多重解像度マップを持つ補集合 GAN を用いた正常データのみの学習による外観検査法*

中 塚 俊 介** 加 藤 邦 人***

Visual Inspection Method by Learning Only Normal Data using Complementary GAN with Multi-scale Map

Shunsuke NAKATSUKA and Kunihiro KATO

In the case of industrial products, visual inspection is essential to improve the quality of products. In recent years, automation with neural networks has been considered. However, the conventional model discriminating good or defective products requires a large amount of good and defective samples for learning. In fact, it is difficult to ensure a large amount of defective samples. Therefore, in this research, anomaly (= defective samples) are detected by modeling the normal distribution and its complement from only a large amount of good products. Since the defect is a part of the image and its size varies, we propose the structure of Multi-scale Patch Discriminator in this paper.

Key words: visual inspection, neural networks, anomaly detection, Generative Adversarial Networks, multi scale

1. 緒 言

工業製品の製造工程において、出荷製品の品質安定化のために検査は必要不可欠である。とくに外観検査工程では人による目視検査が主であるが、検査員毎の判断基準のばらつきが長年の課題であった。近年では深層学習技術の進歩により、これまで困難とされてきた目視検査工程でも自動化が検討されている。従来の深層学習では正常品と異常品を識別するために大量のサンプルが必要であるが、現実には大量の異常品サンプルを確保することは困難である。そのため、正常品のみ、もしくは正常品と少数の異常品から良否識別できる枠組みが求められている。また、画像検査においては、1枚の異常画像の中に正常な領域と異常な領域が混在していることがある。このとき、異常な領域が占める割合が大きければ検出は容易であるが、小さい場合は画像のほとんどの部分が正常なため検出は難しい。また、異常には様々な種類が存在し、それぞれを検出するのに適した解像度が存在するはずである。そのため、本稿では Complementary GAN¹⁾に Multi-scale Patch の枠組みを加えたモデルに正常品のみを学習させ、正常分布とその補集合分布をモデリングし異常検知を行う手法を提案する。

本稿の貢献は以下の2点である。

- Complementary GAN の Discriminator に Multi-scale Patch を適用することで異常ごとに適した解像度での出力を可能にした。
- 画像のような高次元データから情報の圧縮無しに直接、異常検出を行った。

1 点目は上述の通り、多重解像度戦略を Complementary GAN に取り入れたことである。また、Discriminator の出力を Multi-scale Patch にするにあたり、目的関数の変更、画像検査に適した異常度の算出方法を導入した。

2 点目に関しては、本来 Complementary GAN は時系列データに

対して LSTM Autoencoder²⁾を適用し得られた潜在ベクトルに対して適用していた。しかし、異常検知において Autoencoder を用いた情報圧縮を行うと、異常データを入力した際に異常の情報がドロップしてしまい、後段の異常検出において異常を見逃してしまうことが知られている³⁾。そのため、本稿では画像に対して情報の圧縮無しに Complementary GAN を適用する。また Complementary GAN を直接画像に適用するにあたり、従来の Complementary GAN の Feature Matching は画像のような高次元データには不十分であり、特に画像検査で重要となるエッジや小さい欠陥の情報が消失してしまう可能性がある。1 点目で導入した Multi-scale Patch の枠組み、目的関数はこの問題への解決策であり、本研究の新規性でもある。

2. 背景技術の概要

本章では関連手法を述べた後、背景技術である Generative Adversarial Networks (GAN)⁴⁾、Complementary GAN¹⁾の概要を述べる。それぞれの数式的説明については3章にて述べる。

2.1 関連手法

近年、正常を深層学習モデルを用いてモデリングし異常を検出する手法が多数提案されている。大きな枠組みとして、Autoencoder を用いた手法と GAN⁴⁾を用いた手法が存在する。

Autoencoder を用いた手法では、正常のみを学習させ、正常のみを復元可能なモデルを学習させる。推論時にはテスト画像の入力と復元結果の差分値の総和を異常度とする。この手法は非常に単純であり大きな異常を検出することに適しているが、Autoencoder の性質上、入出力で低周波成分が差分値として大きく出てしまうことに加えて、Autoencoder が汎化してしまうと異常なテスト画像までも復元できてしまうという問題がある。Autoencoder の派生として、Deep Autoencoding Gaussian Mixture Model (DAGMM)⁵⁾が提案されている。この手法では Autoencoder の潜在ベクトルと入出力の差分値の総和を結合し、特徴ベクトルとし、その特徴ベクトルに対して GMM を適用することで対数尤度 (異常度) を算出するモデルである。このモデルは異常なのに復元されてしまうデータを潜在ベクトルで、復元されないデータを入出力の差分値で検

* 原稿受付 令和2年 5月 8日

掲載決定 平成22年 9月 15日

** 正 会 員 パナソニック株式会社 (大阪府門真市大字門真 1048 番地)

*** 正 会 員 岐阜大学 (岐阜県岐阜市柳戸 1-1)

出することを目的としている。しかし、潜在ベクトルと差分値を結合し、GMM に入力しても潜在ベクトルの情報はあまり重要視されず差分値の寄与率が大きいため、潜在ベクトルの次元数が高いと学習が失敗してしまう。

GAN を用いた手法としては、AnoGAN⁹⁾や Anomaly Detection GAN (ADGAN)⁶⁾が存在する。どちらも正常データのみを使って、正常を生成する GAN を学習させる。推論時にはテストデータを入力とし、それに対応する潜在ベクトルを Inverse Mapping により求め、得られた潜在ベクトルから生成したデータと入力との差分値、潜在空間上での距離を用いて異常を検出する。GAN は元来、Inverse Mapping の機構を持たないため、両手法では差分値、潜在空間上が小さくなるように潜在ベクトルを勾配降下法により探索することで Inverse Mapping を実現した。しかし、これは推論時にも学習が必要であることを意味し、短いタクトを要求される外観検査で使用することは難しい。

上記の2つの手法に共通していることは、正常データのみから学習し、その潜在ベクトルと差分値を使うことで正常モデルを構築するという点である。上記のモデルが算出する異常度と、人間の知覚する正常データと異常データとの差が一致することは多くない⁷⁾。そのため、差分や潜在ベクトルではなく人間の知覚に近いとされる深層学習モデルを用いて、その差を算出するのは自然なことである。しかし、異常検知においては異常が訓練データ内に存在しないため、その差を算出することは難しい。そのため本研究では、正常と正常ではない（言い換えると補集合）をモデリングする Complementary GAN を用いて、異常検知を行う。

2.2 Generative Adversarial Networks

Generative Adversarial Networks (GAN)⁴⁾は、Goodfellow らによって提案されたニューラルネットワークである。GAN では生成モデルの1つで、データ分布 p_{data} からデータの生成過程を学習することで、 p_{data} からサンプリングされたようなデータを生成することが可能である。GAN は Generator と Discriminator と呼ばれる役割の異なる2つのネットワークを有する。Generator には潜在空間 p_z からサンプリングされたベクトル z が入力され、データ $x' \sim p_g$ を生成する。ここで、 p_g は生成分布である。Discriminator にはデータ分布からサンプリングされたデータ $x \sim p_{data}$ と $x' \sim p_g$ が入力され、入力が $x \sim p_{data}$ (real) であるか、 $x' \sim p_g$ (fake) であるかを識別する。Generator が Discriminator を騙すような p_{data} に近いデータを生成するよう Generator を最適化し、Discriminator が real と fake をより良く識別できるように最適化を行う。このように学習則の異なる2つのネットワークを敵対的に交互に学習することで、最終的には生成分布 p_g はデータ分布 p_{data} に近い分布となる。

Discriminator は入力が real であるか fake であるかをより良く識別できるように最適化が行われる。そのため、Discriminator の識別結果を正常度として、異常検知を行う手法⁸⁾も提案されている。しかし、Discriminator が識別しているのは、 $x \sim p_{data}$ (real)、 $x' \sim p_g$ (fake) であり、異常データの情報は一切考慮されていない。そのためこの手法⁸⁾では、Discriminator の fake として、生成データと異常データの両方を与えている。また、GAN の学習が進むと Generator が生成するデータが real に近くなり、Discriminator は $x \sim p_{data}$ (real)、 $x' \sim p_g$ (fake) の識別が困難になってしまう。

Fig. 1 Difference between CGAN and RGAN before and after training

Fig. 2 Conceptual diagram representing the difference between RGAN (left) and CGAN (right). The gray dots, green and the pink dots indicate data sampled from the data distribution, the generated distribution and the anomalous distribution respectively. In RGAN, the distribution and the generated distribution are close to each other, while in CGAN, the generated distribution is away from the data distribution

2.3 Complementary GAN

Complementary GAN (CGAN)は通常の GAN (Regular GAN; RGAN)とは異なり, 生成分布 p_g をデータ分布 p_{data} に近づけるように最適化を行うのではなく, p_{data} の補集合分布 p^* に近づけるように最適化を行う。CGAN と RGAN の学習前, 学習後の生成分布の違いの概念図を図 1 に示す。

RGAN とは異なり, CGAN では生成分布 p_g は補集合分布 p^* に近づき, Discriminator は $x \sim p_{data}$ (real), $x' \sim p_g$ (fake=complementary) を識別することができる。また, p^* はデータ分布 p_{data} の補集合分布であるため, p_{data} を正常分布とするならば, p^* は正常ではない分布, 言い換えると異常分布であるといえる。そのため, $x \sim p_{data}$ (real), $x' \sim p_g$ (fake=complementary) を識別することができる Discriminator は異常検知において有用であるといえる。この概念図を図 2 に示す。

3. 背景技術の数式的説明

3.1 Generative Adversarial Networks

前章では, Discriminator に対しては入力 $x \sim p_{data}$ (real)であるか, $x' \sim p_g$ (fake)であるかを良く識別できるように最適化を行い, Generator に対しては Discriminator を騙すような p_{data} に近いデータを生成するように最適化していると述べた。

しかし実質的には, Discriminator は p_g と p_{data} の分布間がどれだけ離れているかを正確に測るよう学習を行っている。Generator は Discriminator によって算出された“どれだけ離れているか”を最小化するように p_g を更新する。Goodfellow らは, p_g と p_{data} の分布間がどれだけ離れているかを表す指標として, Jensen-Shannon Divergence (JSD)を用いた。JSD は分布間の相対エントロピーを表す Kullback-Leibler Divergence (KLD)の非対称性を考慮したダイバージェンスである。このとき, GAN の目的関数は式(1)となる。

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}} [\log D(x)] + \mathbb{E}_{x' \sim p_g} [\log 1 - D(x')] \quad (1)$$

ここで, G は Generator の演算であり, D は Discriminator の演算である。 D の出力層には活性化関数として, sigmoid 関数を適用することで, 出力は $[0, 1]$ になるため, Discriminator の出力は入力 x が real である確率を表す。このように, Generator は式(1)を最小化し, Discriminator は式(1)を最大化するミニマックスゲームの枠組みになっているため, Adversarial Learning と呼ばれる。

3.2 Complementary GAN

p_{data} の補集合分布 p^* を

Fig. 3 Conceptual diagram of p^* . The gray curve represents the probability density function of the data distribution p_{data} and the pink curve represents the probability density function of p^* . In this case, a normal distribution is assumed as the data distribution

$$p^*(x') = \begin{cases} \frac{1}{\tau p_{data}(x')}, & \text{if } p_{data}(x') > \epsilon \\ C, & \text{otherwise} \end{cases} \quad (2)$$

と定義する。ここで, ϵ は生成データ x' を確率変数とした確率密度 $p_{data}(x')$ に対するしきい値であり, これを超える生成データは high density であるとする。 τ は正規化定数であり, C は非常に小さな定数である。 p^* の概念図を図 3 に示す。

p_g を p^* に近づけるために, Generator は p_g と p^* 間の KL Divergence $D_{KL}[p_g || p^*]$ を最小化するように最適化を行う。

$$\begin{aligned} D_{KL}[p_g || p^*] &= -\mathcal{H}(p_g) - \mathbb{E}_{x' \sim p_g} [\log p^*(x')] \\ &= -\mathcal{H}(p_g) + \mathbb{E}_{x' \sim p_g} [\log p_{data}(x') \mathbb{I}[p_{data}(x') > \epsilon]] \\ &\quad + \mathbb{E}_{x' \sim p_g} [\mathbb{I}[p_{data}(x') > \epsilon] \log \tau - \mathbb{I}[p_{data}(x') \leq C]] \end{aligned} \quad (3)$$

ここで, $\mathcal{H}$ はエントロピーを表し, $\mathbb{I}$ は指示関数を表す。第 3 項目に関しては, Generator の重みパラメータで偏微分した際には定数となるため, 最適化の際には無視することができる。また, $\mathcal{H}$ は計算不可能であるため, pull-away term (PT)⁹⁾により近似する。

$$L_{PT} = \frac{1}{m(m-1)} \sum_i \sum_{j \neq i} \left(\frac{f(x'_i)^T f(x'_j)}{|f(x'_i)| |f(x'_j)|} \right)^2 \quad (4)$$

ここで, m はミニバッチサイズであり, f は Discriminator の出力層直前の層の出力である。

通常, データ分布 p_{data} を定式化することは困難である。そのため, CGAN では通常の GAN (今後は RGAN とする)の Discriminator の識別結果を用いる。

$D_{KL}[p_g || p^*]$ を最小化することで, p_g は p^* に近づくが, p_{data} から離れすぎてしまうため, 生成データが特徴空間上では $x \sim p_{data}$ と同じ特徴空間上に制限されるように Feature Matching Loss¹⁰⁾を用いる。

$$L_{FM} = \left| \mathbb{E}_{x' \sim p_g} [f(x')] - \mathbb{E}_{x \sim p_{data}} [f(x)] \right|^2 \quad (5)$$

以上より, CGAN の Generator の目的関数は, RGAN の Discriminator を D_R とすると, 式(6)で表すことができる。

Fig. 4 Structure of the Discriminator in MSP-CGAN. Unlike the usual Discriminator, it outputs a multi-scale Discrimination Map

$$\min_G L_{PT} + L_{FM} + \mathbb{E}_{x' \sim p_g} [\log D_R(x') \mathbb{I}[D_R(x') > \epsilon]] \quad (6)$$

Discriminator は $x \sim p_{data}$ (real) と $x' \sim p_g$ (fake) とをうまく識別できるように最適化を行う。そのため、目的関数は

$$\max_D \left[\mathbb{E}_{x \sim p_{data}} [\log D(x)] + \mathbb{E}_{x' \sim p_g} [\log(1 - D(x'))] + \lambda \mathbb{E}_{x \sim p_{data}} [D(x) \log D(x)] \right] \quad (7)$$

と表される。第1項目、第2項目はRGANのDiscriminatorの目的関数に等しい。第3項目はテスト時に正常データが入力された際に出力値が大きくなることを促す項であり、 λ はその強さを調節するハイパーパラメータである。

CGANにおいてはRGANと異なり、最終的には p_g が p^* に近づき、Discriminatorは $x \sim p_{data}$ (real) と $x' \sim p_g \approx p^*$ (fake)の識別が可能であり、異常データは p^* に存在するはずなので、 $D(\hat{x})$ は正常度であり、未知データ $\hat{x}$ に対する異常度は

$$a(\hat{x}) = 1 - D(\hat{x}) \quad (8)$$

で表される。

4. Multi-scale Patch Complementary GAN (MSP-CGAN)

4.1 Multi-scale Patch

RGANでも、CGANでもDiscriminatorの出力はスカラーとなる。画像データセットにおける異常検知においては、画像全体が異常データである場合は少なく、局所的にデータを見ると、正常データと区別がつかない場合がある。言い換えると、1枚の異常画像の中に正常な領域と異常な領域が混在していることがほとんどである。このとき、異常な領域が占める割合が大きければ、CGANの出力(正常度)は低くなるが、異常な領域が占める割合が小さい場合、正常度が大きくなってしまふ。しかし、工業製品や食品、医用画像といった異常画像検知においては、少しでも異常な領域があれば、異常と判断されるべきである。そのため、Discriminatorの出力をスカラーではなく、多重解像度構造を持つPatch GAN⁽¹⁾の枠組みを適用し、Discrimination Map $\mathbb{D}(\bullet) \in \mathbb{R}^{h_m \times w_m}$ とすること

で、この問題を解決する。 h_m, w_m はそれぞれ、Mapの高さと幅である。また、画像内の異常には様々な種類があり、その検出において最適解像度は異なる。そのため、本手法ではMulti-scaleなDiscrimination MapをDiscriminatorに出力させる。具体的には、図4のようにDiscriminatorの異なる解像度を持つ複数の中間層から、 1×1 Convolutionを用いることで、階層ごとの出力とする。

オリジナルのCGANでは正常な1次元データで学習したAutoencoderのEncoderを用い、入力データを更に圧縮することで得られる低次元の特徴ベクトルに対して、CGANを適用していた。本手法ではAutoencoderによる圧縮無しに画像を入力としている。これは、AutoencoderのEncoderが抽出する特徴はあくまで正常の潜在変数であり、異常の情報が入力から消失されてしまう場合があるからである³⁾。画像を直接CGANに適用するにあたり、本手法では目的関数・異常度の算出の2つの工夫を施した。

4.2 MSP-CGANの目的関数

CGANではDiscriminatorの出力層の直前の層に対してFeature Matchingを適用している。本手法では画像を入力としているため、出力層直前の情報はセマンティックな情報であり、異常画像検知特に外観検査で重要となるエッジなどのプリミティブな情報が消失している可能性が高い。そのような重要な情報が消失した状態ではFeature Matchingの効果が低下し、 p_g が p_{data} から離れすぎてしまう。そのため、本手法ではDiscriminatorがMulti-scaleな複数のDiscrimination Mapを出力するため、それぞれの直前の層をFeature Matchingに適用することで p_g が p_{data} から過度に離れることを防止する。

本手法のGeneratorの目的関数は

$$\min_G \frac{1}{R} \sum_{r=1}^R L_{PT_r} + \frac{1}{R} \sum_{r=1}^R L_{FM_r} + \mathbb{E}_{x' \sim p_g} [\log D_R(x') \mathbb{I}[D_R(x') > \epsilon]] \quad (9)$$

$$L_{PT_r} = \frac{1}{m(m-1)} \sum_i^m \sum_{j \neq i}^m \left(\frac{f_r(x'_i)^T f_r(x'_j)}{|f_r(x'_i)| |f_r(x'_j)|} \right)^2 \quad (10)$$

$$L_{FM_r} = \left| \mathbb{E}_{x' \sim p_g} [f_r(x')] - \mathbb{E}_{x \sim p_{data}} [f_r(x)] \right|^2 \quad (11)$$

で表され、Discriminator の目的関数は

$$\max_D \frac{1}{R} \sum_{r=1}^R \left[\mathbb{E}_{x \sim p_{data}} [\log D_r(x)] + \mathbb{E}_{x' \sim p_g} [\log (E_r - D_r(x'))] \right] + \lambda \mathbb{E}_{x \sim p_{data}} [D_r(x) \log D_r(x)] \quad (12)$$

と表される。ここで、 D_r は r 番目の Discrimination Map を表し、 f_r は D_r の直前の Feature Map をベクトルに変形したものである。 E_r は D_r と同じ形状の行列であり、その要素は全て 1 の 1 Fill Map である。 R は階層の総数である。

4.3 異常度の算出

前述の通り、異常画像検知においては様々な大きさの欠陥が存在し、例えばその欠陥が微小な領域であっても異常と判定されるべきである。本手法ではそのような異常画像検知に適した異常度の算出を行う。

本手法では、Discriminator の出力はスカラーではなく、Multi-scale の Discrimination Map である。そのため、オリジナルの CGAN のように 1 次元の出力ではなく、正常度の集合が出力される。複数階層の Discrimination Map 中の全要素と E_r から算出される異常度の中で最も大きい値を異常度とする。未知データ $\hat{x}$ に対する異常度は

$$\alpha(\hat{x}) = \max\{E_1 - D_1(\hat{x}), E_2 - D_2(\hat{x}), \dots, E_R - D_R(\hat{x})\} \quad (13)$$

で表される。

5. 実験

5.1 データセット

本実験では、2 種類のデータセットを用いる。1 つ目は LED チップ画像のデータセットであり、2 つ目は bottle である。

5.1.1 LED チップ画像

LED チップ画像データセットは、 64×64 の画像サイズのモノクロ画像データセットである。図 5 にその良品画像（正常）と不良品画像（異常）を示す。不良品の中には異物、欠けの 2 つのモードが存在する。異物は埃やごみ、フラックスが付着したもので画像には黒い塊、あるいは細長い線状に現れる。欠けは LED チップの端が欠けチップの 4 角形が崩れている状態とチップ表面が欠け画像に白く映っているものの 2 種類がある。両モードとも様々な大きさの程度があり、画像内の面積の 20% を占めるものもあれば、1% ほどにしかならないものもある。良品内には欠けは一切存在しないが、軽微な異物は存在する。

訓練データは 196,242 枚の良品画像のみからなり、テストデータは良品 10,000 枚と不良品 204 枚から構成される。

5.1.2 bottle

bottle は MVTEC Anomaly Detection¹³⁾ のカテゴリの 1 つである。元々は 900×900 の画像サイズのカラー画像であり、非常に高解像度なデータセットである。本実験では LED チップ同様 64×64 にリサイズしている。ただチャンネル数は 3 でカラー画像のままである。

訓練データは 209 枚の良品画像のみであり、テストデータは良

Fig. 5 Good and defective LED chip images. There are defective modes such as chips, stains, and contamination in defective images. The bounding boxes in (b) indicate the defective part

Fig. 6 Model diagram of the Discriminator. The structure of the discriminator is adjusted so that each pixel of the 8×8 , 4×4 , and 2×2 MAPs has an appropriate receptive field. Conv in the figure indicates Convolution Layer, BN indicates Batch Normalization Layer, and Avg Pool indicates Average Pooling

Table 1 Experimental results. The AUROC, AUPRC and processing time for each method are shown. The processing time represents the computation time of the neural network only, and does not include the time for loading the images and so on. The machine used to measure the processing time is equipped with NVIDIA TITAN V.

		RGAN	Autoencoder + CGAN	CGAN	AnoGAN	Ours $\epsilon = 0.1$	Ours $\epsilon = 0.2$	Ours $\epsilon = 0.3$	Ours $\epsilon = 0.4$	Ours $\epsilon = 0.5$
LED チップ	AUROC	0.879	0.538	0.747	0.828	0.956	0.954	0.933	0.966	0.972
	AUPRC	0.567	0.020	0.191	0.552	0.631	0.664	0.720	0.805	0.691
bottle	AUROC	0.845	0.640	0.677	0.860	0.910	0.866	0.844	0.891	0.887
	AUPRC	0.953	0.830	0.886	0.955	0.973	0.958	0.953	0.965	0.964
processing time sec		0.024	0.014	0.024	3.419			0.041

Fig. 7 Generated image of an LED chip by MSP-CGAN

Fig. 8 Anomaly Map by Discriminator in MSP-CGAN

品 20 枚と不良品 63 枚から構成される。不良品は broken_large, broken_small, contamination の 3 カテゴリに分けられている。

5.2 モデル

本実験では、MSP-CGAN に Generator と Discriminator が存在する。Generator は Transposed Convolution (stride 2) + Batch Normalization¹³⁾ + Leaky ReLU を 1 つの Upsampling Block とし、その Block を複数層積み重ねることで生成を行う。

Discriminator は Convolution (stride 2) + Batch Normalization¹³⁾ + Leaky ReLU を 1 つの Downsampling Block とし、その Block を複数層積み重ねることで判別を行う。また、学習の安定化のために Generator の各 Transposed Convolution 層、Discriminator の各 Convolution 層には Spectral Normalization¹⁴⁾が適用され、Generator と Discriminator の中間層に Non Local Operation^{15), 16)}が適用されている。

MSP-CGAN の Discriminator は複数解像度の Discrimination Map が出力されるが、本実験では 8×8 , 4×4 , 2×2 の 3 つの Map が出力されるよう Downsampling Block の数を調整した。また、 8×8 , 4×4 , 2×2 の各 MAP の各画素が適切な受容野となるよう Discriminator の構造を調整した。LED チップ、bottle 共に同じ構造のモデルを用いている。Discriminator のモデルを図 6 に示す。LED では学習回数を 15 epoch、bottle では学習回数 3,000 epoch とし、両実験で batch size は 64、ハイパーパラメータである ϵ は 0.1 から 0.5 ままで 0.1 刻みで変動させ実験した。また最適化手法には Adam¹⁷⁾を用い、Generator は $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ とし Discriminator は $\eta = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ とした。

5.3 比較手法

本実験では、同画像データセットで学習した通常の GAN の Discriminator (RGAN)、画像を Autoencoder の Encoder で圧縮して得られる低次元特徴に対して CGAN を適用したモデル (Autoencoder+CGAN)、画像に対して Multi-scale Patch を用いない CGAN を適用したモデル (CGAN)、AnoGAN と比較を行う。なお、本実験の提案手法、CGAN、では RGAN で学習した Discriminator を p_{data} とした。AnoGAN も同様に RGAN で学習することで得られた Generator と Discriminator を用いて異常度を算出した。

5.4 結果

実験の結果を表 1 に示す。本実験では ROC 曲線下の面積である AUROC、Recall Precision 曲線下の面積である AUPRC を評価指標とした。全実験において AUROC, AUPRC はそれぞれ算出される異常度に対して、しきい値を変動させることで得られた曲線下の面積を用いている。

表 1 の通り、提案手法は他の手法と比較して高い精度で異常の検出が行えていることが確認できる。これは提案手法が情報の圧縮無しに画像から直接多重解像度な Discrimination Map を出力することで、小さな異常部の情報が消えることなく、様々なスケールの異常に対して頑健になったためだと考えられる。

LED チップの実験における MSP-CGAN の Generator による生成画像と Discriminator による Discrimination Map を図 7, 図 8 に示す。生成画像は正常の LED チップには近いが、正常には存在しないような画像となっていることが確認できる。異常度 Map の欠陥

(a) Overlooked images by our method

(b) Images overdetected by AnoGAN but detected by our method

Fig. 9 Overlooked images by our method and AnoGAN

箇所に対する Localization は正確ではないものの、概ね異常を検出できていることが確認できる。しかし、図 8 の 3 行目の $D_1(x)$, $D_2(x)$ では欠陥が存在しない領域も異常度が高くなっている。これは本実験ではそれぞれ Map のピクセルが適切な受容野となるように Convolution 層、Pooling 層を構成したが、Batch Normalization 層や Non Local 層は Feature Map の全ピクセルを考慮して計算を行うためであると考えられる。

提案手法により見逃しとなってしまうデータの考察を行う。見逃しとは提案手法の出力値である異常度が低いサンプルである。異常度が最も低かった不良品の画像 5 つを図 9(a) に示す。どれも軽微な異物であり検出の困難なデータであると推測できる。また、これらのサンプルよりも軽微な異物があるサンプルは正常データとして訓練データに存在するため見逃してしまったと考えられる。AnoGAN でも同様にこれらのサンプルは非常に低い異常度を出力し、見逃していた。しかし、AnoGAN での見逃していた図 9(b) のような画像を提案手法は異常であると判定していた。AnoGAN は入力と Generator の出力の差分の平均を異常度としているため、図 9(b) のような面積が小さく濃淡値が正常画素と大きな違い異常について検出が困難である。提案手法は差分を使わず表現力豊かな Discriminator により局所的な特徴を用い判定を行うため図 9(b) のようなサンプルが検出できたと考えられる。

6. 結 論

本稿では、異常画像検出の枠組みとして、Multi-scale Patch Complementary GAN を提案した。この手法は Complementary GAN の構造に Multi-scale Patch の考えを導入し、多重解像度戦略を取ることで様々な大きさの欠陥に対して頑健な検出を可能とした。また、Complementary GAN のように情報を圧縮してから学習を行うのではなく、画像を直接入力とすることで、異常情報の欠落を防いだ。LED チップ、はんだ画像データセットに対して実験を行い、従来手法よりも高い精度で異常を検出できることを確認した。し

かし、実利用する上では検出精度、Localization の精度共に十分であるとはいえない。そのため今後の課題としては、実用化に向けたさらなる精度の向上と正確な欠陥箇所の検出が挙げられる。

参 考 文 献

- 1) P. Zheng, S. Yuan, X. Wu, J. Li and A. Lu: One-Class Adversarial Nets for Fraud Detection, arXiv preprint arXiv:1803.01798, (2018).
- 2) S. Hochreiter and J. Schmidhuber: Long Short-Term Memory, Neural Comput., 9, 8, (1997), 1735.
- 3) B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho and H. Chen: Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection, International Conference on Learning Representations, (2018). <https://openreview.net/forum?id=BJJLHbb0-> accessed 2020.05.04.
- 4) I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio: Generative Adversarial Nets, in Advances in neural information processing systems, (2014), 2672.
- 5) T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth and G. Langs: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in International Conference on Information Processing in Medical Imaging, (2017), 146.
- 6) L. Deecker, R. Vandermeulen, L. Ruff, S. Mandt and M. Kloft: Anomaly Detection with Generative Adversarial Networks, (2018). <https://openreview.net/forum?id=S1EfylZ0Z> accessed 2019.01.31.
- 7) J. Johnson, A. Alahi and L. Fei-Fei: Perceptual losses for real-time style transfer and super-resolution, in European Conference on Computer Vision, (2016), 694.
- 8) M. Kimura and T. Yanagihara: Anomaly Detection Using GANs for Visual Inspection in Noisy Training Data, arXiv preprint arXiv:1807.01136, (2018).
- 9) J. Zhao, M. Mathieu and Y. LeCun: Energy-based Generative Adversarial Network, arXiv preprint arXiv:1609.03126, (2016).
- 10) T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen: Improved Techniques for Training GANs, arXiv preprint arXiv:1606.03498, (2016).
- 11) P. Isola, J. Y. Zhu, T. Zhou and A. A. Efros: Image-to-image translation with conditional adversarial networks, in Proceedings of the IEEE conference on computer vision and pattern recognition, (2017), 1125.
- 12) P. Bergmann, M. Fauser, D. Sattlegger and C. Steger: MVTec AD -- A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2019).
- 13) S. Ioffe and C. Szegedy: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, arXiv preprint arXiv:1502.03167, (2015).
- 14) T. Miyato, T. Kataoka, M. Koyama and Y. Yoshida: Spectral normalization for generative adversarial networks, arXiv preprint arXiv:1802.05957, (2018).
- 15) X. Wang, R. Girshick, A. Gupta and K. He: Non-local Neural Networks, arXiv preprint arXiv:1711.07971, (2017).
- 16) H. Zhang, I. Goodfellow, D. Metaxas and A. Odena: Self-Attention Generative Adversarial Networks, arXiv preprint arXiv:1805.08318, (2018).
- 17) D. P. Kingma and J. Ba: Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, (2014).