92.1)の両方で測定された、最先端の異常セグメンテーションを達成する。図1のサンプルセグメンテーションは、正確な異常位置の特定に関する定性的な印象を与える。

さらに、コアセットのメモリサブサンプリングの効果により、PatchCore-1%をより高い解像度の画像 (例えば、224の代わりに280/320) に適用することができ、また、PatchCore-1%をより高い解像度の画像 (例えば、224の代わりに280/320) に適用することができる。

PatchCore-10%よりも少ない推論時間を保持しながら、デフォルトの解像度で、より多くの異常検出を行うことができる。これにより、Tab.4 (詳細な結果は補足を参照) で強調されているように、部分的には再び誤差を半分以上にすることができる (例えば画像レベルのAUROCでは1% → 0.4%)。

4.3.推論時間

もう一つの次元は推論時間である。WideResNet50を用いたSPADE[10]とPaDiM[14]の再実装と、可能な限りGPUでの演算を比較した結果を表5に示す (実装の詳細は付録A)。これらの推論時間は、バックボーンを通るフォワードパスを含む。見てわかるように、PatchCore-100% (サブサンプリングなし) の画像とピクセルレベルの共同異常検出の推論時間は、SPADE [10]よりも低いが、性能は高い。コアセットサブサンプリングにより、Patchcoreはさらに高速化され、PaDiMよりも推論時間が短くなる一方で、最先端の画像レベルの異常検出とセグメンテーション性能を保持することができる。最後に、推論時間を短縮する直交的な方法として、近似最近傍探索 (IVFPQ [27]) を用いたPatchCore-100%を検証する (これはSPADEにも適用できるが、既にPatchCore-1%よりも著しく性能が悪い)。我々は、特に画像レベルの異常検出において性能の低下を発見したが、推論時間はPatch-Core-1%よりも依然として高い。Core-1% .v26}100%での近似最近傍探索は、性能の低下にもかかわらず、他の手法よりも優れている。コアセットと近似最近傍探索を組み合わせることで、さらに推論時間が短縮され、より大きなデータセットへの拡張が可能になる。

4.4.アブレーション研究

局所的に認識されるパッチ特徴とコアセット削減法のアブレーションについて報告する。補足実験として、異なるバックボーン間での一貫性 (§C.2)、画像解像度の向上によるスケーラビリティ (§C.3)、残存エラーの定性的解析 (§C.4) を示す。

表3.MVTec AD [5]の異常検知性能 (PRO [%]で測定) [5, 10].

Method	AE _{SSIM} [5]	Student [6]	SPADE [10]	PaDiM [14]	PatchCore-25%	PatchCore-10%	PatchCore-1%
PRO ↑	69.4	85.7	91.7	92.1	93.4	93.5	93.1
Error ↓	30.6	14.3	8.3	7.9	6.6	6.5	6.9

表4.PatchCore-1%の高解像度/大きなバックボーン/アンサンブル。コアセットのサブサンプリングは、高速な推論を維持しながら、計算コストの高いセットアップを可能にする。

Metric→	AUROC	pwAUROC	PRO
DenseN-201 & RNext-101 & WRN-101 (2+3), Imagesize 320			
Score ↑	99.6	98.2	94.9
Error ↓	0.4	1.8	5.6
WRN-101 (2+3), Imagesize 280			
Score ↑	99.4	98.2	94.4
Error ↓	0.6	1.8	5.6
WRN-101 (1+2+3), Imagesize 280			
Score ↑	99.2	98.4	95.0
Error ↓	0.8	1.6	5.0

表5.MVTec ADの画像ごとの平均推論時間。スコアは (画像AUROC、ピクセルAUROC、PROメトリック)。

Method	PatchCore-100%	PatchCore-10%	PatchCore-1%
Scores	(99.1, 98.0, 93.3)	(99.0, 98.1, 93.5)	(99.0, 98.0, 93.1)
Time (s)	0.6	0.22	0.17
Method	PatchCore-100% + IVFPQ	SPADE	PaDiM
Scores	(98.0, 97.9, 93.0)	(85.3, 96.6, 91.5)	(95.4, 97.3, 91.8)
Time (s)	0.2	0.66	0.19

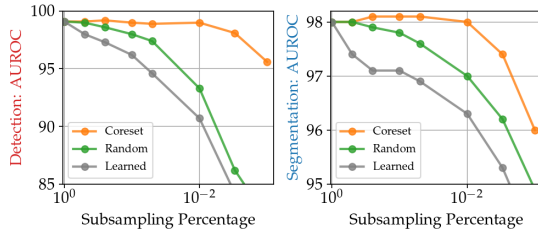


図5.異なるサブサンプラーのパフォーマンス保持率、PROスコアの結果は補足を参照。

4.4.1 パッチフィーチャーと階層を局所的に認識する

式1の異なる近傍サイズにおける異常検出性能の変化を評価することにより、局所的に認識されるパッチ特徴 (§3.3) の重要性を調査する。図4の上半分の結果は、パッチベースの異常予測における局所性と大域的コンテキストの間の明確な最適を示しており、近傍サイズ $p =$ の動機付けとなる。3よりグローバルなコンテキストはネットワーク階層を下げることによって達成できるが (例えば [10, 14] を参照)、解像度の低下と ImageNet のクラスバイアスが重くなる (§3.1)。最初の3つの WideResNet50 ブロックに1~3のインデックスを付けると、図4 (bottom) は再び、高度にローカライズされた予測、よりグローバルなコンテキスト、ImageNet バイアスの間の最適を強調する。見てわかるように、階層レベル2からの特徴量

はすでに最先端の性能を達成しているが、それ以降の高階層レベル (デフォルト設定として選択されている2+3) から取られた追加特徴マップによって恩恵を受ける。

4.4.2 コアセットのサブサンプリングの重要性

図5は、異なるメモリバンク M のサブサンプリング方法を比較したものである: 貪欲なコアセット選択、ランダムサブサンプリング、サブサンプリング目標パーセンテージ p_{target} に対応する基底プロキシのセットの学習。後者では、プロキシ $p_i \in \mathcal{P} \subset \mathbb{R}^d$ を $|\mathcal{P}| = p_{\text{target}} \cdot |M|$ でサンプリングし、そのプロキシは基底再構築目的 p_{target} を最小化するように課される。

$$\mathcal{L}_{\text{rec}}(m_i) = \left\| m_i - \sum_{p_k \in \mathcal{P}} \frac{e^{\|m_i - p_k\|_2}}{\sum_{p_j \in \mathcal{P}} e^{\|m_i - p_j\|_2}} p_k \right\|_2^2, \quad (8)$$

メモリバンクデータ M を最もよく記述する N プロキシを見つける。図5では、3つの設定を比較し、コアセットに基づくサブサンプリングが他の可能な選択肢よりも性能が良いことを示す。サブサンプリングなしの性能は、サイズが2桁小さいコアセット削減メモリバンクに匹敵する。また、サブサンプリングされたメモリバンクは、再ダンダンシーが非常に少ない。サブサンプリングされていないメモリバンクとコアセット・サブサンプリングされたメモリバンクについて、テスト時に使用されるメモリバンク・サンプルの割合を記録しました。当初はメモリバンク・サンプルの30%未満しか使用されていませんでしたが、コアセット・サブサンプリング (1%) により、この割合は95%近くまで増加しました。特定のサブサンプリング間隔 (約50%から10%の間) では、サブサンプリングしない PatchCore と比較して、異常検出とローカライゼーションの共同パフォーマンスが部分的に向上することさえ分かります。最後に、ストライドの増加 (式3参照) によりメモリバンクサイズ M を減少させると、解像度の低下によりパフォーマンスが低下し、ストライド $s = 2$ は画像の異常検出を行います。2は97.6%の画像異常検出AUROCを与え、ストライド $s = 3$ は96.6%のAUROCを与える。3は96.8%のAUROCであった。

4.5. ローショット異常検知

限られた公称データを利用することは、実世界の検査に関連するセットである。従って、MVTec AD のフルデータでの結果の再ポータリングに加えて、より少ない訓練例でのパフォーマンスも研究する。学習サンプルの量を1 (全名目学習データの0.4%に相当) から50 (21%) まで変化させ、同じバックボーン (WideResNet50) を使用した SPADE [10] と PaDiM [14] の再実装と比較する。結果は図6にまとめられており、詳細な結果は Supp.

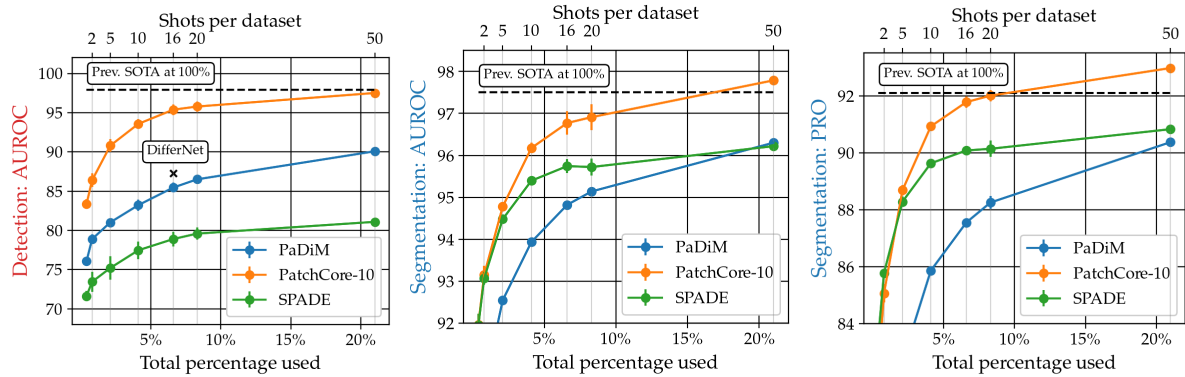


図6. PatchCoreは、競合他社よりも顕著に高いサンプル効率を示しており、わずかな公称学習データで、以前の最先端技術と一致する。PaDiMとSPADEは比較のためWideResNet50で再実装されている。

表 6.42]で報告された結果と比較したmSTC[32, 52]での異常セグメンテーションとMTD[26]での異常検知。

mSTC	CAVGA-R _n [52]	SPADE [10]	PaDiM [14]	PatchCore-10
Pixelwise AUROC [%]	85	89.9	91.2	91.8
MTD	GANomaly [2]	1-NN [35]	DifferNet [42]	PatchCore-10
AUROC [%]	76.6	80.0	97.7	97.9

§C.1.示されるように、公称学習データの1/5だけを使用しても、PatchCoreは以前の最先端性能に匹敵することができる。さらに、[42]で実行された16ショットの実験と比較すると、PatchCoreは既に事前学習された特徴量の上に正規化フローモデルを適応させる彼らのアプローチを上回ることが分かる。10]の画像レベルメモリアプローチと比較すると、わずか5/1の公称ショットで、一致するローカライゼーションと検出性能が得られる。

4.6.他のベンチマークでの評価

v4}のベンチマークを、さらに2つの異常検出性能ベンチマークで行う：ShanghaiTech Campus dataset (STC) [32]とMagnetic Tile Defects dataset (MTD) [26]である。4.1節で述べるSTCの評価は[52]、[14]、[10]に従う。我々は、画像を256xにリサイズしたSTCビデオデータのサブサンプリング版 (mSTC) に対する教師なし異常位置特定性能を報告する[14]。256 [14]検出コンテキストはImageNetで利用可能な自然画像データにかなり近い。階層レベル3と4でより深いネットワーク特徴マップを利用するが、それ以外はPatchCoreのハイパーパラメータチューニングは行わない。表6の結果 (top) は、PatchCoreのこのようなドメインへの良好な移植性を示唆する、最先端の異常局在化性能を示す。最後に、PaDiMのような空間的に硬直したアプローチが直接適用できない、様々なサイズの磁気タイル欠陥画像を含むMTDを検証する。ここで、公称データは既に異常サンプル[42]に見られるような高い変動性を示している。我々は[42]で提案されたプロトコルに従い、画像レベルの異常検出性能を測定し、その性能は[42]と同等であることを確認した。

42]のもの (表6, bottom) をわずかに上回る。

5.結論

この論文ではコールドスタート異常検出のためのPatchCoreアルゴリズムを紹介したが、このアルゴリズムではテスト時に異常データを検出しセグメント化するために、名目的な例のみの知識を活用しなければならない。PatchCoreは、ImageNetの事前学習されたネットワークから抽出された、局所的に認識された名目的なパッチレベルの特徴表現からなるメモリバンクの使用により、テスト時に名目的なコンテキストを最大限に保持すること、コアセットのサブサンプリングにより実行時間を最小限に抑えることの間でバランスをとる。その結果、産業用異常検知ベンチマークにおいて、低い計算コストで最先端のコールドスタート画像異常検知・ローカライゼーションシステムを実現した。MVTECにおいて、我々は9%以上の画像異常検出AUROCを達成し、関連する小さなトレーニングセット領域において最高のサンプル効率を実現した。

より広い影響。自動化された産業用異常検知は、コンピュータ・ビジョンの最も成功したアプリケーションの1つであるため、Patch-Coreによって得られた改善は、この分野の実務者にとって注目すべきものとなります。Coreは、この分野の実務者にとって注目すべきものである。我々の研究は工業的な異常検知に特化しているため、社会的な悪影響は限定的である。また、基本的なアプローチは、より論争的な領域における検出システムに活用できる可能性があるが、我々の改善は、そのようなシステムの社会的応用を変えるほど重大なものではないと考える。

限界。v14}は、手元の問題ドメインに特別に適應する必要なく、産業用異常検知に高い有効性を示すが、一般的に、適用可能性は、利用される事前学習された特徴の移植性によって制限される。これはPatchCoreの有効性と利用される特徴の適應を融合することで対処できる。この興味深い拡張は今後の研究に委ねる。

謝辞

Yasser JadidiとAlex Smolaには、計算インフラのセットアップを支援していただいた。K.R.はInternational Max Planck Research School for Intelligent Systems (IMPRS-IS)とEuropean Laboratory for Learning and Intelligent Systems (ELLIS)のPhDプログラムのサポートに感謝する。

参考文献

[1] Pankaj Agarwal, Sariel Har, Peled Kasturi, and R Varadarajan. コアセットによる幾何学的近似. *Combinatorial and Computational Geometry*, 52, 11 2004.2, 4 [2] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: 敵対的学習による半教師付き異常検知. In *Asian Conference on Computer Vision*, pages 622-637. Springer, 2018.1, 2, 5, 8, 4 [3] Jerone Andrews, Thomas Tanay, Edward Morton, and Lewis Griffin. 異常検知のための転送表現学習. 07 2016.4 [4] Liron Bergman, Niv Cohen, and Yedid Hoshen. Deep nearest neighbor anomaly detection. {このような場合、"Deep nearest neighbor anomaly detection"が有効である。1, 2, 3 [5] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad - 教師なし異常検知のための包括的な実世界データセット. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.1, 2, 5, 6, 7, 4 [6] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students : 生徒と教師の異常検出. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.2, 5, 7, 4 [7] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2019.2 [8] Wieland Brendel and Matthias Bethge. CNNをbag-of-local-featuresモデルで近似すると、imagenetで驚くほどうまくいく. In *International Conference on Learning Representations*, 2019.2 [9] Kenneth L. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Trans. Algorithms*, 6(4), Sept. 2010.2 [10] Niv Cohen and Yedid Hoshen. ディープピラミッド対応によるサブ画像異常検出. {このような場合、"abs/2005.02357, 2020"を参照する必要がある。1, 2, 3, 4, 5, 6, 7, 8 [11] Sanjoy Dasgupta and Anupam Gupta. (1)偶数(2)偶数. *Random Structures & Algorithms*, 22(1):60-65, 2003.4 [12] Diana Davletshina, Valentyn Melnychuk, Viet Tran, Hitansh Singla, Max Berrendorf, Evgeniy Faerman, Michael Fromm, and Matthias Schubert. X線画像の教師なし異常検出, 2020.1

[13] Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. 生成的敵対ネットワークを用いた画像異常検出. Michele Berlingerio, Francesco Bonchi, Thomas Gärtner, Neil Hurley, and Georgiana Ifrim, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 3-17, Cham, 2019. Springer International Publishing.2 [14] Thomas Defard, Aleksandr Setkov, Angélique Loesch, and Romaric Audigier. Padim: A patch distribution modeling framework for a anomaly detection and localization. Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, Hugo Jair Escalante, and Roberto Vezzani, editors, *Pattern Recognition. ICPR International Workshops and Challenges*, pages 475-489, Cham, 2021. シュプリンガー・インターナショナル・パブリッシング. 2, 3, 5, 6, 7, 8, 1, 4 [15] David Dehaene, Oriel F rigo, Sébastien Combexelle, and Pierre Eline. (注)本論文は、「データ多様体上のエネルギーに基づく反復投影による異常局在化」を目的としたものである. in *International Conference on Learning Representations*, 2020.6, 4 [16] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248-255, 2009.2 [17] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 実NVPを用いた密度推定. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.2 [18] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. *A Geometric Framework for Un-supervised Anomaly Detection*, pages 77-101. Springer US, Boston, MA, 2002.2 [19] Dan Feldman, Matthew Faulkner, and Andreas Krause. (訳注:この論文では、このような「曖昧さ」を「曖昧さ」と呼ぶことにする)。そのため、このような学習は、その学習がどのようなものであったのか、また、その学習がどのようなものであったのかを知ることができる. *Systems*, 第24巻, 2142-2150ページ. Curran Associates, Inc.2 [20] Izhak Golan and Ran El-Yaniv. 幾何学的変換を用いた深い異常検出. S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 9758-9769. Curran Associates, Inc.2, 4 [21] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.2 [22] Sariel Har-Peled and Akash Kushal. k-中央値とk-meansクラスタリングのためのより小さなコアセット. *Discrete and Computational Geometry*, 37:3-19, 12 2007.2 [23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 画像認識のための深い残差学習. In *Proceed-*

ings of the *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.3, 6 [24] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean.ニューラルネットワークの知識を抽出する。In *NIPS Deep Learning and Representation Learning Workshop*, 2015.2 [25] Chaoqing Huang, Jinkun Cao, Fei Ye, Maosen Li, Ya Zhang, and Cewu Lu.(訳注:この論文では、"anomaly detection"のための逆変換オートエンコーダについて記述している。CoRR, abs/1911.10676, 2019.2, 4 [26] Yibin Huang, C. Qiu, and K. Yuan.磁気タイルの表面欠陥顕著性。The *Visual Computer*, 36:85-96, 2018.2, 5, 8 [27] Jeff Johnson, Matthijs Douze, and Hervé Jégou.gpusによる10億スケールの類似性探索。IEEE *Transactions on Big Data*, pages 1-1, 2019.6, 1 [28] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Viegas, and Michael Terry.Xrai: リージョンによるより良い帰属。In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.2 [29] Ki Hyun Kim, Sangwoo Shim, Yongsub Lim, Jongseob Jeon, Jeongwoo Choi, Byungchan Kim, and Andre S. Yoon.Rapp: 投影経路に沿った再構成による新規性検出。in *International Conference on Learning Representations*, 2020.2 [30] Durk P Kingma and Prafulla Dhariwal.Glow: 反転可能な1x1畳み込みによる生成フロー。S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. *Systems*, 第31巻。カラン・アソソエイツ社、2018年2 [31] Wenqian Liu, Runze Li, Meng Zheng, Srikrishna Karamam, Ziyang Wu, Bir Bhanu, Richard J. Radke, and Octavia Camps.このような場合、「曖昧さ」が「曖昧さ」の原因になっている可能性があることを、「曖昧さ」が「曖昧さ」の原因になっている可能性があることを示す。in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.4 [32] W. Liu, D. Lian W. Luo, and S. Gao.v17}において。*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.5, 8 [33] Prasanta Chandra Mahalanobis.統計学における一般化距離について。*Proceedings of the National Institute of Sciences(Calcutta)*, 2:49-55, 1936.2 [34] Ben Mussay, Margarita Osadchy, Vladimir Braverman, Samson Zhou, and Dan Feldman.データ非依存のニューラル・プルーニング。in *International Conference on Learning Representations*, 2020.2 [35] Tiago S. Nazaré, Rodrigo Fernandes de Mello, and Moacir A. Ponti.監視カメラ映像の異常検出において、事前に訓練されたCNNは良い特徴抽出器か? *CoRR*, abs/1811.08495, 2018.8 [36] Duc Tam Nguyen, Zhongyu Lou, Michael Klar, and Thomas Brox. Anomaly detection with multiple-hypotheses predictions. Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*において。Machine Learning, *Proceedings of Machine*の97巻。Learning Research, pages 480-480.PMLR, 09-15 Jun 2019.1 [37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala.ピトーチ: An imperative style, high-performance deep learning library, 2019.1 [38] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang.Oc-gan: Oc-gan: 制約付き潜在表現を持つganを用いた1クラスの新規性検出。in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. *Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.2 [39] Stanislav Pidhorskyi, Ranya Almoheisen, Donald A. Adjeroh, and Gianfranco Doretto.Generative probabilistic novelty detection with adversarial autoencoders.v34}にて。*32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 6823-6834, Red Hook, NY, USA, 2018.Curran Associates Inc.1, 2 [40] Oliver Rippel, Patrick Mertens, and Dorit Merhof.v37} *ference on Pattern Recognition (ICPR)*, pages 6726-6733, 2021.2, 6, 4 [41] Karsten Roth, Timo Milbich, Samartha Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen.(訳注:この論文では、「汎化」を「汎化」と呼んでいる。(訳注:この論文では、「深層計量学習における学習戦略と汎化性能の再検討」) Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119.v42}, volume 119 of *Proceedings of Machine Learning Research*, pages 824-825.PMLR, 13-18 Jun 2020.3 [42] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn.Same same but different: を使用した半教師付き欠陥検出。v45} *Computer Vision (WACV)*, Jan 20. *Computer Vision (WACV)*, Jan. 2021.1, 2, 5, 6, 8, 4 [43] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli.in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.1, 2 [44] 櫻田真由, 矢入武久.非線形次元削減を用いたオートエンコーダによる異常検知.in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning*.櫻田真由, 矢入武久, 非線形次元削減を用いたオートエンコーダによる異常検知。In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning*, MLSDA'14, page 4-11, New York, NY, USA, 2014.Association for Computing Machinery.1, 2 [45] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad Hossein Rohban, and Hamid R. Rabiee.異常検知のための多重解像度知識蒸留、2020年。2 [46] Bernhard Schölkopf, Robert C. Williamson, Alex J. Smola, John Shawe-Taylor, and John C. Platt.新規性検出のためのサポートベクトル法。In *Advances in Neural Information Processing Systems 12*, pages 582-588, Cambridge, MA, USA, June 2000.Max-Planck-Gesellschaft, MIT Press.2 [47] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra.Grad-cam: 勾配ベースの定位によるディープネットワークからの視覚的説明。v55}にて。

International Conference on Computer Vision (ICCV) ページ 618-626, 2017.2 [48] Ozan Sener and Silvio Savarese. 畳み込みニューラルネットワークの能動学習: コアセットのアプローチ. In *International Conference on Learning Representations*, 2018.2, 4 [49] Samarth Sinha, Han Zhang, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle, and Augustus Odena. Small-GAN: コアセットを用いたGAN学習の高速化. において、ハル・ダウム (Hal Daumé III)、アーティ・シン (Aarti Singh)、編集者、*Proceedings of the 37th International Conference on Machine Learning*, 第119巻. v5}, volume 119 of *Proceedings of Machine Learning Research*, pages 9005-9015. PMLR, 13-18 Jul 2020.3, 4 [50] David M. J. Tax and Robert P. W. Duin. サポートベクトルデータの記述. *Machine Learning*, 54:45-66, 2004.2 [51] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.1 [52] Shashanka Venkataramanan, Kuan-Chuan Peng, Rajat Vikram Singh, and Abhijit Mahalanobis. 画像における注意誘導型異常局在化. このような画像における注意誘導型異常ローカライゼーションは、画像における注意誘導型異常ローカライゼーションと呼ばれる。シュプリンガー・インターナショナル・パブリッシング。2, 5, 6, 8, 4 [53] Ross Wightman. Pytorch image models. <https://github.com/rwightman/pytorch-image-models>, 2019.1 [54] Laurence A. Wolsey and George L. Nemhauser. *Integer and Combinatorial Optimization*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2014.4 [55] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.6 [56] Jihun Yi and Sungroh Yoon. Patch svdd: 異常検出とセグメンテーションのためのパッチレベル svdd. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.1, 2, 5, 6, 4 [57] Sergey Zagoruyko and Nikos Komodakis. ワイド残差ネットワーク. Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC) Conference (BMVC)*, pages 87.1-87.12. BMVA Press, September 2016.3, 1, 6 [58] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. このような場合、"enjoy" することで、"enjoy" することができる。 *on Machine Learning, Proceedings of Machine Learning Research*, pages 1100-1109, New York, New York, USA, 20-22 Jun 2016. PMLR.4 [59] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600-612, 2004.2 [60] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. (訳注: この論文では、このような「異常」を検出するために、「自己符号化ガウス混合モデル」を使用する)。 In *International Conference on Learning Representations*, 2018.2

補足産業用異常検知における全回収を目指して

A. 実施内容

我々はPython 3.7[51]とPyTorch[37]でモデルを実装した。実験はNvidia Tesla V4 GPUで実行した。torchvisionとPyTorch Image Modelsリポジトリ[53]のtorchvision ImageNet-事前学習済みモデルを使用した。デフォルトでは、[10]と[14]に従い、PatchCoreは直接比較できるようにWideResNet50-backbone [57]を使用しています。パッチレベルの特徴は、ブロック2と3のfinal出力の特徴マップ集約から取得される。すべての最近傍検索と距離計算にはfaiss [27]を使用する。

B. フルMVTec ADの比較

このセクションでは、MVTec ADに関するより詳細な比較を行う。MVTec ADのサブデータセットが利用可能な場合には、より多くのモデルや、より詳細なパフォーマンス比較を含む。論文本編では、§4.2で言及している。対応する結果表はS1、S2、S3である。v1}25%は15個のMVTecデータセットのうち6個を解き、ほとんどのデータセットと平均で最高のAUROC性能を達成していることがわかる。

図S3は、WideResNet50バックボーンを使用したSPADE[10]とPaDiM[14]の再実装と、PatchCore変種のPrecision-Recall曲線とROC曲線を示す。また、比較可能な作業ポイントを与えるために、100%リコール時とF1最適閾値の下での分類誤差もプロットします。見てわかるように、{v3}は、SPADEやPaDiMとは対照的に、データセット全体でほぼ最適なPrecision-Recall曲線とROC曲線を持ち、定義された作業点でも一貫して低い分類誤差を達成する。

最後に、表S4は、より大きな画像サイズ(280×280)と、より大きな画像でも推論時に効率的な異常検知を可能にするPatchCore-1%を使用してさらに性能を向上させるためのWideResNet-101バックボーンに対する、すべてのMVTec ADサブデータセットでの詳細な性能を示している。

C. その他のアブレーションと詳細

C.1. ローショットの詳細実験

このセクションは、本研究の主要部分(§4.5)で提供されたローショットメソッドの研究に詳細な数値を提供する。その結果は表S5に含まれており、検出と異常局在化のメトリクスにおいて一貫して高い数値を示している。

C.2. 事前学習済みネットワークへの依存

v6}を異なるバックボーンでテストした結果をS6に示す。異なるバックボーンを選択しても、結果はほぼ安定していることがわかる。WideResNet50の選択はSPADEやPaDiMと比較できるようにした。

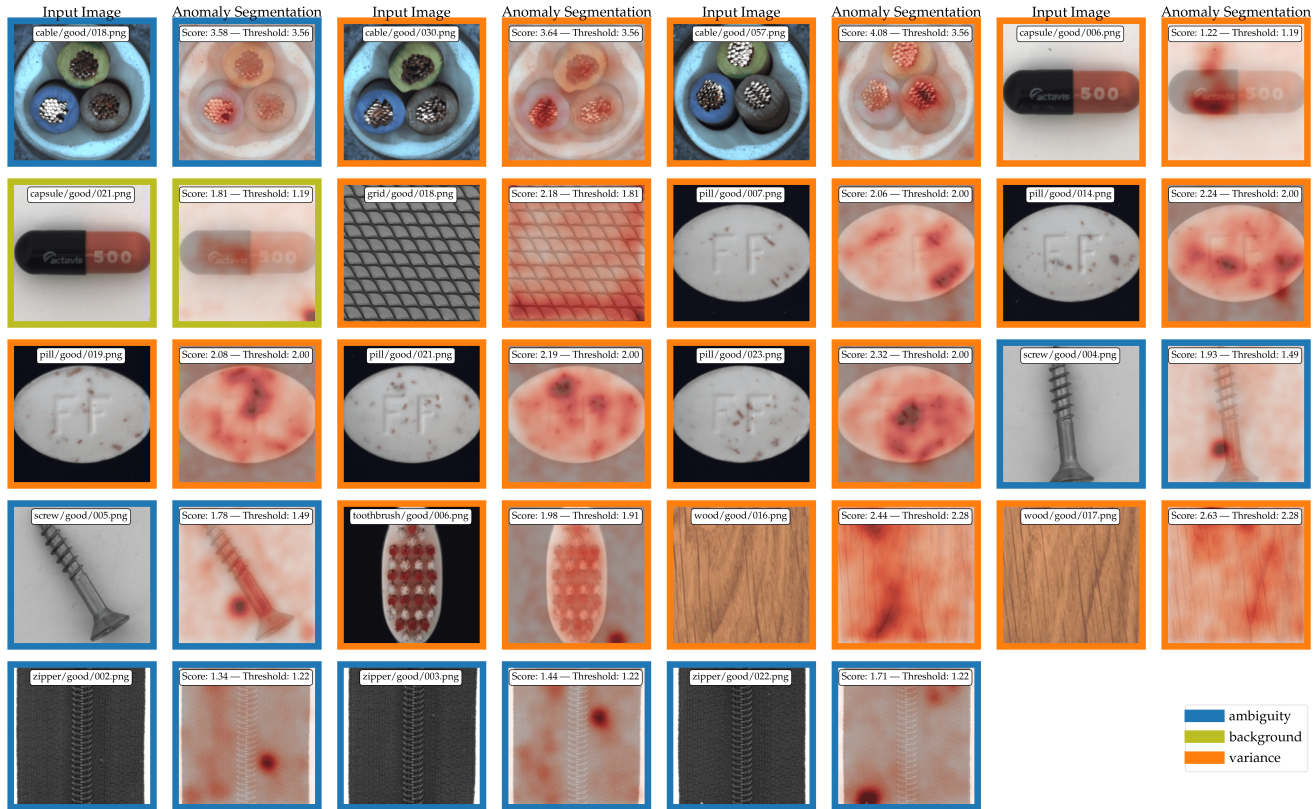
C.3. 画像解像度の影響

次に、画像サイズが性能に与える影響を調べる。本論文では、先行研究と比較するため、224×を使用し、224を使用し、先行研究と比較できるようにした。図S4では、画像サイズを288 {v8} 288, 360×360から448×448、近傍サイズ(P)を3、5、7、9の範囲で変化させた。検出性能はわずかに向上し、性能はPatchCoreで飽和する。異常のセグメンテーションについては一貫した増加が観察されるため、良好なローライゼーションが重要であれば、これは検証すべき要素である。

C.4. 残りの誤分類

高い画像レベルの異常検出性能により、残りの全ての誤分類を詳細に調べることができる。F1最適点を用いて作業点(スコアが異常とみなされる閾値)を計算する。この閾値では、合計19の偽陽性エラーと23の偽陰性エラーが残っており、そのすべてを図S1とS2に可視化した。各セグメンテーション・マップは閾値に対して正規化されているため、バックグラウンドのスコアが不均衡に顕著になるケースがある。

図S1を見ると、偽陽性エラーの大部分は、a)(青字)ラベリングのあいまいさ、すなわち異常とラベリングされる可能性のある画像変化、およびb)(オレンジ字)非常に高い公称分散から生じていることがわかる、



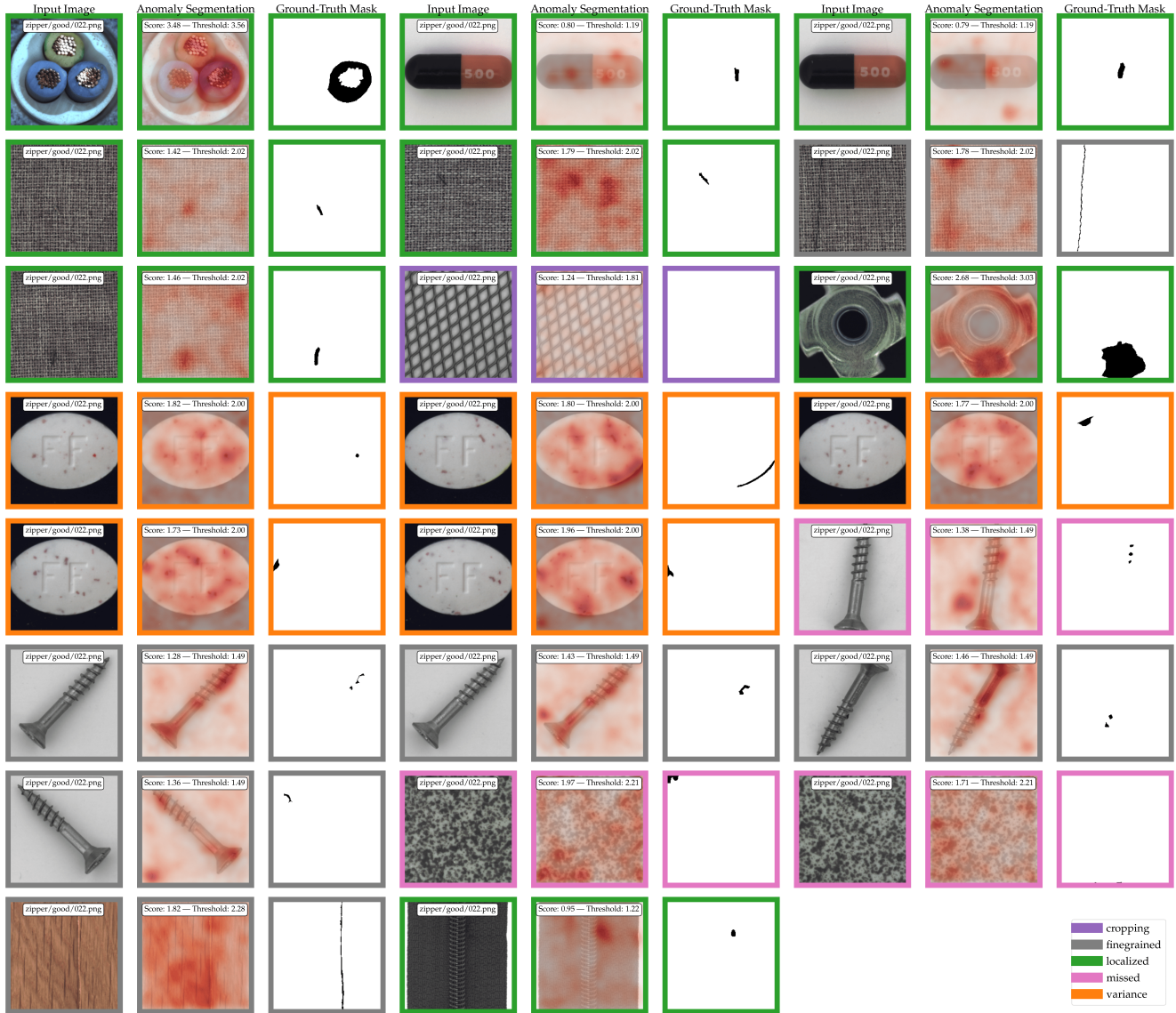
図S1.F1-最適閾値の下での) 残りの偽陽性分類の視覚化。色は異なるエラー源を示す。オレンジは、異常と間違われた高度の名目上の分散を示し、青は、ラベリング文脈の異常による誤分類を示し、オリーブ色は、異常な内容と間違われた背景の分散を示す。

潜在的なアノマリーに似ている。前者は提案された手法ではほとんど対処できないが、後者は公称データに何らかの形で適応することで対処できる。しかし、PatchCoreは適応的な方法を凌駕するので、このような適応はPatchCoreのような事前訓練に基づく方法と並行して運用するのが最も有望であろう。

偽陰性エラーを理解するために、生成されたセグメンテーションマップとグランドトゥルースマスクを図S2に示す。見てわかるように、異常の大部分はうまくローカライズされているが、異常領域に対する重み付けが不十分であり、何らかの後処理によって対処できる可能性がある。その他の誤分類の主な原因は、異常な文脈と誤認される公称分散が大きいことと、より高い画像解像度に移行したときに捕捉される可能性のある粒度の細かい異常である。完全に見逃された異常の量は、それに比べれば少なく、1つのケースでは、画像の前処理が実際の異常領域を切り取ることによって引き起こされた。

C.5.地域認識とサブサンプリング

完全を期すため、S5とS6にPROスコアの結果を含めて、本論文の図4と5を繰り返す。



図S2.F1-最適閾値の下での) 残りの偽陰性分類の可視化。色は異なるエラー源を示す。オレンジは異常と間違われた高度の公称分散を、緑は実際に局所的な異常を示すが、これらの異常の重みが少なすぎることを、ピンクは回復されなかった異常を、紫はトリミングに基づく画像処理のために見逃された異常 (合計1つの異常) を、灰色はより高い画像解像度で操作した場合に回復される可能性のある微細な異常を示す。

表 S1.MVTec AD [5]における異常検知性能 (AUROC)。PaDiM*は[14]の結果であり、画像レベルの異常検出のタスクのために特別に選択されたバックボーンを持つ。

↓ Method \ Dataset →	Avg	Bottle	Cable	Capsule	Carpet	Grid	Hazeln.	Leather	Metal Nut	Pill	Screw	Tile	Toothb.	Trans.	Wood	Zipper
GeoTrans [20]	67.2	74.4	78.3	67.0	43.7	61.9	35.9	84.1	81.3	63.0	50.0	41.7	97.2	86.9	61.1	82.0
GANomaly [2]	76.2	89.2	75.7	73.2	69.9	70.8	78.5	84.2	70.0	74.3	74.6	79.4	65.3	79.2	83.4	74.5
DSEBM [58]	70.9	81.8	68.5	59.4	41.3	71.7	76.2	41.6	67.9	80.6	99.9	69.0	78.1	74.1	95.2	58.4
OCSVM [3]	71.9	99.0	80.3	54.4	62.7	41.0	91.1	88.0	61.1	72.9	74.7	87.6	61.9	56.7	95.3	51.7
ITAE [25]	83.9	94.1	83.2	68.1	70.6	88.3	85.5	86.2	66.7	78.6	100	73.5	100	84.3	92.3	87.6
SPADE [10]	85.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CAVGA-R _w [52]	90	96	92	93	88	84	97	89	82	86	81	97	89	99	79	96
PatchSVDD [56]	92.1	98.6	90.3	76.7	92.9	94.6	92.0	90.9	94.0	86.1	81.3	97.8	100	91.5	96.5	97.9
DifferNet [42]	94.9	99.0	95.9	86.9	92.9	84.0	99.3	97.1	96.1	88.8	96.3	99.4	98.6	91.1	99.8	95.1
PaDiM [14]	95.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MahalanobisAD [40]	95.8	100	95.0	95.1	100	89.7	99.1	100	94.7	88.7	85.2	99.8	96.9	95.5	99.6	97.9
PaDiM* [14]	97.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PatchCore-25	99.1	100	99.5	98.1	98.7	98.2	100	100	100	96.6	98.1	98.7	100	100	99.2	99.4
PatchCore-10	99.0	100	99.4	97.8	98.7	97.9	100	100	100	96.0	97.0	98.9	99.7	100	99.0	99.5
PatchCore-1	99.0	100	99.3	98.0	98.0	98.6	100	100	99.7	97.0	96.4	99.4	100	99.9	99.2	99.2

表 S2.MVTec [5]における異常セグメンテーション性能 (ピクセル単位のAUROCで測定)。

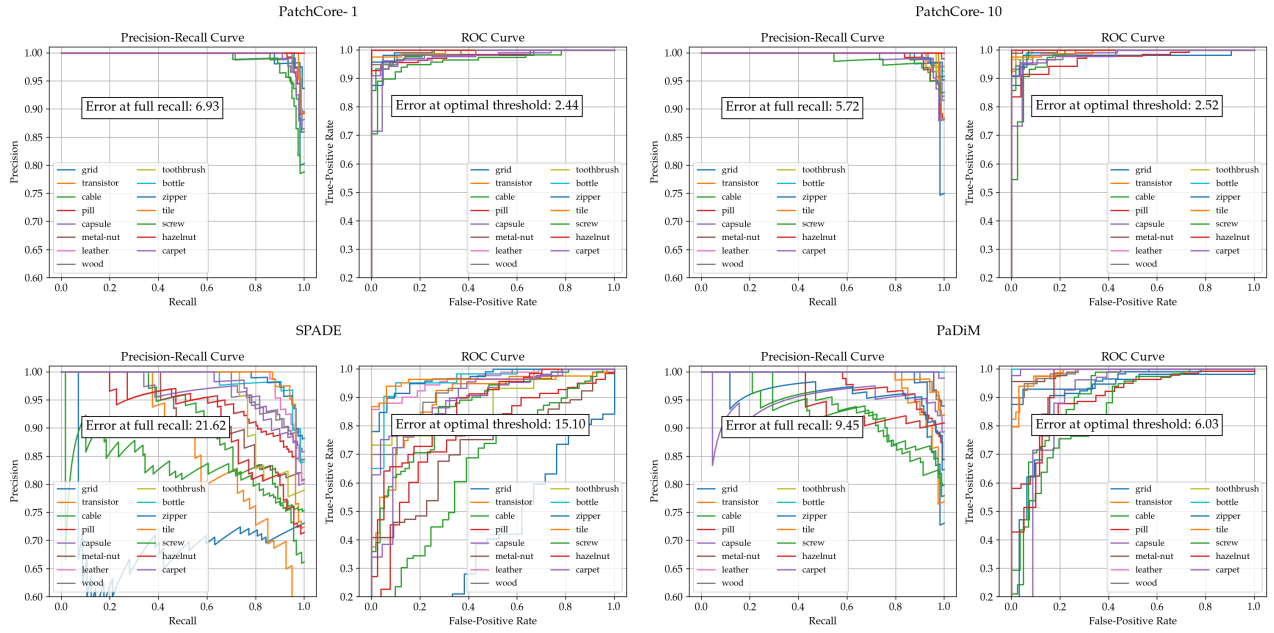
↓ Method \ Dataset →	Avg	Bottle	Cable	Capsule	Carpet	Grid	Hazeln.	Leather	Metal Nut	Pill	Screw	Tile	Toothb.	Trans.	Wood	Zipper
vis. expl. VAE [31]	86	87	90	74	78	73	98	95	94	83	97	80	94	93	77	78
AE _{SSIM} [5]	87	93	82	94	87	94	97	78	89	91	96	59	92	90	73	88
γ-VAE + grad. [15]	88.8	93.1	88.0	91.7	72.7	97.9	98.8	89.7	91.4	93.5	97.2	58.1	98.3	93.1	80.9	87.1
CAVGA-R _w [52]	89	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PatchSVDD [56]	95.7	98.1	96.8	95.8	92.6	96.2	97.5	97.4	98.0	95.1	95.7	91.4	98.1	97.0	90.8	95.1
SPADE [10]	96.0	98.4	97.2	99.0	97.5	93.7	99.1	97.6	98.1	96.5	98.9	87.4	97.9	94.1	88.5	96.5
PaDiM [14]	97.5	98.3	96.7	98.5	99.1	97.3	98.2	99.2	97.2	95.7	98.5	94.1	98.8	98.5	94.9	98.5
PatchCore-25	98.1	98.6	98.4	98.8	99.0	98.7	98.7	99.3	98.4	97.4	99.4	95.6	98.7	96.3	95.0	98.8
PatchCore-10	98.1	98.6	98.5	98.9	99.1	98.7	98.7	99.3	98.4	97.6	99.4	95.9	98.7	96.4	95.1	98.9
PatchCore-1	98.0	98.5	98.2	98.8	98.9	98.6	98.6	99.3	98.4	97.1	99.2	96.1	98.5	94.9	95.1	98.8

表 S3.MVTec [5]における異常セグメンテーション性能 (PRO [%]で測定) [5, 10].

↓ Method \ Dataset →	Avg	Bottle	Cable	Capsule	Carpet	Grid	Hazeln.	Leather	Metal Nut	Pill	Screw	Tile	Toothb.	Trans.	Wood	Zipper
AE _{SSIM} [5]	69.4	83.4	47.8	86.0	64.7	84.9	91.6	56.1	60.3	83.0	88.7	17.5	78.4	72.5	60.5	66.5
Student [6]	85.7	91.8	86.5	91.6	69.5	81.9	93.7	81.9	89.5	93.5	92.8	91.2	86.3	70.1	72.5	93.3
SPADE [10]	91.7	95.5	90.9	93.7	94.7	86.7	95.4	97.2	94.4	94.6	96.0	75.6	93.5	87.4	87.4	92.6
PaDiM [14]	92.1	94.8	88.8	93.5	96.2	94.6	92.6	97.8	85.6	92.7	94.4	86.0	93.1	84.5	91.1	95.9
PatchCore-25	93.4	96.2	92.5	95.5	96.6	96.0	93.8	98.9	91.4	93.2	97.9	87.3	91.5	83.7	89.4	97.1
PatchCore-10	93.5	96.1	92.6	95.5	96.6	95.9	93.9	98.9	91.3	94.1	97.9	87.4	91.4	83.5	89.6	97.1
PatchCore-1	93.1	95.9	91.6	95.5	96.5	96.1	93.8	98.9	91.2	92.9	97.1	88.3	90.2	81.2	89.5	97.0

表 S4.より大きな画像 (280 × 280) とWideResNet101バックボーンを使用したPatchCore-1によるMVTec AD [5]の異常検知とローカライズのパフォーマンス (AUROC)。

↓ Metric \ Dataset →	Avg	Bottle	Cable	Capsule	Carpet	Grid	Hazeln.	Leather	Metal Nut	Pill	Screw	Tile	Toothb.	Trans.	Wood	Zipper
PatchCore-1, Hierarchies (2, 3), Imagesize 280																
AUROC	99.4	100	99.6	98.2	98.4	99.8	100	100	100	97.2	98.9	98.9	100	100	99.5	99.9
pwAUROC	98.2	98.6	98.4	99.1	98.7	98.7	98.8	99.3	98.8	97.8	99.3	96.1	98.8	96.4	95.1	98.9
PRO	94.4	96.6	93.8	96.0	97.4	96.8	91.2	99.1	94.8	94.0	97.5	89.5	95.5	84.8	91.7	97.8
PatchCore-1, Hierarchies (1, 2, 3), Imagesize 280																
AUROC	99.2	100	99.7	98.1	98.2	98.3	100	100	100	97.1	99.0	98.9	98.9	99.7	99.9	99.7
pwAUROC	98.4	98.6	98.7	99.1	98.7	98.8	98.8	99.3	99.0	98.6	99.5	96.3	98.9	97.1	95.2	99.0
PRO	95.0	96.6	94.6	96.3	97.5	97.0	91.5	99.1	95.4	96.0	98.1	90.0	95.8	85.9	92.0	98.0



図S3.v0)の精度-再現率曲線(左)とROC曲線(右)。SPADE [10]とPaDiM [14]。線の色の違いはMVTecクラスの違いに対応する。

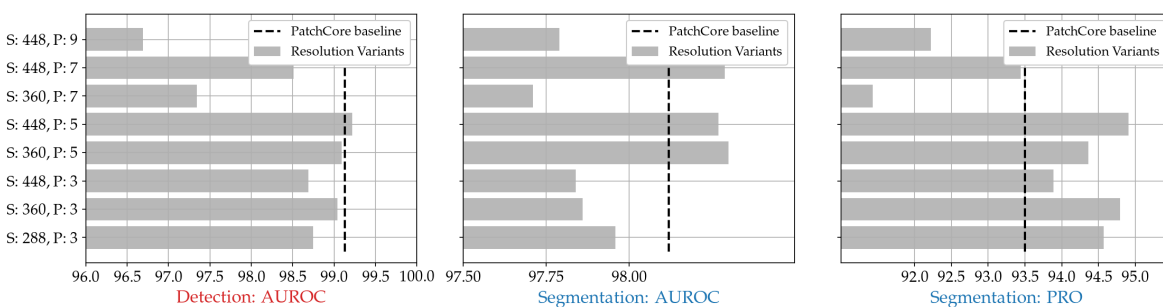
表S5.MVTec [5]における低ショット異常検知性能 (AUROC測定による)。

↓ Method \ Shots →	1	2	5	10	16	20	50
Retained %	0.4	0.8	2.1	4.1	6.6	8.3	21
IMAGE-LEVEL AUROC							
SPADE	71.6 ± 0.7	73.4 ± 1.3	75.2 ± 1.5	77.5 ± 1.1	78.9 ± 0.9	79.6 ± 0.8	81.1 ± 0.4
PaDiM	76.1 ± 0.4	78.9 ± 0.6	81.0 ± 0.2	83.2 ± 0.7	85.5 ± 0.6	86.5 ± 0.3	90.1 ± 0.3
DifferNet	-	-	-	-	87.3	-	-
PatchCore-10	83.4 ± 0.6	86.4 ± 0.9	90.8 ± 0.8	93.6 ± 0.6	95.4 ± 0.7	95.8 ± 0.6	97.5 ± 0.3
PatchCore-25	84.1 ± 0.7	87.2 ± 1.0	91.0 ± 0.9	93.8 ± 0.5	95.5 ± 0.6	95.9 ± 0.6	97.7 ± 0.4
PIXEL-LEVEL AUROC							
SPADE	91.9 ± 0.3	93.1 ± 0.2	94.5 ± 0.1	95.4 ± 0.1	95.7 ± 0.2	95.7 ± 0.2	96.2 ± 0.0
PaDiM	88.2 ± 0.3	90.5 ± 0.2	92.5 ± 0.1	93.9 ± 0.1	94.8 ± 0.1	95.1 ± 0.1	96.3 ± 0.0
PatchCore-10	92.0 ± 0.2	93.1 ± 0.2	94.8 ± 0.1	96.2 ± 0.1	96.8 ± 0.3	96.9 ± 0.3	97.8 ± 0.0
PatchCore-25	92.4 ± 0.3	93.3 ± 0.2	94.8 ± 0.1	96.1 ± 0.1	96.8 ± 0.3	96.9 ± 0.3	97.7 ± 0.0
PRO METRIC							
SPADE	83.5 ± 0.4	85.8 ± 0.1	88.3 ± 0.2	89.6 ± 0.1	90.1 ± 0.2	90.1 ± 0.3	90.8 ± 0.1
PaDiM	72.4 ± 1.2	77.8 ± 0.7	82.7 ± 0.2	85.9 ± 0.2	87.5 ± 0.2	88.2 ± 0.2	90.4 ± 0.1
PatchCore-10	82.4 ± 0.3	85.1 ± 0.3	88.7 ± 0.2	90.9 ± 0.1	91.8 ± 0.2	92.0 ± 0.2	93.0 ± 0.1
PatchCore-25	83.7 ± 0.5	86.0 ± 0.3	88.8 ± 0.2	90.9 ± 0.1	91.7 ± 0.1	91.9 ± 0.2	92.8 ± 0.0

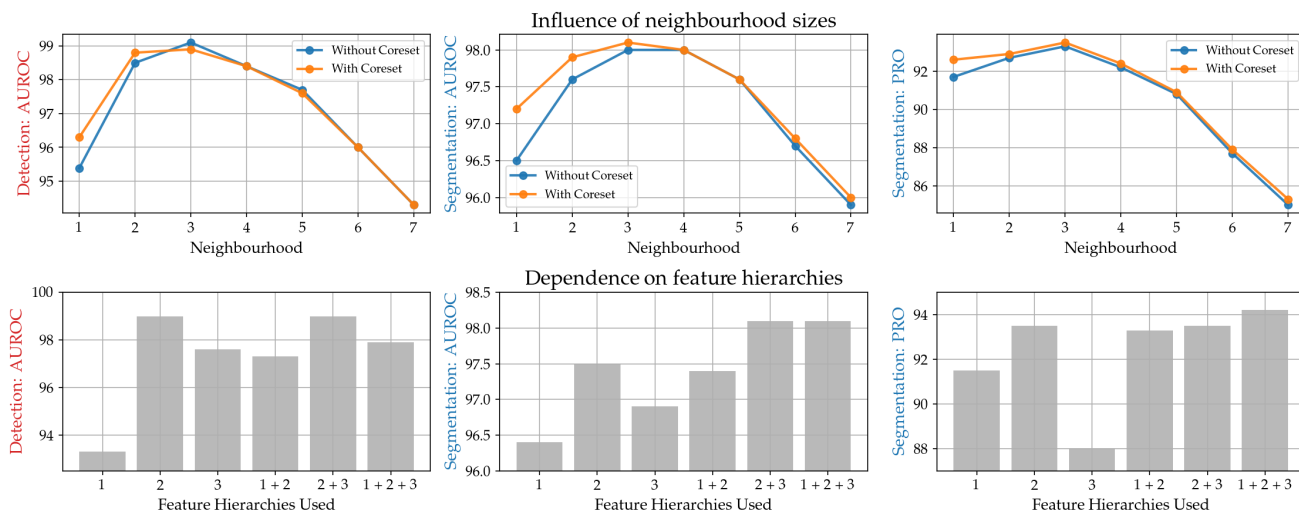
タブ S6.MVTec [5]での異常検知性能。

オーロック

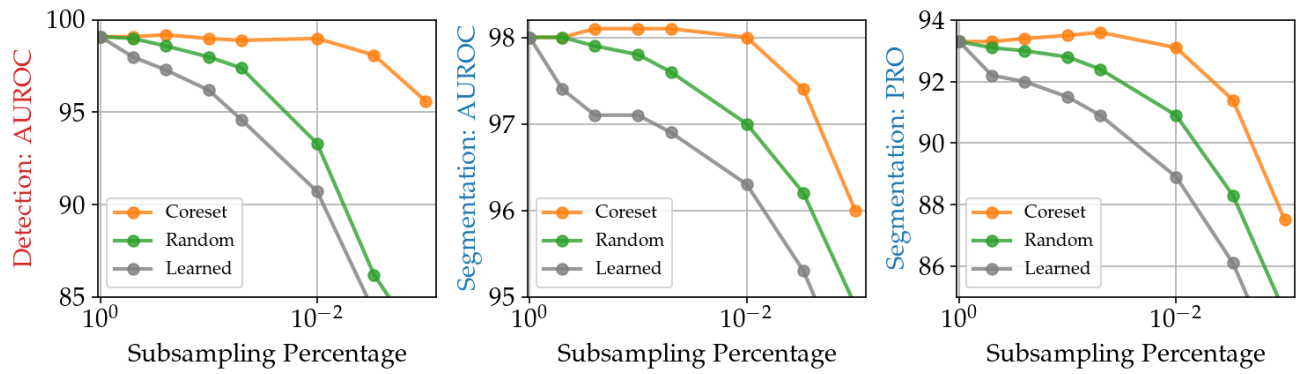
↓ Backbone	% of \mathcal{M}	Img. AUROC	Pw. AUROC	PRO
ResNet50 [23]	10	99.0	98.1	93.3
	1	98.7	97.8	93.3
WideResNet50 [57]	10	98.9	98.1	93.5
	1	99.0	98.0	93.1
ResNet101 [23]	10	98.6	97.9	92.5
	1	98.7	97.8	92.2
WideResNet101 [57]	10	99.1	98.2	93.4
	1	99.0	98.1	93.0
ResNeXt101 [55]	10	98.9	98.0	92.8
	1	98.7	97.8	92.6



図S4.画像サイズ(S)と近傍領域サイズ(P)のPatchCore性能への影響。v2}のベースライン (デフォルト値) を参考として示す。



図S5.ローカルウェアネスとネットワーク特徴の深さが異常検知性能に及ぼす影響。



図S6.異なるサブサンプラーのパフォーマンス保持。