

GAN を用いた画像の輪郭線に着目した画像修復

皆藤 優太[†] 田村 仁[‡]日本工業大学 機械システム工学専攻[†] 日本工業大学 先進工学部 ロボティクス学科[‡]

1. はじめに

画像修復は、長年盛んに研究が行われている。画像修復とは、画像内に入り込んだノイズや画像の欠損箇所を修復する技術の一つである。使用用途は、画像の劣化による傷の修復や、写真の赤目除去、ストリーミングビデオにおける符号化及び伝送過程で失われた部分の置き換え、動画のロゴの削除に用いられたりするなど、多岐に渡る。本稿では、画像の欠損箇所の修復を扱う。

近年では、機械学習を用いた画像修復手法が主流となっている。それらは目を見張るような成果を挙げている一方で、それらの手法は高周波成分(輪郭線)の生成に問題がある。

これらの手法では本質的に損失関数として訓練画像(生成器への入力画像)と正解画像(訓練画像から生成器が出力する画像の目標画像)との誤差を用いている。損失関数は画素ピクセルの平均誤差を最小化するように設定されているため、輪郭線がぼやけた画像を出力するように生成器の学習が進む傾向が見られる。

そこで、輪郭線を先に再現をした上で着色することでこの問題の解決を図る。

本稿の目的は、輪郭線を修復するにあたり損失関数の種類や識別器の学習方法の違いがどの程度輪郭線の修復に影響を検討し、輪郭線修復手法を提案することである。

2. 関連研究

近年、高周波成分の修復を強化するために、輪郭線に着目した画像修復手法が提案され始めている。多くの輪郭線に着目した手法は輪郭線を修復したのちに着色を行う。

輪郭線を用いた画像修復術として、[1]や[2]がある。

[1]では、Holistically-nested edge detection (HED) [3]法を用いてカラー画像をまず線画化し、その後細線化(線を1ピクセル幅にするこ

と。)を施す。線画化された画像内の欠損箇所を、畳み込み層で構成されたニューラルネットワーク(encoder-decoder)にて欠損箇所の輪郭線を修復する。修復済みの線画画像に対して [4]の手法をもとにしたフレームワークによって着色を行う。輪郭線の修復と着色を一組とする画像修復手法である。

[2]ではCanny法[5]にてカラー画像から輪郭線を抽出した画像を輪郭線修復モデルの学習に用いている。線画として修復してから着色を行う流れは[1]とあまり大きな違いはない。

3. 提案手法

深層学習を用いた画像修復フレームワークの多くは、画像内の欠損領域の輪郭線や色などを同時に修復するように学習されている。

輪郭線に着目した画像修復フレームワークでは、入力画像を線画化し、着色より先に画像内の欠損領域の輪郭線を修復し、その後着色する。

そのため、フレームワーク内に2種類の修復モデルを用意する。1つは輪郭線修復を行うモデル、もう1つは着色を行うモデルである。2種類のモデルを配置することにより、輪郭線に着目した画像修復を実現する。

先行研究の[6]ではpix2pix[7]を用いて実験を行った。本稿では、[6]で用いたpix2pixのDiscriminatorへのInputを変更した修復器と、WGAN[8]を参考にDiscriminatorへのInputを変更した修復器の損失関数をWasserstein距離に置き換えた修復器のこれら2つの輪郭線専用の修復器を作成した。

pix2pixにおけるDiscriminatorへのInputの変更について説明する。変更前はGeneratorの出力をそのままDiscriminatorへのInputとしていたが、変更後はGeneratorの出力画像を一度2値化した画像をDiscriminatorへのInputとした。2値化を施す際の輝度の閾値は100とした。2値化処理を修復器の学習フィードバックに加えたのは、線の出力の有無をはっきりさせることではっきりとした線を出力するようにGeneratorを仕向けることが目的である。

損失関数をWasserstein距離に置き換えたのは、最適値付近で勾配が消失しない性質が輪郭線修

Image Inpainting Focusing on Edge of Images Using GAN
[†] Yuta Kaito, Nippon Institute of Technology Department of Mechanical Systems Engineering
[‡] Hitoshi Tamura, Nippon Institute of Technology Department of Advanced Engineering Faculty of Robotics

復にどのように影響するか検討するためである。

4. 実験

Discriminator への Input を 2 値化する修復器と、その修復器の損失関数を Wasserstein 距離に置き換えたそれぞれの修復器は、ImageNet [9] からランダムに選んだ約 10 万枚の画像を [6] の結果を受けて HED 法で線画抽出し、2 値化・4 値化を施したデータセットを用いて学習させた。どの修復器もエポック数は 100、バッチサイズは 32 とした。

学習完了後、定量評価を行った。評価指標として PSNR 値と SSIM 値を用いた。これらは値が大きいほどよい。また、今回作成した修復器は輪郭線専用である。

5. 結果

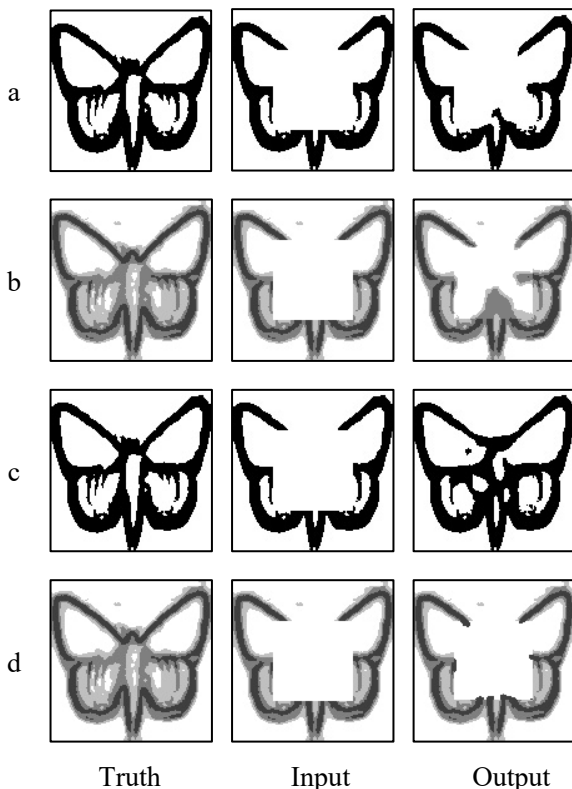


図 1 各修復器の出力画像

図 1 に各修復器に画像を入力した際の出力画像を示す。図 1 中の a は 2 値化した線画像データセットで学習を施した pix2pix, b は 4 値化した線画像データセットで学習を施した pix2pix, c は 2 値化した線画像データセットで学習を施した Wasserstein 距離修復器, d は 4 値化した線画像データセットで学習を施した Wasserstein 距離修復器に画像を与えた際の出力画像である。

定量評価の結果を 表 1 にまとめた。入力画像は正解画像内に 64px 四方のマスクをその中央に載せた画像である。

表 1 定量評価の結果

	PSNR[dB]	SSIM
2 値化・2 値化 F/B pix2pix	11.04	0.8241
4 値化・2 値化 F/B pix2pix	15.99	0.8092
2 値化・Wasserstein 距離	12.13	0.8253
4 値化・Wasserstein 距離	14.64	0.7940

6. 考察

図 1 に示されている出力画像において入力画像のマスクの領域をうまく修復できずにいる修復器があるのは、学習回数不足によるものであると考えられ、回数を増やすことでより性能が向上すると考えられる。

表 1 にて対象画像が 2 値化画像か 4 値化画像によって PSNR 値に大きな差があるのは値の算出方法に起因すると考えられる。画素階調が絞られる分だけ正解画像と出力画像の対応するピクセル間の差の影響が大きくなり、そのため 2 値化画像の PSNR 値が 4 値化画像で測定した場合よりも小さくなると考えられる。

表 1 の定量評価の結果だけを見れば 2 値化 F/B pix2pix の方が良いが、図 1 の c の出力画像を見ると他と同じ学習回数に関わらず線の接続性も SSIM 値も良い。よって、損失関数を Wasserstein 距離とした輪郭線修復器を輪郭線の修復手法として用いるのが良いと考える。

7. おわりに

本稿では損失関数の種類や識別器の学習方法の違いの影響を検討し輪郭線の修復手法を提案した。今後は提案した手法にてさらなる修復品質の向上を目指す。

参考文献

- [1] L. Liao, R. Hu, J. Xiao, and Z. Wang, "Edge-aware context encoder for image inpainting," 2018.
- [2] K. Nazeri et al. "EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning", arXiv:1901.00212v3 [cs.CV] 11 Jan 2019.
- [3] S. Xie and Z. Tu. "Holistically-nested edge detection," In IEEE International Conference on Computer Vision (ICCV), pp: 1395-1403, 2016.
- [4] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell and A. A. Efros, "Context Encoders: Feature Learning by Inpainting," In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp: 2536-2544, 2016.
- [5] J. Canny, "A computational approach to edge detection", IEEE Transactions on pattern analysis and machine intelligence, (6):679-698, 1986.
- [6] 皆藤 優太, 田村 仁, Pix2pix を用いた画像の輪郭線に着目した画像修復, 第 18 回情報科学技術フォーラム, 第 2 分冊 pp. 53-58, June, 2019.
- [7] pix2pix-keras-byt, "https://github.com/tommyfms2/pix2pix-keras-byt", (参照 2018-10-12).
- [8] M. Arjovsky et al. "Wasserstein GAN", arXiv:1701.07875v3 [stat.ML] 6 Dec 2017.
- [9] ImageNet, "http://image-net.org/index", (参照 2018-11-16).