

Neural Landmark Transfer による機械部品の打痕画像生成手法

山口 裕希[†] 青野 雅樹^{††}

[†] 豊橋技術科学大学 情報・知能工学課程 〒441-8580 愛知県豊橋市天伯町字雲雀ヶ丘 1-1

^{††} 豊橋技術科学大学 情報・知能工学系 〒441-8580 愛知県豊橋市天伯町字雲雀ヶ丘 1-1

E-mail: [†]yamaguchi@kde.cs.tut.ac.jp, ^{††}aoono@tut.jp

あらまし 我々は、部品メーカーとの共同研究の中で、打痕などのキズが入った不良部品を正確に認識する技術を模索している。前提として、車のギア部品は品質検査行程で固定カメラで一定の証明条件下で部品の外観を撮影され、その画像はメーカーから与えられるとする。また、撮影された画像のうち、不良箇所（例、打痕）を含む割合は非常に少ない。この条件下で、深層学習を用いて正確に良品か不良品かを判定しなくてはならない。本研究では、少量の学習用の画像（打痕画像）から、打痕の発生する場所、大きさ、向き、形などのバリエーションを出せるようなデータ拡張技術を目指す。実際の打痕を含む画像例が少ないため、我々は、1枚の画像からでも、スタイルを模写できる NeuralStyle Transfer (NST) 技術に着目した。NST ではコンテンツ画像とスタイル画像から、スタイル画像の画風を反映した新たなコンテンツ画像が生成できるのに対して、我々は、スタイル画像の代わりに独自に考案した「ランドマーク画像」とそれに伴う損失関数を導入し、人工的であるが、撮影画像にはなかった打痕のリアルなパターン生成 (NeuralLandmark Transfer) を試みた。評価手法として、FID (Fréchet Inception Distance) を導入し、実験したところ、ベースラインよりも高精度となることを確認した。

キーワード 画像, 生成, 深層学習, 社会応用

1 はじめに

深層学習による画像認識は工業分野への導入が進んでおり、部品の不良判定などに応用されている。これらの深層学習による画像認識モデルの学習には、一般的に大量のデータセットが必要となる。一方で、不良部品の発生確率は少なく、十分な量のデータセットを用意することは困難である。我々は1枚単位での画像の生成手法として Neural Style Transfer (NST) [1] に着目して研究を行った。本研究では、NST をベースとしてより打痕画像生成に適した1枚単位での画像生成手法を提案し、不良部品の判別精度向上を目指す。

以降、次節で関連研究について述べ、3節でデータセット、4節で提案手法、5節で実験とその結果について報告する。6節ではまとめと今後の課題について述べる。

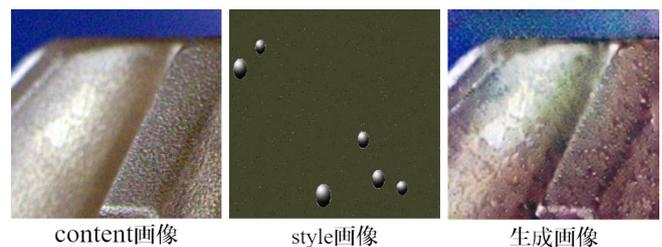
2 関連研究

画像生成モデルとしては、解像度と多様性において GAN [2] と VAE [3] が特に優れている。近年、これらの手法は 1024×1024 程度の高解像度な画像の生成に成功しているが [4] [5]、画像の生成に教師となる画像データセットが必要であるため、発生頻度の少ない打痕画像を生成することには不適である。1枚単位での画像の生成が可能な手法として Neural Style Transfer (NST) [1] が挙げられる。NST はコンテンツ画像とスタイル画像を用意し、コンテンツ画像へスタイル画像の画風を転移させる手法である。一般的に絵画などに用いられる手法であるが、発展研究で写実的な画像に対する手法が提案されている [6]。NST 手法を打痕画像生成に適用する際の問題として、生成し

たい打痕形状を与えるスタイルが不明であること、生成する打痕の位置を制御できないことが挙げられる。



(a) NST による画風の変換。スタイルを画像特徴のグラム行列と定義し、出力画像とスタイル画像のスタイルの二乗誤差と出力画像とコンテンツ画像の二乗誤差を最小化する。



(b) NST を打痕画像に適用した例。NST の損失関数では、打痕の形状と位置を制御することができない。

図1 Neural Style Transfer (NST) について

3 ベベルギア画像データセット

3.1 全体画像

ベベルギアとは、差動歯車装置などで用いられる傘上の歯車である。本実験に使用したデータセットでは、ベベルギアを横から撮影したデータを使用する。画像サイズは縦 $370 \times$ 横

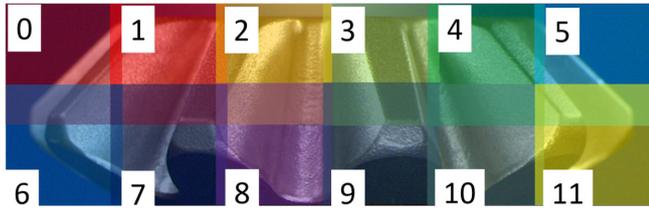
1149 となっており、打痕ありクラスと打痕なしクラスがそれぞれ 1000 枚の画像を持つ。



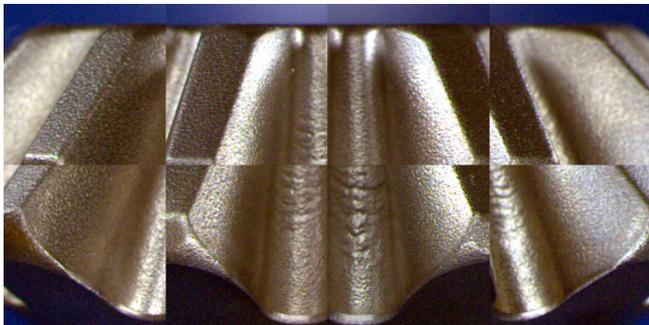
図 2 ベベルギア全体画像の例

3.2 分割画像

今回の実験では、共同研究先からの要件で、図 3(a) のように 12 分割された部分画像が与えられた。ただし、左右の両端には背景が多いこと、ならびに、打痕自体も一部しか撮影されない場合があるため、これらの部分 (0,5,6,11 番) は対象外とし、データとしても与えられていない。画像のサイズは縦 224 × 横 224 の正方形として与えられた。各位置における打痕ありクラスと打痕なしクラスの画像枚数は表 1 に示すとおりである。



(a) ベベルギア全体画像を 12 分割した 224 × 224 画像がデータとして与えられる。ただし、左右の両端画像は対象外。



(b) (a) の方針で作成された分割画像例。左上から 1-4, 7-10 番の画像。

図 3 ベベルギア分割画像の例

表 1 分割画像の各位置での枚数

図 3(a) の位置	正常画像数	打痕画像数
1	610	614
2	634	633
3	487	488
4	285	287
7	279	282
8	280	284
9	280	284
10	223	226

3.3 打痕の例

図 4 にデータセットの打痕の例を示した。図に示すように、打痕と称される傷には、ぶつかった痕のようなものや線状のもの、その他の珍しいものも存在する。これらの傷がベベルギアの谷以外の部分に存在するとき、ベベルギアは不良と判定される。ベベルギアの谷部分に見られる傷は製品上問題はなく、打痕とは見なされない。

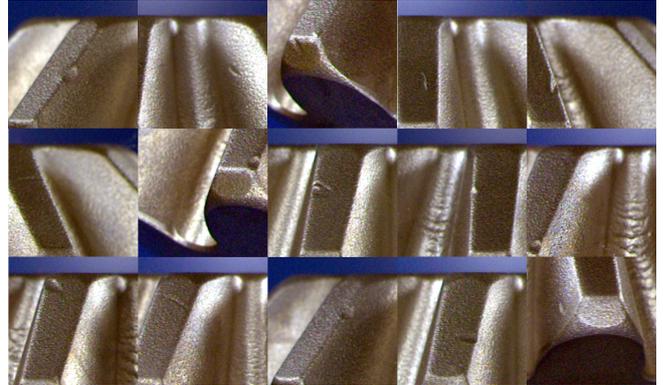


図 4 打痕の例

3.4 打痕位置・形状の指定について

ベベルギア画像データセットの打痕画像から打痕部分の長方形領域を切り出し、打痕以外の画素を黒色で上書きした後にグレースケール変換した打痕形状画像セットを用意した。図 5 に打痕形状画像の例を示した。

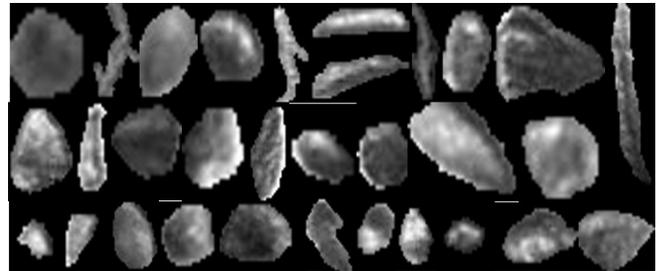


図 5 打痕形状画像セットの例

4 提案手法

我々は、NST を用いた画像生成における打痕形状と打痕位置の制御の問題に対応する手法として、NST 手法をベースとした Neural Landmark Transfer (NLT) を提案する。また、NLT の拡張提案として、学習済みの打痕分類器を用いてより本物に近い画像を生成するための損失項を定義する。

4.1 提案手法 1: Neural Landmark Transfer (NLT)

打痕なし画像 $x_{content}$ と打痕なし画像に手で打痕形状を移したランドマーク画像 $x_{landmark}$ の 2 つを用意する。ランドマーク画像に用いる打痕形状は、本物の打痕画像から打痕部分を抽出したものである。ランドマーク画像は、抽出した打痕

形状を打痕なし画像の任意の位置に設置することで、打痕なし画像を所望の位置及び形状の打痕あり画像へ転移させるための目印としての役割を持つ。出力画像 \mathbf{x}_{output} の初期値をランドマーク画像で与える。それぞれの画像は、縦 224 × 横 224 で RGB の 3 チャンネルを持つ (1, 224, 224, 3) のシェイプの 4 次元テンソルである。これら 3 つの画像に対して以下の損失関数を定義する。

$$L_{content,i} = \frac{1}{2}(F_i(\mathbf{x}_{content}) - F_i(\mathbf{x}_{output}))^2 \quad (1)$$

$$L_{landmark} = \frac{1}{2}(L(\mathbf{x}_{landmark}) - L(\mathbf{x}_{output}))^2 \quad (2)$$

ここで F_i は ImageNet [7] で事前学習済みの VGG19 モデル [8] の i 番目の ConvLayer の出力であり、 L は Laplacian Filter の出力である。学習では $L_{content}$ と $L_{landmark}$ の和を損失関数として出力画像 \mathbf{x}_{output} を変化させる。これらの損失関数を最小化することは、ランドマーク画像の輪郭を保つ制約のもとで、ランドマーク画像を打痕なし画像の特徴に近づける操作に等しい。式 (1) と式 (2) に定数項を乗算し加算した式 (3) を提案手法の損失関数として定義する。

$$L_{loss} = \alpha \sum_{i=1}^n L_{content,i} + \beta L_{landmark} \quad (3)$$

本提案の損失関数のイメージ図を図 6 に表示する。

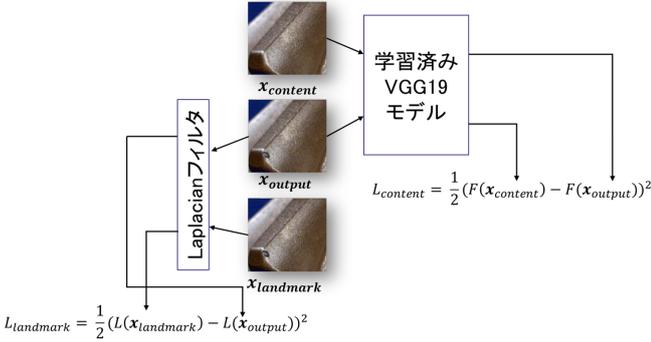


図 6 NLT の損失関数のイメージ図

4.2 提案手法 2: 打痕判定損失項

まず 3.2 節の画像を分割して 224 × 224 のサイズにクロップされたベベルギア画像データセットに対して、打痕分類タスクを学習した。ImageNet 学習済みの ResNet50 [9] をモデルとしてベベルギア分割画像データセットに対して転移学習を行うことで、224 × 224 のサイズのベベルギアの画像に対して打痕を分類できる打痕分類モデルを用意した。提案手法 2 として、提案手法 1 の損失関数に以下で定義する打痕判定損失項 $L_{detector}$ を加えた。

$$L_{detector} = \frac{1}{2}(1 - cnn(\mathbf{x}_{output}))^2 \quad (4)$$

ここで、 cnn は、上記で示した打痕の有無を判別する打痕分

類モデルである。この分類器は打痕なし画像に対して 0 を出力し、打痕画像に対して 1 を出力する。式 (4) の打痕判定損失項を最小化することにより、打痕分類器に打痕と判断させる方向への勾配降下が行われる。これにより、生成されたランドマーク画像を転移した出力画像の陰影や形状を、より打痕と認識されるように調整する操作が行われる。

本提案の損失関数のイメージ図を図 7 に表示する。

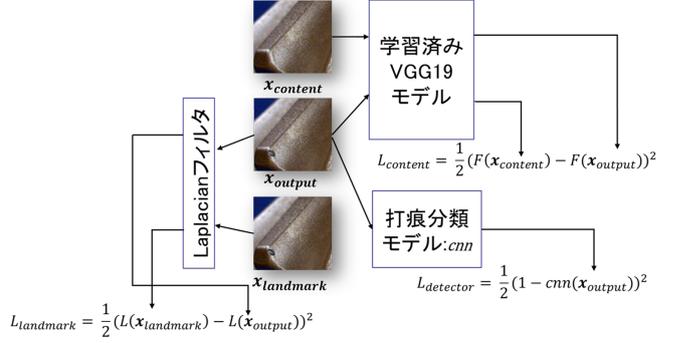


図 7 提案手法 2 の損失関数のイメージ図

5 実 験

本節では、本研究で実施した実験について説明する。我々はシミュレーションにより擬似的なランドマーク画像を 1000 枚作成し、これらの画像について提案手法を適用して画像を生成した。擬似的なランドマーク画像をベースラインとして、本物の打痕画像データセットと生成された画像のデータセットの距離を Fréchet Inception Distance (FID) スコア [10] で測定し、ベースラインとの FID スコアと比較した。

5.1 実験環境

本実験で使用した環境について以下に示す。CPU は Intel(R) Xeon(R) CPU @ 2.30GHz プロセッサ、メモリ 12.9GB を使用した。GPU は Tesla K80、メモリ 12GB を使用した。深層学習フレームワークとして Pytorch1.3.1 を利用した。入力は 3.1 節のベベルギア全体の画像を 224 × 224 にクロップしたものを使用し、生成された 224 × 224 の画像をクロップ前の位置に埋め込んだ画像を出力とした。提案手法 1 と提案手法 2 において、共通して ImageNet 学習済み VGG19 モデルの最初の ConvLayer3 層を $L_{content,i}$ として利用した。出力画像 \mathbf{x}_{output} の初期値をランドマーク画像 $\mathbf{x}_{landmark}$ として、最適化手法は共通して L-BFGS [11] で学習した。各 Loss 項 $L_{content}$, $L_{landmark}$, $L_{detector}$ の係数 α, β, γ は、提案手法 1 では $L_{content}$, $L_{landmark}$ の 2 つに共通して $\alpha = \beta = N(2, 0.2)$ のガウス分布の乱数で与えた。提案手法 2 では $L_{content}$ に対して $\alpha = N(10^6, 0.08)$, $L_{landmark}$ に対して $\beta = N(10^5, 0.1)$, $L_{detector}$ に対して $\gamma = N(10^4, 0.05)$ のガウス分布の乱数で与えた。FID スコアの測定は Keras フレームワークを用いた別プログラムで実装した。サンプル数は本物の打痕画像の総数である 1000 枚とし、ベベルギア全外画像を 299 × 299 にリサイズして FID スコアを測定した。

5.2 擬似的なランドマーク画像の生成について

提案手法では、画像の生成に手で打痕形状を転移させたランドマーク画像を必要とする。FID スコアの測定のためには、本物の打痕画像 1000 枚と同数の生成画像が必要だが、1000 枚のランドマーク画像を用意することは困難である。そこで本実験では、提案手法のアプリケーションとして、マスク処理と同様乱数を用いて擬似的なランドマーク画像を生成し、それを用いて打痕画像との FID スコアを測定した。

図 8 のフローチャートに示す主な処理を以下に列挙する。

元画像 擬似的なランドマーク画像生成の入力として、3.1 節の打痕のないベベルギア全体の画像データセットから一様乱数で一枚を選択して与える。この入力画像は縦 370 × 横 1149 のサイズを持つ。

中心をクロップ 入力となる元画像の中心をクロップし、 $x = 100, y = 45$ の座標から縦 285 × 横 950 の長方形領域を切り出す。

マスク作成 このクロップされた画像に対して、後述のマスク処理を適用することで、打痕の存在しうる位置を指定するマスク画像を生成する。

224 × 224 クロップ このマスク画像の白色領域の座標の中から、上下左右に $224/2 = 112$ ピクセルの画素が存在する座標を一様乱数により決定し、縦 224 × 横 224 の領域を指定する。

打痕形状画像セット 3.4 節で述べた、グレースケール変換した本物の打痕画像から打痕形状を切り取った画像データセットである。

打痕形状選択 打痕形状画像セットの中から一様乱数により 1 枚の打痕形状画像を選択する。

打痕付与 選択された打痕形状画像について、 $h_d =$ 打痕形状画像の高さ/2、 $w_d =$ 打痕形状画像の幅/2 と定義する。224 × 224 にクロップされたマスク画像の白色領域の座標の中から、上下に h_d 、左右に w_d の画素が存在する座標を一様乱数により決定し、その座標を中心に打痕形状画像の画素値=0 でない画素を 224 × 224 にクロップされた画像に上書きする。

カラー復元 打痕を付与した画像を、後述するカラー復元によって打痕部分の色をグレースケールから打痕付与前の画像の色に戻す。得られた 224 × 224 の画像を擬似的なランドマーク画像として提案手法の入力に用いる。

元画像に埋め込み 擬似的なランドマーク画像を元画像のクロップした位置に埋め込むことで、打痕形状を付与したベベルギア全体の画像を生成する。この画像を後述するベースラインとし、本物のベベルギア打痕画像との FID を測定する。

5.2.1 マスク画像作成について

マスク画像作成の目的は、擬似的なランドマーク画像の生成の際、ベベルギア画像の画素の中で打痕の存在しうる位置に打痕を生成するためである。3.3 節で述べたように、打痕はベベルギアの谷以外の部分に現れる。特に図 4 に見られるように、打痕はベベルギアの山部分のエッジ上に出現することが多い。こ

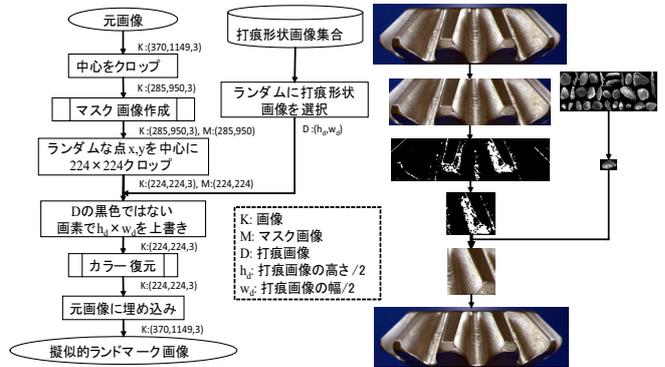


図 8 擬似的なランドマーク画像生成のフローチャート

の特徴に基づいて、1 次微分フィルタの輪郭抽出を利用することで打痕の存在しうる範囲を指定するマスク画像の生成を行う。

マスク画像の生成は、OpenCV ライブラリ [12] による図 9 の処理で行う。入力画像は縦 285 × 横 950 にクロップされたベベルギア全体の画像となる。最初に画像に対してグレースケール変換を行い、微分フィルタにより画像の輪郭が得られる。輪郭画像に対してメディアンフィルタを利用してノイズを減らす処理を行う。この画像に対して、大津の 2 値化 [13] を適用して 2 値のマスク画像を生成する。大津の 2 値化で生成される 2 値画像はノイズが激しい為、収縮とメディアンフィルタを 2 回ずつ適用することにより、ノイズを削除する。

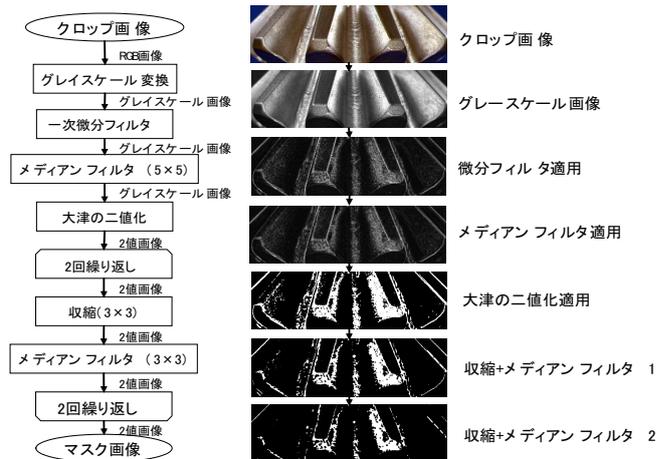
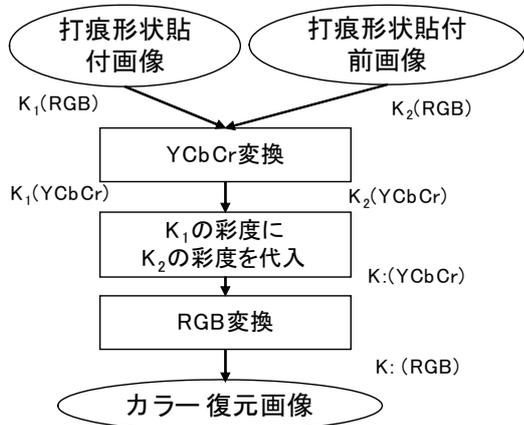


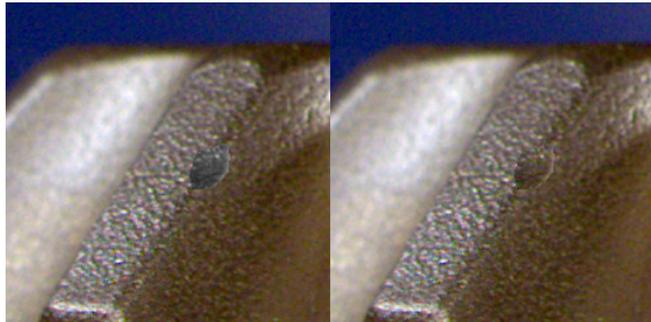
図 9 マスク画像の生成の流れ

5.2.2 カラー復元について

打痕形状画像はグレースケールの画像であるため、カラー復元前のランドマーク画像は打痕の領域が元の画像の色空間と異なる。これを元の画像の色空間に戻す処理として、カラー復元関数を用意する。図 10(a) にフローチャートを示した。関数は、入力として打痕形状を上書きされた画像と上書きされる前の画像の 2 つを持つ。これら 2 つの画像を RGB から YCbCr [14] 空間に変換し、上書き後の画像の CbCr チャンネルを元の画像の CbCr チャンネルで上書きして RGB 空間へ戻す処理を行う。これにより上書き後の画像の彩度を上書き前の画像の彩度に戻すことができる。図 10(b) にカラー復元前後のランドマーク画像を示した。



(a) カラー復元のフローチャート



(b) 復元前と復元後の画像

図 10 カラー復元処理の流れと復元前後の画像

5.3 ベースライン画像

実験において提案手法と比較するベースライン画像として、5.2 節で生成される擬似的なランドマーク画像を使用する。この擬似的なランドマーク画像は、打痕の存在する位置に本物の打痕画像から抽出した打痕形状画像を上書きした画像となる。提案手法の入力として、ベースライン画像と同様の処理で生成される擬似的なランドマーク画像を用いるため、本実験はベースライン画像となるランドマーク画像とランドマーク画像に対して提案手法を適用した結果との差の測定実験とみなせる。

5.4 Fréchet Inception Distance (FID) スコア

FID スコアは、GAN や VAE などの画像生成モデルに用いられる評価尺度である。FID スコアは、以下に定義される式で 2 つの画像データセット間の距離を表現する。

$$d^2 = \|\mathbf{m}_1 - \mathbf{m}_2\|_2^2 + \text{Tr}(\mathbf{C}_1 + \mathbf{C}_2 - 2(\mathbf{C}_1\mathbf{C}_2)^{1/2}) \quad (5)$$

ここで \mathbf{m}_1 は、画像データセット 1 を ImageNet 学習済みの InceptionV3 [15] モデルに入力した、モデルの最後の平均プーリング層の出力である 2048 次元ベクトルの平均である。 \mathbf{m}_2 も同様に、画像データセット 2 に対する Inception モデル出力の平均である。 Tr は行列のトレースである。 \mathbf{C}_1 は、画像データセット 1 を入力とした Inception モデル出力 2048 次元ベクトルに対する共分散行列であり、 \mathbf{C}_2 も同様に画像データセット 2 に対する共分散行列である。式 (5) のように、FID スコアは 2 つの画像データセットに対して、Inception モデルに画像

を入力することで得られる特徴表現の平均と分散の距離を評価する尺度である。

FID スコアの評価には、GAN などの生成モデルなどの例では、多くの場合 10000 枚を超える画像が使用される。これは、画像の平均と分散のばらつきを小さく抑えるためである。FID スコアに信頼性を持たせるために、本実験では 1000 枚の画像に対して打痕画像と生成画像との FID スコアを測定する。

5.5 実験結果

以下に実験結果を述べる。

5.5.1 FID スコアの比較

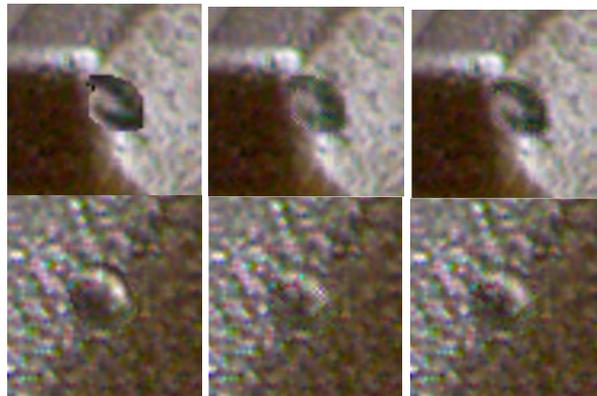
表 2 に計算した FID スコアを示す。3.1 節に示したベベルギア全体の打痕画像 1000 枚に対して、各手法で生成した画像 1000 枚との FID スコアを計算した。本物の打痕画像との FID スコアは、提案手法 2 である NLT に打痕判定損失項を加えたものが最も良い結果となった。

表 2 FID Score

	FID Score
ベースライン画像	16.31
Neural Landmark Transfer (NLT)	16.14
NLT+打痕判定損失項	15.90

5.5.2 生成画像の比較

図 11 に各手法で生成された画像を示した。ベースライン画像では打痕の輪郭が強く残っているが、NLT では輪郭が元の画像に合うように調整されていることが分かる。また、NLT に打痕判定損失項を加えたものは、打痕の陰影や形状がより本物に近いように変化していることが確認できる。



(a) ベースライン画像 (b) NLT (c) NLT+ $L_{detector}$

図 11 生成された画像の比較

6 まとめと今後の課題

本研究では小規模データセットへの画像生成手法として、NST 手法を応用した手描きの画像を用いる Neural Landmark Transfer 手法を提案した。NLT 手法は手描きの画像を本物らしい画像に近づけることができる。実験の結果、ベースライン

画像より2つの提案手法の方が本物の打痕画像データセットとのFIDスコアが近いことが示された。生成された画像を比較すると、NLT手法はベースライン画像よりも自然な画像を生成することができた。特に提案手法2のNLTに打痕判定損失項を加えた手法では、打痕の陰影や形状のパターンが本物らしく見えるように変化することが確認された。今後の課題としては、実際に手動でランドマーク画像を1000枚用意してFIDスコアを確かめることや、提案手法2で用いた打痕分類器の学習に使用していないパターンのランドマーク画像によるNLT画像生成の実験などが挙げられる。

謝 辞

本研究の一部は、武蔵精密工業株式会社の支援を受けて遂行した。

文 献

- [1] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, June 2016.
- [2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14*, pages 2672–2680, Cambridge, MA, USA, 2014. MIT Press.
- [3] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.
- [4] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.
- [5] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. *CoRR*, abs:1906.00446, 2019.
- [6] Shaohua Li, Xinxing Xu, Liqiang Nie, and Tat-Seng Chua. Laplacian-steered neural style transfer. In *Proceedings of the 25th ACM International Conference on Multimedia, MM '17*, pages 1716–1724, New York, NY, USA, 2017. ACM.
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE Computer Society, 2009.
- [8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, pages 6629–6640, USA, 2017. Curran Associates Inc.
- [11] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Math. Program.*,

45(1-3):503–528, August 1989.

- [12] G. Bradski. The opencv library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [13] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.
- [14] Richard Szeliski. *Computer vision algorithms and applications*. Springer, London; New York, 2011.
- [15] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826, 2016.