H-020

相対位置推定を導入した異常画像検出手法
Anomaly Localization in Images Using Position-Classifier

向 雲† 伊藤 聡†
Yun Xiang Satoshi Ito

1. はじめに

異常画像検出技術は、画像から撮影対象の異常を検出する技術であり、幅広く応用できるため、近年、注目が集まっている。たとえば、医療分野において、人間組織の病理検査を異常画像検出で対応する手法が提案されている[1]。また、工場内で、部品購入時の受入検査、製品出荷時の検査向けのデータセットと手法なども公開されている[2]。なお、異常画像検出技術には、画像中に異常があるかどうかだけを判断するものと、画像中の異常箇所まで特定するものがある。実用上は、画像中の異常箇所に対して対処しなければならないことが多いため、異常箇所まで特定する技術が望まれる。

異常箇所を精確に特定するには、画素レベルで正常・異常ラベルが付与されたデータを用いて教師あり学習するのが理想的である。しかし、画素レベルで教示データを用意するには莫大なアノテーションコストがかかるため、現実的ではない。また、異常画像は入手が困難であることが多いため、異常画像を学習に利用できない場合もある。そこで、本稿では正常画像のみで学習し（半教師あり学習）、異常画像の検出及び異常箇所の特定を行う手法の検討を行っている。本稿の手法による、MVTec Anomaly Detection (MVTec AD)データセット[2]を用いた、異常箇所の特定の検出例を図 1 に示す。

画像内の異常は、取り込む情報量の違いにより、大きく下記二種類に分けられる。キズ、変形など、局所的なパターンのみで判断できる局所異常と、配置間違いなどのコンテキスト的な判断が必要なコンテキスト異常である。図 1 の 1 行目の異常画像（板の穴）は局所異常（キズ）の例である。一方、2 行目（トランジスタ欠品）と 3 行目（3 色ケーブルに同色ケーブルが 2 本ある）の異常画像はコンテキスト異常の例である。なお、コンテキスト異常は、局所的なパターン（例えば、1 本のケーブル）から得られる情報のみで処理すると、正常として判断されてしまう。

上記二種類の異常箇所の特定について、それぞれ特性が存在する。前者の異常に対しては、局所的なパターンに着目することが重要であるため、検出時に使用する画素の範囲を制限する方法が有効と考えられる。一方、後者の異常に対しては、画像内の要素の関係性（コンテキスト情報）を十分に取り込む必要があるため、画像全体または位置に関連する情報を使用する方法が有効と思われる。

本稿では、局所的なパターンに着目する手法に対して、コンテキスト異常の検出精度を改善するためのフレームワークを提案する。また、MVTec-AD[2]データセットを用いた実験により、検出性能が向上することを示す。

†株式会社東芝　研究開発センター　Toshiba Corporation Corporate Research & Development Center

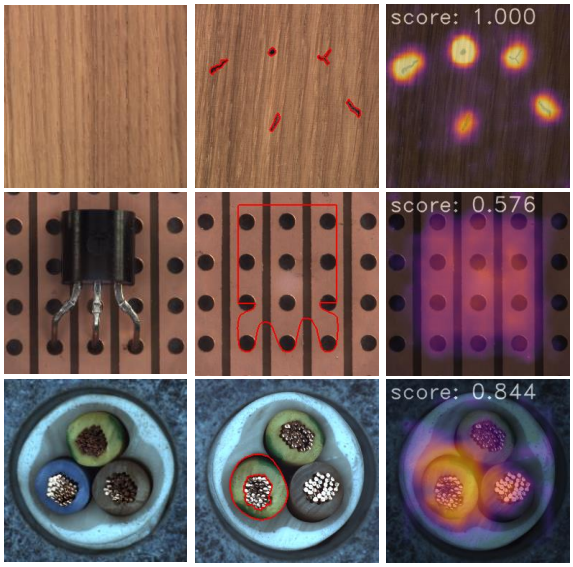

図 1　MVTec-AD を用いた検出の一例
左から：正常画像、異常画像（GT=赤）、検出結果（発火部異常）※中段と下段における正常な場合のコンテキスト条件は、トランジスタ：用品が垂直に置かれて、脚が最下部中心の 3 つにささっている。ケーブル：絶縁被覆の順番は上部中心から時計回りに黄緑⇒灰色⇒青である。

2. 関連研究

近年、深層学習技術の発展により、様々なネットワーク構造を用いた半教師あり学習の異常画像検出手法が提案されている。たとえば、Variational Autoencoders(VAEs)[1]、Generative Adversarial Networks (GANs)[3]などの生成モデルに基づく手法や、深層モデルの埋め込み特徴量を利用する手法などが存在する。

生成モデルに基づく手法では画像を符号化して、再び復元する。正常画像はうまく復元できるために再構成誤差が小さくなるのに対し、異常画像の再構成誤差は大きくなるという仮説に基づく。しかしながら、生成モデルに基づく手法では、復元画像の再現度が不足していることもあり、過検出が多い傾向がある。

深層モデルの埋め込み特徴量を利用する手法においては、既存の学習済みモデルの特徴量に対して、One Class SVM[4]、k-mean[5]や AutoEncoder[2]などを適用する手法が知られている。異常検出では十分な学習データ数の確保が困難であるため、単独ではよい特徴量を得にくいが、上記手法は、ImageNet[6]などの大規模データベースで学習済みのモデルの表現能力の高い特徴量を利用することで、検出精度の向上を図っている。ただし、学習済みモデルの特徴量は畳み込み演算を繰り返すことで算出するのが一般的であり、レセプティブフィールド（ある位置に対する特徴量の算出に利用した画素の範囲）はモデル毎に定まっている。

Copyright © 2021 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

したがって、レセプティブフィールドの大きい学習済みモデルを用いた場合、前述した局所的な情報で判断できる異常に対して、過剰な情報を扱っている問題がある。

上記問題を踏まえて、レセプティブフィールドが小さい学習済み教師モデルから、生徒モデルへ知識の蒸留を用いる手法が提案されている[7][8]。しかしながら、この手法は局所的な情報のみを用いるために、コンテキスト異常に十分対応できないという問題がある。

また、同様にレセプティブフィールドを制限したエンコーダを利用し、最近傍探索[9]を適用する手法も存在する[10]。学習時に Position Classifier を導入することにより、エンコーダが出力する特徴量は位置に関連する情報を含んだものとなる。したがって、検出時には暗に位置情報を含んだ最近傍探索となり、検出精度が向上している。ただし、本手法は正常画像の特徴量を辞書とし、k-NN で最近傍特徴量を探索するため、検出処理時間が問題となる。

3. 提案手法

3.1 提案ネットワークの構造

本稿では、レセプティブフィールドを制限したモデルを利用する手法(文献[7][8])に対して、コンテキスト情報を取り込む方法について提案する。具体的には、文献[10]において学習時にのみ用いていた Position Classifier を、検出時にも利用することで、コンテキスト異常の検出精度の改善を図る。なお、図 2 に示す提案フレームワークは、ベースである局所異常を検出するための教師-生徒モデル(3.1.1 節)と、コンテキスト異常を検出するための Position Classifier による相対位置識別モデル(3.1.2 節)の 2 部分から成る。

3.1.1 教師-生徒モデル

教師-生徒モデルを図 2 の破線内に示す。生徒モデルはレセプティブフィールドの小さいモデルを採用する。画像に対してスライディングウィンドウを行い、入力画像と同サイズの特徴量マップを生成する。Avgpool 層は教師モデルの特徴量マップのサイズと合わせるために使用する。

大量の自然画像(ImageNet[6]など)で事前学習済みの教師モデルの特徴量を教師信号として利用する。学習時は、異常検出向けの学習データ(正常画像のみ)から、教師モデル、生徒モデルそれぞれ特徴抽出を行い、生徒モデルの特徴量が教師モデルの特徴量と一致するように生徒モデルを学習する。検出時は、生徒モデルと教師モデルの特徴量が離れている場合に異常とする。

なお、本稿では教師-生徒モデル単独で構成する手法をベースラインとして扱い、文献[8]の生徒モデルを少し変更した手法に相当する。

3.1.2 相対位置識別モデル

相対位置識別モデルを図 2 の一点鎖線内に示す。中心パッチと近傍パッチにおける生徒モデルの特徴量の差分を、Position Classifier に入力し、パッチペア間の相対位置関係を推測する。Position Classifier は、パッチペア間の特徴量の差分からその相対位置関係を推定する識別モデルであり、自己教師で学習することが可能である。画像から近傍パッチペアを抽出し、その位置関係を示す正解ラベルを教師信号として生成する。学習時は、Position Classifier が正常画像の近傍パッチ間の相対位置関係を正しく推定するように学習する。検出時は、Position Classifier が近傍パッチ間の相対位置関係を正しく推定できない場合を異常とする。

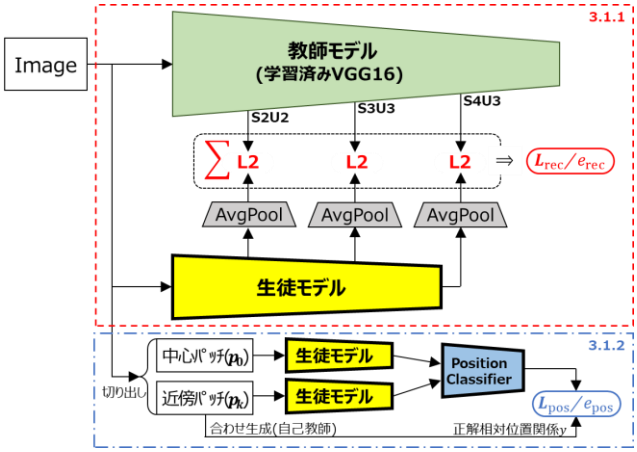

図 2 提案手法のフレームワーク

3.2 損失関数

学習時の損失関数は、教師モデルと生徒モデル間の損失関数$\mathcal{L}_{\mathrm{rec}}$と Position Classifier の損失関数$\mathcal{L}_{\mathrm{pos}}$から成る。テクスチャ系の画像では、いずれのパッチも類似したパターンとなり、そもそも相対位置を推定することが困難な場合がある。そのような場合に自動的に重みを下げて過度な学習をしないようにしている[12]。

$$\mathcal{L} = \frac{1}{2\lambda_{\mathrm{rec}}^2}\mathcal{L}_{\mathrm{rec}} + \frac{1}{2\lambda_{\mathrm{pos}}^2}\mathcal{L}_{\mathrm{pos}} + \log\lambda_{\mathrm{rec}} + \log\lambda_{\mathrm{pos}}$$

ここで、λ_{rec}、λ_{pos}は重み係数である。$\mathcal{L}_{\mathrm{rec}}$は教師モデルと生徒モデルの特徴量により計算される。

$$\mathcal{L}_{\mathrm{rec}} = \sum_{\ell}\sum_{(r,c)} \left\| f_{\theta}^{\ell}{}_{(r,c)} - f_{0}^{\ell}{}_{(r,c)} \right\|_2^2$$

ただし、$f_{0(r,c)}^{\ell}$、$f_{\theta(r,c)}^{\ell}$はそれぞれ教師モデル、生徒モデルの座標(r,c)の画素におけるℓ層レイヤーの特徴量である。

$\mathcal{L}_{\mathrm{pos}}$は下記の式の通り、Position Classifier による位置推定の Cross Entropy ロスで計算される。

$$\mathcal{L}_{\mathrm{pos}} = \frac{1}{N}\sum_{\boldsymbol{p_0}} \mathrm{CrossEntropy}\left(y, C_{\phi}\left(f_{\theta}(\boldsymbol{p_0}) - f_{\theta}(\boldsymbol{p_k})\right)\right)$$

ただし、f_{θ}は生徒モデルの出力特徴量である。$\boldsymbol{p_0}$は画像内のランダム位置から切り出す中心パッチであり、$\boldsymbol{p_k}$は、ランダムで切り出す$\boldsymbol{p_0}$に近傍するパッチである。Nは、1 枚の画像から切り出すパッチペア$(\boldsymbol{p_0}, \boldsymbol{p_k})$の数である。$C_{\phi}$は Position Classifier モデルである。yは自己生成した$\boldsymbol{p_0}$と$\boldsymbol{p_k}$の相対位置関係を示す正解ラベルである。

3.3 検出

検出時は、教師モデルと生徒モデルの各層特徴量が離れている場合、または Position Classifier が近傍パッチ間の相対位置を正しく推定できない場合を異常とする。座標(r,c)の画素の異常スコア$e_{(r,c)}$は、その画素における教師モデルと生徒モデルの特徴量間の L2 距離$e_{\mathrm{rec}(r,c)}$と、Position Classifier の識別誤差$e_{\mathrm{pos}(r,c)}$から計算される。

$$e_{(r,c)} = \frac{e_{\mathrm{rec}(r,c)} - \mu_{\mathrm{rec}}}{\sigma_{\mathrm{rec}}} + \frac{e_{\mathrm{pos}(r,c)} - \mu_{\mathrm{pos}}}{\sigma_{\mathrm{pos}}}$$

ただし、μ_{rec}、μ_{pos}、σ_{rec}、σ_{pos}はすべての学習データの全画素におけるそれぞれの異常度の平均値と標準偏差であり、L2 距離と識別誤差に対して正規化を行う。

なお、$e_{\mathrm{rec}(r,c)}$ は下記の式の通り、各画素の教師モデルと生徒モデルの特徴量により計算される。

Copyright © 2021 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

$$e_{\mathrm{rec}(r,c)} = \sum_{\ell} \left\| f_{\theta}^{\ell}{}_{(r,c)} - f_{0}^{\ell}{}_{(r,c)} \right\|_2^2$$

$e_{\mathrm{pos}(r,c)}$ は下記式の通り、画素(r,c)を中心としたパッチ$\boldsymbol{p_0}$と、各方向の近傍パッチ$\boldsymbol{p_k}$間の異常度$e_{\mathrm{pos}(\boldsymbol{p_0},\boldsymbol{p_k})}$により算出される。

$$e_{\mathrm{pos}_{(r,c)}} = \frac{1}{n}\sum_{\boldsymbol{p_k}} e_{\mathrm{pos}_{(\boldsymbol{p_0},\boldsymbol{p_k})}}$$

ただし、nは 1 つの中心パッチ$\boldsymbol{p_0}$に対して抽出する近傍パッチ$\boldsymbol{p_k}$の数である。なお、中心パッチ$\boldsymbol{p_0}$または近傍パッチ$\boldsymbol{p_k}$のいずれかが入力画像の範囲以外にある場合は$e_{\mathrm{pos}(\boldsymbol{p_0},\boldsymbol{p_k})} = 0$とし、計算から除外する。

なお、$e_{\mathrm{pos}(\boldsymbol{p_0},\boldsymbol{p_k})}$は、Position Classifier による正解クラスの推定確率$p(y)$により計算される。

$$e_{\mathrm{pos}_{(\boldsymbol{p_0},\boldsymbol{p_k})}} = 1 - p(y)$$

4. 実験

4.1 評価データセット

MVTec-AD データセットは製品検査を目的としたデータセットであり、5 種類のテクスチャと 10 種類の物体の高解像度画像を含むデータセットである。学習用データには約 250 枚の正常画像、評価用データは約 100 枚の正常画像と様々な種類の異常画像が含まれている。

なお、本データセットでは用品の検査を想定しているため、用品は画像の中心に配置されている。また、本稿が着目しているコンテキスト的な異常を含むカテゴリとしては、Transistor、Cable などがある。

4.2 評価指標

評価指標として、下記の 3 つを用いた。

・Image-AUC(画像単位の ROCAUC)：画像に対する異常スコアから算出する ROC 曲線の下側面積。本稿では、全画素の異常スコアの最大値を画像に対する異常スコアとして用いた。

・Pixel-AUC(画素単位の ROCAUC)：各画素に対する異常スコアから算出する ROC 曲線の下側面積。この評価指標では異常領域内の画素の数だけ正検出の評価が行われるので、結果として面積が大きい異常領域ほど重要視される。

・PRO(per-region-overlap)[2]：正解の異常領域について結合領域毎に正しく異常画素として検出できた割合を計算し、その平均を縦軸、正常画素に対する誤検出率(FPR)を横軸とした曲線の、誤検出率(FPR)が 30%までの積分値を正規化して利用する。

Pixel-AUC の縦軸は全異常画素に対する正検出率(TPR)であるのに対して、PRO は各異常領域における異常画素に対する正検出率(TPR)の平均値であるため、PRO は Pixel-AUC の面積バイアスを解決したものと言える。

4.3 実験設定

ベースラインと提案手法の実験設定は下記とする。教師モデルは Pytorchcv[13] ライブラリの学習済み VGG16[11]とし、Stage2Unit2、Stage3Unit3、Stage4Unit3 の特徴量を使用する。生徒モデルは教師モデルのダウンサンプリングに合わせて、複数出力レイヤーを持つ、Conv 層と Maxpool 層で構成されたモデルとする。Position Classifier は 3 つの全結合層で構成され、提案手法のみ適用される。

表 1　ベースラインと提案手法の実験結果

	Image-AUC			Pixel-AUC			PRO		
	Base	ProP.	Diff.	Base	ProP.	Diff.	Base	ProP.	Diff.
Carpet	87.2%	89.8%	**2.6pt**	96.6%	97.7%	**1.1pt**	89.7%	92.6%	**2.9pt**
Grid	98.7%	99.0%	**0.3pt**	98.1%	98.2%	**0.1pt**	94.2%	94.3%	**0.1pt**
Leather	97.9%	98.2%	**0.3pt**	99.3%	99.3%	**0.0pt**	98.2%	98.3%	**0.1pt**
Tile	88.5%	88.7%	**0.2pt**	89.4%	90.7%	**1.3pt**	73.7%	74.1%	**0.4pt**
Wood	99.0%	99.1%	**0.0pt**	95.5%	95.6%	**0.1pt**	92.5%	92.8%	**0.3pt**
Bottle	99.9%	99.9%	**0.1pt**	96.9%	97.3%	**0.5pt**	92.5%	92.3%	-0.2pt
Cable	60.5%	89.1%	**28.6pt**	89.9%	97.3%	**7.4pt**	77.5%	87.2%	**9.7pt**
Capsule	95.5%	95.0%	-0.6pt	98.4%	98.5%	**0.2pt**	96.1%	95.9%	-0.2pt
Hazelnut	99.9%	100.0%	**0.1pt**	98.5%	98.7%	**0.2pt**	96.6%	96.1%	-0.5pt
Metal_nut	97.0%	97.9%	**0.8pt**	96.1%	97.8%	**1.8pt**	91.7%	91.1%	-0.6pt
Pill	90.0%	90.0%	**0.0pt**	98.2%	97.7%	-0.6pt	96.8%	95.8%	-1.0pt
Screw	95.5%	95.1%	-0.4pt	99.4%	99.3%	-0.1pt	97.0%	96.6%	-0.4pt
Toothbrush	97.9%	98.6%	**0.7pt**	98.6%	98.7%	**0.1pt**	93.1%	92.9%	-0.2pt
Transistor	88.6%	92.2%	**3.6pt**	79.3%	93.4%	**14.1pt**	68.7%	88.3%	**19.6pt**
Zipper	92.2%	93.2%	**1.0pt**	96.2%	96.7%	**0.4pt**	86.2%	87.7%	**1.5pt**
Mean	92.6%	95.0%	**2.5pt**	95.4%	97.1%	**1.8pt**	89.6%	91.8%	**2.1pt**
Mean(cxt)	74.5%	90.6%	**16.1pt**	84.6%	95.4%	**10.8pt**	73.1%	87.8%	**14.7pt**

入力画像はバイリニア補間により 256×256 へリサイズする。Ground Truth は面積平均補間によりリサイズする。学習のミニバッチサイズは 1 とするが、Position Classifier を学習するために、1 枚の画像から切り出すパッチペア数は N=64 とする。なお、パッチサイズは生徒モデルのレセプティブフィールドの大きさであり、本稿では 65 である。また、近傍パッチは中心パッチの周囲（上下、左右、および対角線）の 8 方向(n=8)の近傍パッチとする。学習時はランダムで 1 つを選び、検出時は 8 方向をそれぞれ計算する。

Optimizer は Adam を使用し、学習率は 0.0005、weight decay は 0.00001 である。学習回数は 30000 とする。

5. 結果

5.1 実験結果

ベースラインと提案手法の実験結果を表 1 にまとめる。太字は同カテゴリ同指標で提案手法がベースラインより改善ができたことを示す。ベースラインに比べて提案手法は、各評価指標において、全カテゴリ平均値の高い結果が得られた。加えて、半分以上のカテゴリに対して、すべての評価指標の値が向上できた。

さらに、表内下線のデータに示すように、本報告が着目しているコンテキスト異常の検出が必要な Cable と Transistor のカテゴリに対して、各評価指標の値が大幅に改善された。Position Classifier の導入により、コンテキスト系カテゴリに対する Image-AUC の平均が約 16pt、Pixel-AUC の平均が約 11pt、PRO の平均が約 15pt 向上していることが分かる。

図 3 に異常検出の可視化結果の一例を示す。左の 2 列が局所異常、右の 2 列がコンテキスト異常の検出結果である。なお、提案手法において、e_{rec}とe_{pos}の何れの異常によるものかを確認するために、それぞれのヒートマップについても可視化した。局所異常については、ベースラインと提案手法による検出結果が同等であることがわかる。一方、コンテキスト異常において、Position Classifier の導入により、検出された異常領域が GT 領域に近づいてる結果が得られた。

Copyright © 2021 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

表 2 アブレーションスタディの結果

手法	条件		Image-AUC		Pixel-AUC		PRO	
	L_{pos}	e_{pos}	mean	std.	mean	std.	mean	std.
①ベースライン	✗	✗	92.6%	5.3	95.4%	1.3	89.6%	**1.8**
②提案手法(e_{rec}のみ)	✓	✗	92.5%	**4.4**	95.3%	1.1	89.5%	2.1
③提案手法	✓	✓	**95.0%**	4.6	**97.1%**	**1.0**	**91.7%**	2.5

Grid と Hazelnut の異常検出結果において、異常箇所であるキズ部位は、e_{rec}により検出された。また、e_{pos}による異常スコアマップでは、相対位置を推定することが困難なカテゴリにおいても、誤検出がされていないことが確認できた。

Transistor の異常検出結果において、用品の輪郭線を含む領域は、類似するパターンが学習データにないため、ベースラインすなわちe_{rec}でも異常として検出できた。ただ、トランジスタの頭部の黒い領域は、類似するパターンが学習データに存在するため、ベースラインでは正常として判断される。一方、この異常領域とその周囲の近傍パッチの相対位置関係は、Position Classifier の学習データに、類似する関係が存在しないため、提案手法のe_{pos}では、異常として検出できた。同様に、Cable の欠損についても、ベースライン(e_{rec})では、欠損部の輪郭線のみ検出されたが、提案手法(e_{pos})では、欠損部の中心まで異常として検出できた。

5.2 アブレーションスタディ

アブレーションスタディの結果を表 2 に示す。ベースラインと提案手法の各条件において、それぞれ 6 回の実験結果の各評価指標の平均値 mean(↑)と標準偏差 std.(↓)についてまとめる。

①と②の比較は Position Classifier の導入による生徒モデルへの影響に関する考察である。提案手法の学習において、L_{rec}とL_{pos}は生徒モデルを共有しているため、Position Classifier の学習により、e_{rec}による異常検出の性能が落ちる懸念があった。しかし、比較結果から、ベースラインで学習した生徒モデルと提案手法で学習した生徒モデルにおけるe_{rec}による各評価指標およびその標準偏差は、ほぼ同等であることがわかった。したがって、Position Classifier を導入しても、e_{rec}による異常検出の精度が低下することはほぼないことが確認できた。

②と③の比較は Position Classifier を検出時にも導入したことによる検出結果への影響に関する考察である。e_{pos}を利用しない場合に比べて提案手法では、各評価指標の値が約 2pt 向上した。標準偏差は少し悪化しているが、全体としてすべての評価指標は改善しており、Position Classifier を検出時に導入するのは有効であることがわかった。

6. おわりに

教師モデルと生徒モデルの特徴量の L2 距離を用いた異常検出手法において、Position Classifier を学習時と検出時に導入することにより、コンテキスト異常の検出精度を向上する異常検出手法について提案した。

MVTec-AD データセットを用いた実験を通じて考察を行い、提案フレームワークは異常検出、特にコンテキスト異常の検出に有効であることを示した。

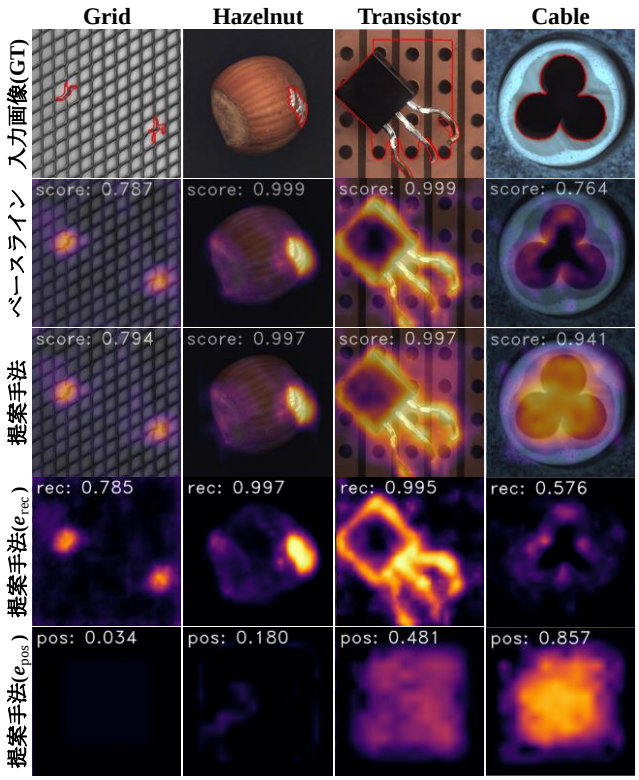

図 3　可視化結果

参考文献

[1] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, Nassir Navab, “Deep Autoencoding Models for Unsupervised Anomaly Segmentation in Brain MR Images”, arXiv:1804.04488 (2018)
[2] Paul Bergmann, Michael Fauser, David Sattlegger, Carsten Steger, “MVTec AD—A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection”, CVPR (2019).
[3] Thomas Schlegl, Philipp Seeb¨ock, Sebastian M Waldstein, Ursula Schmidt-Erfurth, Georg Langs, “Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery”, IPMI, Springer, p. 146–157 (2017).
[4] Philippe Burlina, Neil Joshi, I-Jeng Wang. “Where’s wally now? deep generative and discriminative embeddings for novelty detection”, CVPR (2019).
[5] Paolo Napoletano, Flavio Piccoli, Raimondo Schettini, “Anomaly Detection in Nanofibrous Materials by CNN-Based Self-Similarity”. Sensors, 18(1):209 (2018).
[6] Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, Advances in Neural Information Processing Systems, 25:1097–1105 (2012).
[7] Paul Bergmann, Michael Fauser, David Sattlegger, Carsten Steger, “Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings”, CVPR (2020).
[8] Guodong Wang, Shumin Han, Errui Ding, Di Huang, “Student-Teacher Feature Pyramid Matching for Unsupervised Anomaly Detection”, arXiv:2103.04257 (2021).
[9] David M.J. Tax, Robert P.W. Duin. "Data description in subspaces." ICPR-2000. Vol. 2. IEEE, (2000).
[10] Jihun Yi, Sungroh Yoon. "Patch SVDD: Patch-level SVDD for Anomaly Detection and Segmentation", ACCV (2020).
[11] Karen Simonyan, Andrew Zisserman, “Very deep convolutional networks for large-scale image recognition”, arXiv:1409.1556 (2014).
[12] Alex Kendall, Yarin Gal, Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics", CVPR (2018).
[13] Oleg Sémery. “Convolutional neural networks for computer vision", URL: https://github.com/osmr/imgclsmob. [Online; accessed 29-May-2020]

Copyright © 2021 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.