GAN によるカラー調整と画像補完の同時実行 *

秋本直郁 ** 林 昌希 *** 秋月秀一 ** 青木義満 **

Simultaneous Execution of Color Adjustment and Image Completion by Generative Adversarial Network

Naofumi AKIMOTO, Masaki HAYASHI, Shuichi AKIZUKI and Yoshimitsu AOKI

In this paper, we address the problem of performing natural paste synthesis by color adjustment and image completion, in order to solve the completion problem that can specify an object appearing in a completion area. We propose a synthesis network that can extract the context features of the input image and reconstruct an image with the feature, making the inserted object appear in the completion region. In addition, we propose a ingenious method to make input images and learning method using Generative Adversarial Network (GAN) that do not require collection of high cost learning data. We show that color adjustment and image completion based on context features are executed at the same time, and natural pasting synthesis can be performed by using these proposal methods.

Key words: computer vision, computer graphics, machine learning, generative adversarial nets, convolutional neural network, image completion, inpainting, color adjustment

1. 序 論

画像補完はコンピュータビジョンやコンピュータグラフィックスの領域で盛んに研究されている問題である。従来の研究によって得られた技術は、古いフィルム写真をデジタル化した際に取り込まれる傷やノイズを修復する場合や写真に写った不要なオブジェクトを写真内から除去する場合に利用でき、Adobe Photoshop など市販ソフトに組み込まれ広く利用される技術となっている。そして、写真投稿型 SNS の普及の影響により、一部のデザイナーのみならず、多くの一般的な人々が画像の編集・加工を行うようになっていく。しかし、デザイナーのようなプロフェッショナルなスキルを持たなければ、画像補完といった高度な編集を自然に仕上げることは困難である。したがって、スキルを持たない人々による利用を想定し、簡単な操作で画像補完を可能とする補助的な技術が求められている。さらに、小さな傷やノイズを除去する補完以上に、より大きな欠損領域に対して補完可能な技術が求められている。

このような大きな欠損領域を補完する場合、補完対象の画像のコンテキストを理解した上での補完が必要となる。極端な例として、水中の画像の一部を補完する際に、補完領域を魚の画像で補完することは良いが、ノートパソコンの画像で補完することは好ましくない。水中にノートパソコンがあることは現実的ではなく、補完対象の画像は水中の画像であるというコンテキストを理解していれば防げる誤りである。同様に、オブジェクトの一部を補完する際にも、どのようなパーツで補完すると自然であるかはコンテキストを理解していると有利である。このようにコンテキストを理解して画像補完を行う手法が Pathak ら¹⁾により提案され、この手法を元にしたいいくつかの手法²⁾³⁾が提案されている。

しかしこれらの手法はいずれも、補完領域に出現するオブ

ジェクトは Deep Neural Network (DNN) によって自動的に決定されてしまう。しかし本来、画像編集は編集者にとって理想の画像に近づけるための行為であり、望ましい結果を得やすくするために、補完の際に編集者の意思が反映される余地があるべきである。そこで本稿では、カットアンドペーストによって補完領域にオブジェクトを貼り付けることで、補完領域に出現するオブジェクトを編集者が明示的に指定し、オブジェクトが指定されていない領域は DNN によって自動的に補完される方法をとる。この方法により、Pathak ら¹⁾の手法と同様にコンテキストを利用して大きな領域に対して、編集者の意思を反映した画像補完が可能となる。しかし、オブジェクト画像を元のまま貼り付けると不整合な結果となり易い。補完対象である背景画像と挿入するオブジェクト画像は別の状況で作成された2枚の画像であり、色合いが異なる場合が多い。したがって、補完対象である背景画像に合わせてオブジェクト画像の色合いを変更することで、オブジェクト画像を自然に背景画像に溶け込ませた生成画像が得られる。このように、画像補完を行いつつ、カラー調整をしてオブジェクトを自然に貼り付けることができれば、出現するオブジェクトを自由に選択し、画像補完を行うことができたと言える。

したがって本研究では、出現するオブジェクトをカットアンドペーストで指定する補完問題を、カラー調整と画像補完で自然な貼り付け合成を行う問題として取り組む。

この問題の解決に利用できる従来手法には、二画像の輝度勾配を用いたカラー調整の手法と Convolutional Neural Network (CNN) を用いた画像補完手法がある。詳細な説明は第2章で行うが、これら従来手法の課題をまとめると、局所的な特徴のみでカラー調整が行われていることと画像補完の際に出現するオブジェクトが指定できないことの二点がある。

そこで本稿では、カラー調整と画像補完で自然な貼り付け合成を行う問題を解決するために、コンテキストを考慮したカラー調整を行いつつ、画像補完を行う手法を提案する。Iizuka ら²⁾により提案されたネットワークを本研究の問題設定に合わせて改良し、挿入するオブジェクト画像を明示的に補完領域に

* 原稿受付 平成 30 年 5 月 7 日

掲載決定 平成 30 年 8 月 2 日

** 慶應義塾大学 (神奈川県横浜市港北区日吉 3-14-1)

*** Recreation Lab. (東京都千代田区大手町 1-6-1 大手町ビル 4 階)

出現させるようにする。その際、従来手法と同様にコンテキストを考慮した補完をするために CNN と Generative Adversarial Network (GAN)⁴⁾ を利用し、背景画像全体からコンテキストに関する特徴を抽出する。さらにそのコンテキストの特徴を、画像補完のためだけでなく、カラー調整にも利用することで、コンテキストを考慮したカラー調整を行う。このようにして、上記二点の課題を同時に解決するネットワークを実現する。

具体的な提案手法を以下に簡潔に述べる。ネットワーク構造は、encoder-decoder 型 CNN である Generator と、Discriminator から構成される GAN である。本稿では、Generator 部分のネットワークを合成ネットワーク、Discriminator 部分のネットワークを識別器ネットワークと呼ぶ。合成ネットワークの encoder 側の畳み込み層により背景画像のコンテキストをエンコードし、decoder 側のネットワークがコンテキスト特徴を元に画像を再構成する。さらに、合成ネットワークに skip-connection を持たせることで、入力画像のエッジがある程度保存された状態から画像を再構成することになるので、補完領域に挿入オブジェクトを出力しつつ、カラー調整と画像補完を行うことができる。また、学習時には合成ネットワークの出力画像が自然であるかを評価するための loss として、識別器ネットワークから算出される Adversarial loss⁴⁾ を用いることで、合成ネットワークは画像補完を行うと同時に背景画像のコンテキスト特徴を使いつつ大域的整合性が取れたカラー調整も行うように学習される。また、学習データとしては、一般に背景画像にオブジェクトを挿入する問題では一つの背景につき無数の正解が考えられる上に、人手で自然な挿入後の正解画像を作成するのはコストが高い。そこで本手法は、合成ネットワークへの学習用入力画像の作成方法に関して、人手を必要としない、シンプルかつ有効な方法を提案する。以上の方法を用いて、一つのネットワークで、コンテキストを考慮した挿入オブジェクト画像が背景に馴染むカラー調整の実行と同時に、挿入オブジェクト画像が補完領域に出現する画像補完の実行を実現する。

本稿の構成は以下の通りである。第2章では従来研究を取り上げ、その問題点を指摘し、本研究との差分を明らかにする。第3章では提案手法についてより詳細に解説する。第4章では本手法の有効性の確認のため、ネットワークを風景画像データセット Places2 を用いて学習し、その結果例の図示と考察を行う。最後に第5章で本研究で得られた知見と展望をまとめる。

2. 従 来 手 法

カラー調整：カラー調整の古典的手法として、Poisson Image Editing⁵⁾ があり、背景画像と合成するオブジェクト画像の輝度勾配が滑らかになるように輝度値を変更することで、オブジェクト画像が背景画像に溶け込んだ自然な貼り付け合成を実現する。しかしながら、二画像の輝度が同程度になる性質があるので、オブジェクト画像と背景画像の輝度値が本来大きく異なる場合には不自然な結果となるという課題がある。これは局所的な特徴のみでカラー調整を行うことが原因であり、我々の手法では背景画像のコンテキストと合成するオブジェクトの関係性を考慮した上でカラー調整の度合いを決定しこの課題を解決する。

画像補完：画像補完における機械学習を用いない画像処理的アプローチを用いた古典手法として、diffusion ベース⁶⁾⁷⁾ あるいは patch ベース⁸⁾⁹⁾ の手法があるが、補完できる領域が小さく、セマンティックなコンテンツの補完には十分ではな

い。現在盛んに研究されている学習ベースの手法では CNN や GAN を用いた手法があり、GAN を用いて画像補完を行う手法は特に有望視されている。CNN を用いた初期の補完手法はノイズの除去もしくは小さな領域に対する補完に留まっている一方で、GAN を用いた補完手法では大きな領域を補完することが可能である。画像補完のタスクに初めて GAN を利用したのは、Pathak らによる Context Encoder (CE)¹⁾ である。CE は encoder-decoder 構造の CNN である。生成された画像と正解画像とでのピクセルごとの差を計る L2 loss と、Discriminator から算出される Adversarial loss の二つの loss を同時に最適化してネットワークの学習を行う。この方法により 64×64 pixels の大きな領域に対してコンテキストを考慮した補完が可能であることを示している。Yang ら³⁾ は CE ベースのネットワークを用いてコンテンツを生成し、次の段階において、生成されるテキストチャとその周囲のテキストチャのそれぞれが学習済み分類ネットワークでの中間特徴において近くなるよう繰り返し最適化することで、高解像度で詳細なテキストチャを補完する手法を提案した。しかし、コンテンツとテキストチャの制約を同時に最適化するプロセスが繰り返し必要なため、実行速度が遅いという課題がある。一方で我々の手法は、一度の feed-forward で補完を実行できるため実行速度は比較的早い。このことはインタラクティブな画像編集を行うことを想定した場合、非常に有利な要素となる。Iizuka ら²⁾ は dilated convolution¹⁰⁾ によって受容野を広げることと、大域的な整合性と局所的な整合性の両方を評価する Adversarial loss を用いて Completion Network の学習を行うことが補完の質の向上に有効であることを示した。大域的な整合性は、補完対象画像の全体を畳み込む Global Discriminator により評価される。一方で、局所的な整合性は、補完領域を中心とした局所的な範囲の画像を畳み込む Local Discriminator により評価される。学習が完了した Completion Network は大域的かつ局所的に整合性の取れた補完画像を生成できるようになる。しかし、Iizuka ら²⁾ の手法では、どのようなオブジェクトを用いて欠損領域を補完するかはネットワークにより自動的に決定されてしまっているため、補完領域に出現するオブジェクトを人の意思で決定できないという課題がある。一方で我々の手法は、補完領域に出現するオブジェクトを別の画像からカットアンドペーストすることで明示的にオブジェクトの指定をできるようにする。しかし、そのまま貼り付けを行うのみでは、元の画像と補完したい画像の色合いが異なることに起因した不整合な結果となってしまうため、貼り付けるオブジェクトの色合いを補完したい背景画像に合わせて変化させることで補完領域に対してオブジェクトを自然な見た目で挿入することが可能となっている。

3. 提 案 手 法

3.1 システム概要

図1に提案するシステムの概要をネットワークの学習時とテスト時（画像合成実行時）に分けて示す。学習時は、大きく4つの段階に分けることができ、合成ネットワークに入力する画像の作成、合成ネットワークでの合成画像の生成、識別器ネットワークによる損失の算出、そして誤差逆伝播法によるパラメータ更新である。合成ネットワークに入力する画像の作成は、一つのネットワークによって、画像補完とカラー調整の両方を行えるよう学習させるために重要な処理である。この詳細に関しては3.4節で述べる。また、合成ネットワークと識別器

Fig.1 Overview of our system of (a) learning and (b) test procedure. Note that there is difference between learning procedure and test procedure

ネットワークの構造や働きに関する詳細はそれぞれ 3.2, 3.3 節で詳細に述べる。3.5 節では損失関数と Adversarial Training⁴⁾ に関して述べ、ネットワーク全体を end-to-end で学習出来ることを示す。一方で、テスト時（画像合成実行時）には学習が完了したネットワークを用いることで、図 1(b) のように 2 つの段階のみで合成結果を生成することができる。入力画像の準備として、利用者が背景となる画像と挿入したいオブジェクト画像に簡単な操作を行う。この操作に関する考察は 4.2 節で行う。そして、準備された入力画像を学習済みの合成ネットワークに feed-forward することで、合成画像を生成することが可能である。テスト時（画像合成実行時）の入力画像の準備は学習時の入力画像の作成とは異なること、また、テスト時（画像合成実行時）には識別器ネットワークの使用や損失計算、パラメータ更新を行わないことに注意されたい。

3.2 合成ネットワーク

合成ネットワークの構造を図 2 に示す。合成ネットワークは Iizuka ら²⁾ の Completion Network の構成を元にしており、encoder-decoder 型の CNN である。この CNN は全ての層が畳み込み層である Fully-Convolutional Network¹¹⁾ である。中間層に dilated convolution 層をもっており、受容野を通常の畳み込み層よりも広げる働きをする。コンテキストを元に画像補完を行う際には受容野を広げることが有効であることが Iizuka ら²⁾ によって示されている。我々の問題設定では、画像補完だけでなく、挿入するオブジェクト画像のカラー調整を行う必要がある。そのために、我々が提案する合成ネットワークでは、さらに挿入するオブジェクトのエッジ情報を含む特徴を正確に出力側に伝えるための skip-connection 構造を持たせる。同じような構造を持つ U-net¹²⁾ と異なる点として、skip-connection によって伝えられた feature map は、出力側の feature map にチャネル方向に結合するのではなく、出力側の feature map に加算する。そうすることで、skip-connection によって挟まれている中間層は、入力される画像特徴と出力される画像特徴の変化分である差分を抽出する働きをする。つまり、合成ネットワークのはじめの層からつながる skip-connection により、入力されたオブジェクト画像の特徴は直ちに後方の層に伝えられているので、中間の層では、補完領域の差分とオブジェクト画像領域での色合いの差分を算出する。畳み込み層の構成は Iizuka ら²⁾ の Table2 と同じであるが、入力が 6 ch であり、2 つの skip-connection をもつ構造である点が異なる。

3.3 識別器ネットワーク

識別器ネットワークの構造を図 3 に示す。識別器ネットワークには Iizuka ら²⁾ により提案された Global Discriminator と

Fig.2 Synthesis network architecture. Synthesis network is Fully-Convolutional Network with skip-connection and dilated convolution layer. Input are total 6 ch of foreground and background images. Output is a synthesized RGB image

Fig.3 Discriminator network architecture. Discriminator network is used for the learning of synthesis network by evaluate global consistency and local consistency of input image. Output is the probability that the image is real

Local Discriminator のネットワークと同様のものを利用する^{(2) の Table3 参照}。この識別器ネットワークは 2 つのストリームから成り、一方は合成ネットワークで生成された 256×256 pixels の画像を入力とし、大域的な整合性を評価するためのストリームである。もう一方は、補完領域を中心として 128×128 pixels で切り出した画像を入力とし、局所的な整合性を評価するためのストリームである。合成ネットワークが生成した出力画像の大域的かつ局所的な整合性を評価するために、これらからのストリームを全結合層によって結合し、最終的に入力された画像が Real 画像である確率を算出する。

識別器ネットワークに入力される画像は 2 種類あり、合成ネットワークで生成された合成画像とデータセットに含まれる真の画像である。合成ネットワークで生成された合成画像を Fake 画像、真の画像を Real 画像とみなし、識別器ネットワークに入力することで Adversarial loss を算出できる。識別器ネットワークの出力は入力された画像が Real 画像である確率 $[0, 1]$ であるので、Adversarial loss は、Real 画像が入力された場合、出力値と 1 との差、Fake 画像が入力された場合、出力値と 0 との差である。

3.4 合成ネットワークの入出力画像の操作

3.4.1 学習データの準備

一般に、DNN を高い汎化性能を持つよう学習するためには大量の学習データが必要になる。例えば、分類問題で利用される公開データセット ImageNet¹³⁾ には 1,400 万枚以上の画像が教師データとしてカテゴリラベルとセットで用意されている。ネットワークには多くのパラメータがあり、様々な関数を近似する表現能力をもつ一方で、学習データが少ない場合にはその

Fig.4 Overview of learning procedure. We explain how to make (f)input image is in 3.4.2 and (h)synthesis image is in 3.4.3

データのみに対し最適化されてしまうため、テストデータで性能が出ない過学習という現象に陥ってしまう。したがって、大量の学習データを得られない場合には過学習をさけるために学習データの拡張が行われることが多い。本研究では、背景画像に対して画像補完をしつつ前景オブジェクトを自然に挿入するという問題設定を行なっている。この問題設定に最適な教師データを大量に集めることはコストが高い。背景となる画像を大量に収集することは容易であると言えるが、挿入するオブジェクトの画像を大量に収集することは困難である。理由として、テスト時には一つの背景画像に対して挿入が想定されるオブジェクトの数は無数であるから、教師データとして背景と挿入オブジェクトが1対1の関係になることは好ましくなく、学習時にも一つの背景に対して様々なオブジェクトが用意されることが望ましいが、背景に対してどのオブジェクトが挿入されるに適切か、さらに挿入する位置や大きさを判断するには人手が必要であり、時間のかかる作業であることが挙げられる。3.4.2項で述べる入力画像の作成方法を用いることで、以上のような高コストな学習データの収集を回避することができる。さらに3.5節で提案する方法により自己教師学習が可能のため、人手によるアノテーションやラベル付けを一切必要としない。

3.4.2 合成ネットワークへの入力画像の作成

ここでは、学習時における合成ネットワークへの入力画像の作成について説明する。最終的に、合成ネットワークへ入力される画像は、以下で説明する操作がなされた前景画像と背景画像の計6chで構成される。しかし、 256×256 pixelsの元画像の一部の領域から前景画像を作成するので、学習時には挿入するオブジェクト画像を用意する必要はない。まず、 128×128 pixels以下の矩形領域を持つ 256×256 pixelsのバイナリマスク M_G (図4(a))をランダムに生成する。このバイナリマスクと元画像 x (図4(b))での要素ごとの積をとって前景画像 x_f (図4(c))を作成する。一方で、背景画像 x_b (図4(d))はバイナリマスクを反転させて、元画像と要素ごとの積を取ることで作成する。したがって、オブジェクトを合成する領域では背景情報を利用しない。本手法はオブジェクト除去での利用も想定しており、元画像のその領域に除去する物体があると周辺とは異なる色情報が含まれるので利用すべきでないことが理由として挙げられる。前景画像に対してのみ、さらに次の二つの操作

を行う。まず、前景画像のRGBの三つの画素値それぞれをランダムに0.8から1.2倍して変化を加える。255以上になる画素値は255に丸める。次に、前景画像の領域内で白色の領域を作成する。この際、0以上の幅を持つ枠で白色の領域を作り、画像補完を行うべき領域とみなす。これら二つの操作をした前景画像 $\hat{x}_f$ (図4(e))は、元の前景画像と比べて色合いが異なり、さらに補完すべき領域を持つ画像になっている。式で表すと、非線形関数を g として、 $\hat{x}_f = g(x_f)$ である。この操作した前景画像と背景画像をチャンネル方向に結合し、6chの画像として合成ネットワークへの入力画像 (図4(f))とする。

このように、矩形の補完領域に対して矩形の前景画像を挿入して学習を行っているが、4.1節で見るように、テスト時には、任意の形状の領域を補完することや任意の形状のオブジェクト画像を挿入することができる。また、一つの元画像に対して、前景画像の切り出し位置と前景領域の輝度の変更が無数に考えられるため、データ数の拡張を行える利点がある。さらに、矩形バイナリマスクをランダムに作成することで合成ネットワークへの入力画像を作成し、元画像を生成の正解として扱う自己教師学習として利用することで、ラベル付けやアノテーションを必要とせずに学習データを用意できるという利点がある。

3.4.3 合成ネットワークからの出力画像の操作

次に合成画像 (図4(h))の作成について説明する。合成ネットワークから出力される出力画像 (図4(g))は 256×256 pixelsの画像であるが、利用するのは前景領域のみである。前景画像を切り出す際に利用したバイナリマスクを用いて、出力画像から前景領域を切り出し、背景画像と合わせることで最終的な合成画像とする。この合成画像が識別器ネットワークにFake画像として入力される。

3.4.4 テスト時の入力画像の準備

テスト時 (画像合成実行時) は、背景画像と挿入するオブジェクト画像が必要である。本稿では、オブジェクト画像はあらかじめ人手で領域を切り出したものを利用する。そのため、MattingやSegmentationについては行わない。学習時と同様、合成ネットワークに入力する画像は前景画像と背景画像の計6chである。前景画像は白色で示された補完領域と切り出されたオブジェクトを合わせた画像である。

3.5 学 習

二つのネットワークは、GAN の枠組みの中で Adversarial loss を用いて交互に学習が行われ、合成ネットワークに対してのみ Mean Squared Error loss(MSE loss) も利用する。

Adversarial loss は、識別器ネットワークを通して算出され、合成画像の大域的な整合性と局所的な整合性を評価する損失関数として利用できる。したがって、この損失関数が特に画像補完のための学習に寄与していると考えられる。この Adversarial loss を用いた目的関数 F_{adv} は、

$$F_{adv} = \min_G \max_D \mathbb{E}[\log D(x, M_G) + \log(1 - D(G(\tilde{x}_f, x_b), M_G))] \quad (1)$$

で表され、 D は識別器ネットワークを表す関数で、入力された画像が Real 画像である確率を返す関数である。一方 G は合成ネットワークを表し、出力として 3 ch の画像を返す関数である。

MSE loss は、合成ネットワークの学習に Adversarial loss とともに利用し、前景画像と出力画像のピクセルごとの差を評価する。その際、合成画像を作成するために利用するのは出力画像では前景領域のみなので、前景領域を切り出すために作成したバイナリマスクを利用して、前景領域のみでの MSE loss を算出するようにする。このマスクを使った MSE loss は、

$$L(x, M_G, \tilde{x}_f, x_b) = \|M_G \odot (G(\tilde{x}_f, x_b)) - x\|_2^2 \quad (2)$$

で表され、 $\odot$ は要素ごとの積を表す。この MSE loss による自己教師学習によって、合成ネットワークが、背景画像を条件として、操作をした前景画像を元の前景画像に近くなるように出力するよう明示的に学習することができる。したがって、この損失関数が特にカラー調整のための学習に寄与していると考えられる。また、MSE loss を併用することによって、GAN の学習を安定させることができることが一般に知られている¹⁾。

最終的な目的関数 $F_{MSE+adv}$ は、

$$F_{MSE+adv} = \min_G \max_D \mathbb{E}[L(x, M_G, \tilde{x}_f, x_b) + \log D(x, M_G) + \alpha \log(1 - D(G(\tilde{x}_f, x_b), M_G))] \quad (3)$$

となり、 α は MSE loss と Adversarial loss のバランスを決定するハイパーパラメータである。これらの損失関数と提案した入出力画像の操作方法によって、二つのネットワークを Adversarial Training の枠組みで end-to-end の学習をすることが可能である。

4. 実 験

本章では提案手法の有効性の確認のため、本節でネットワーク学習の詳細と評価方法に関して述べ、4.1 節以下で実験結果の例示とその考察を行う。

データセット：本実験では訓練用のデータセットとして Places2¹⁴⁾ を利用する。Places2 はシーンコンテンツ、物体認識、行動予測やイベント予測といった高度な視覚的理解システムを訓練するために風景画像で構築された公開データセットで、Iizuka ら²⁾ の実験でも利用されている。このデータセットには山岳風景やベツルムのような様々な種類の風景画像が含まれている。各画像に対してシーンカテゴリがラベル付けされており、一つのカテゴリに対して 5,000 から 30,000 枚の訓練画像が用意されている。データセット全体としては、400 以

上のシーンカテゴリに分類できる 1,000 万枚の画像が用意されている。本実験では、そのうち 9.5 万枚の画像を訓練データとして用いてネットワークを学習した。しかしシーンカテゴリのラベルは使用しない。

実装詳細：我々の GAN のネットワーク構造は 3 章で説明した通りである。画像は 256×256 pixels にリサイズされたものを使用した。訓練データの中からサンプルをランダムに選択してミニバッチを作成し、誤差を算出した。ミニバッチサイズは 32 である。MSE loss と Adversarial loss の重み $\alpha = 0.0004$ である。誤差が最小に近付くようネットワーク各層のパラメータを更新するために最適化手法として ADADELTA アルゴリズム¹⁵⁾ を使用した。ADADELTA アルゴリズムは各パラメータの学習率が自動的に決定される特徴をもつ。合成ネットワークの事前学習として MSE loss のみによる学習を 100,000 イテレーション行い、合成ネットワークと識別器ネットワークの交互の学習は 700,000 イテレーション行なった。学習には NVIDIA TITAN 1080Ti 2 枚で約 2 週間を要した。また、深層学習ライブラリとして PyTorch を用いた。

評価方法：GAN により画像生成を行う研究には、スタイル変換¹⁶⁾、カラー化¹⁷⁾、全身画像の生成¹⁸⁾、テキストから画像の生成¹⁹⁾ など様々な種類があるが、いずれにしても生成された結果画像を用いてどの手法が優れているか客観的かつ定量的に評価する方法は確立されていない。超解像問題²⁰⁾ であれば、あらかじめ用意された高解像の画像を低解像に変換し入力として用いることで、どの程度元の高解像画像に近づけることができたか評価することは可能であるが、その他多くの問題設定では生成結果として決定的な正解が存在しないため、定量的な評価を行うことが難しいとされる。本稿では手法の評価のために、元画像からオブジェクトを手動で取り除き、再びそのオブジェクトを比較する各手法で合成する。それら生成結果例の図示で定性的な評価を行う。さらに、合成結果と元画像での Root Mean Square Error (RMSE) をあくまで参考値であるが示し、定量的な評価を行う。

4.1 カラー調整と補完の同時実行

結果：実験の結果例を図 5 に示す。例えば図 5(a) は、矩形で白抜き領域を作成し、その上から木の画像を挿入する例である。この実験では、周囲に合わせて木のカラー調整を行うことと、白抜きされた領域を補完することの二つのタスクを行う。(a) 以外も同様である。また、図 6 に片方のタスクのみの生成結果を参考として示す。図 6(a) は画像補完のタスクのみ、図 6(b),(c) はカラー調整のタスクのみである。

考察：図 5(a) のように、白抜き領域は周囲に合わせて自然に補完されていることがわかる。また、木の葉の色の変化のように輝度値が周囲の葉と似るよう調整されていることがわかる。これらのことから、一つのネットワークで、カラー調整と画像補完が同時に実行されたことが言える。さらに図 6 のように、同様のネットワークでカラー調整と画像補完のどちらか一方のみの実行も可能であることがわかる。また、任意の形状の領域を補完することも可能であることがわかる。

4.2 ユーザーが行う操作

本手法を利用することによって、ユーザーが簡単な操作のみ行えば入力画像を作成でき、自然な貼り付け合成が行えることを示す(図 7)。本実験では、画像中の人物の置き換えを想定する。まず、元画像(図 7(a))に対して、挿入したい画像を上か

Fig.5 Synthesis results by our approach. Inserted object's color is changed according to context. White region represent completion area

Fig.6 Additional results of our approach. (a) is only completion. (b) and (c) are only color adjustment

ら貼り付ける操作を行う（図 7(b)）。次に、上から貼り付けを行なったにもかかわらず、隠しきれなかった不要なオブジェクトがあれば白く塗りつぶす操作を行う（図 7(c)）。以上の手順により、ユーザーが行う入力への準備は終わり、背景を黒塗り画像に差し替えることで、ネットワークに入力する「操作した前景画像」の状態になる（図 7(d)）。挿入画像に被らないように白塗りを行ったり、背景を黒塗り画像に差し替えたりする操作は、市販のペインティングツールに一般的に搭載されているレイヤー機能を使うことで簡単に実現できる。このように、画像の貼り付けと領域の選択という簡単な操作のみをユーザーが行えば良く、指によるスマートフォンの操作でも行える単純

な操作であるので、スマートフォンのアプリケーションに搭載することも可能であると考えられる。

4.3 挿入オブジェクトや補完形状の差異による影響

結果：図 8 に本手法を用いて、同一の元画像に対する異なるオブジェクトもしくは補完領域での合成結果を示す。図 8 の (a),(b),(c),(d) は元画像と補完領域を同一に揃え、複数の異なるオブジェクトを挿入した場合の結果である。(a),(b) は元画像のシーンに適する「人物」の挿入で、(c),(d) はシーンからは想定されないオブジェクトの挿入である。特に (a) は左から二番目の人物をコピーし位置をずらして貼り付けた場合である。また、(e),(f) は元画像と挿入するオブジェクトを同一にし、補完領域の形状を変えた場合である。

考察：元画像からカラー調整と補完の領域を切り出してネットワークの学習をしているため、元画像とオブジェクトが似た状況での学習なので、元画像のコンテキストに合わないオブジェクトでの合成が可能か懸念される。しかし、この実験結果によると、シーンに関係しないオブジェクトの挿入であっても合成後は色合いがより自然になっているので、このような場合の合成も可能であると言える。また、(a) の結果のように、元画像にあるオブジェクトを再び挿入すると色合いは変化することがわかる。補完領域の形状の違いとして、矩形領域である (e) の方が (f) より自然に補完されているように見える。これは、学習時に矩形領域を補完していることに起因していると考えられる。任意の形状にした場合、領域のエッジの色が濃くなっていることが見て取れる。さらに、補完領域の差異によりオブジェクトの色合いに影響を与えていることも見て取れる。

4.4 他手法との比較

定性評価：図 9 に本手法と他の組み合わせ手法の比較を示す。図 9(a) は入力画像である。図 9(b) は本手法を用いた場合の合成結果を示しており、(a) 入力画像を合成ネットワーク

Fig.7 Operations in test procedure. User only performs simple operations of (b) and (c). (e) is an actual synthesis result

Fig.8 Results by our approach for the same background. (a), (b), (c) and (d) are different inserted objects. (e) and (f) are different completion areas

Fig.9 Comparison with existing works. We compare with (c)Photoshop functions and (d)Iizuka²⁾+Poisson Image Editing⁵⁾

Table1 Root Mean Square Error between original images and outputs using some methods

Evaluated Target	Color Adjustment and Completion Area	Completion Area	Color Adjustment Area
PatchMatch + Color Adjustment	9.79	9.27	10.18
Iizuka <i>et al.</i> ²⁾ + Poisson Image Editing	9.28	9.39	9.38
Proposed method	10.0	9.87	10.12

に入力して生成した。一方、図 9(c) は補完手法として Adobe Photoshop に搭載されている PatchMatch (「コンテンツに応じた塗りつぶし」という機能) と、カラー調整として背景を参照してオブジェクト画像の「カラー適用」を行なった場合の結果である。図 9(d) は、補完手法として Iizuka ら²⁾ の手法と、カラー調整として Poisson Image Editing を行なった場合の結果である。(c) と (d) は、はじめに (a) 入力画像の挿入オブジェクトを含まずに PatchMatch・Iizuka ら²⁾ で画像補完のみを行い、次に挿入オブジェクトのカラー調整を行った。

コンテキスト特徴を利用して補完を行う (b) と (d) はカバンの領域を地面と区別して補完できることが確認できる。また、(b) では挿入オブジェクト (人物) が周囲の山の色合いと同じ

にはならず適切に調整されていることが確認できる。

定量評価: 表 1 に各手法の合成結果と元画像でのピクセルごとの画素値 (0~255) の差から算出した RMSE の結果を示す。マスクを使うことで、RMSE を算出する領域を「カラー調整と補完両領域」と「補完領域のみ」、「カラー調整領域のみ」に分けることができる。算出する手順について以下に示す。まず、元画像からオブジェクトを切り出す。そのオブジェクト画像の色合い、コントラスト等をランダムに変化させる。さらに、適切な領域を選び、補完領域とする。オブジェクト画像を元の位置で挿入することで、色合いの調整と画像補完を行なった後で、元画像とピクセルごとの差を比較することができる。使用する手法はいずれも、切り出したオブジェクト領域の情報を利

Fig.10 Failure result. Large object insearting and completion are difficult

用しないため、元画像から切り出すことが許される。以上の方法でそれぞれ 20 パターンの合成結果を用意した。背景に対して自然なオブジェクトの色や補完は元画像と同一である必然性はないので、あくまで参考値であるが、定量評価の指標として RMSE を算出した。補完領域のみの値から元画像に近い補完は PatchMatch により、カラー調整のみの値から元画像に近くなるカラー調整は Poisson Image Editing により可能であることが分かる。必要であれば Poisson Image Editing を提案手法の後処理として利用することも可能である。

4.5 失敗例

結果：提案手法が上手く機能しない例を図 10 に示す。図 10 は比較的大きなサイズのオブジェクトを挿入し、カラー調整と画像補完を行なった場合の結果である。挿入するオブジェクトが画像全体の面積の約 2 分の 1 を占めている。合成結果では、挿入したオブジェクトは全体的に空の色に近くなっており、自然なカラー調整とは言えない。また、補完領域は雲の模様を再現することはできているが、背景の空の色とは異なる色が生成されている。

考察：図 10 は挿入オブジェクトがある程度大きくなると上手くカラー調整できなくなることを示している。学習時にはカラー調整と補完の領域が 4 分の 1 以下になる設定にしているため、大きなサイズではうまく機能しなくなると考えられる。大域的な整合性を評価する識別器での学習により、生成器は生成される画像の大部分は背景画像の色になるべきと学習されているのではないかと考えられる。そのため大きなオブジェクトを挿入しようとすると背景の色がオブジェクトに掛かってしまうと考えられる。また、空のように一様な背景を補完すると境界付近で色の不整合が目立つことがわかる。

5. 結 論

本稿では、補完領域に出現するオブジェクトを指定可能な補完問題を、カラー調整と画像補完で自然な貼り付け合成を行う問題として取り組んだ。この問題設定に合わせて、入力画像のコンテキスト特徴を抽出しつつ、挿入オブジェクトを補完領域に出現させて画像を再構成することのできる合成ネットワークを提案した。さらに、高コストな学習データの収集を必要としない、工夫された入力画像の作成方法と学習方法を提案した。

これらの提案手法を用いて、一つのネットワークでコンテキスト特徴を元にしたカラー調整と画像補完を同時に実行し、自然な貼り付け合成ができることを示した。

今後の展望として、セマンティックセグメンテーションをシステムに組み込むことを考えている。現在、テスト時に挿入オブジェクトを準備する際には手でオブジェクトの切り出しを行なっている。セマンティックセグメンテーションによって、ラベルを指定することで挿入オブジェクトを切り出すことができれば、より一層簡単な操作のみで自然な貼り付け合成を行うことができるようになる。

参 考 文 献

- 1) D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.
- 2) S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017.
- 3) Chao Yang, et al. High-Resolution Image Inpainting Using Multi-Scale Neural Patch Synthesis. In *CVPR*, 2017.
- 4) I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.
- 5) Patrick Perez, et al. Poisson Image Editing. In *SIGGRAPH*, 2003.
- 6) C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE transactions on image processing*, 10(8):1200 - 1211, 2001.
- 7) M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417 - 424. *ACM Press/Addison-Wesley Publishing Co.*, 2000.
- 8) A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341 - 346. *ACM*, 2001.
- 9) A. A. Efros and T. K. Leung. Texture synthesis by nonparametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033 - 1038. *IEEE*, 1999.
- 10) Fisher Yu and Vladlen Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. In *ICLR*, 2016.
- 11) Jonathan Long, et al. Fully Convolutional Networks for Semantic Segmentation. In *CVPR*, 2015.
- 12) Olaf Ronneberger, et al. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, 2015.
- 13) Deng, J. and Dong, W. and Socher, R. and Li, L.-J. and Li, K. and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.
- 14) Bolei Zhou, et al. Places: A 10 million Image Database for Scene Recognition. In *TPAMI*, 2017.
- 15) Matthew Zeiler. ADADELTA: An Adaptive Learning Rate Method. In *CoRR* abs/1212.5701 (2012).
- 16) L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.
- 17) S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Transactions on Graphics (TOG)*, 35(4), 2016.
- 18) L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool. Pose guided person image generation. In *NIPS*, 2017.
- 19) Han Zhang, et al. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. In *ICCV*, 2017.
- 20) Christian Ledig, et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *CVPR*, 2017.