

修士論文

機械学習を用いたイラスト風顔画像の生成

広島大学大学院先進理工系科学研究科

山内優太

令和5年3月

修士論文

機械学習を用いたイラスト風顔画像の生成

指導教官 伊藤靖朗 教授

広島大学大学院
先進理工系科学研究科情報科学プログラム

M213838 山内優太

提出年月: 令和5年2月

概要

近年、機械学習を用いた画像変換に関する研究が盛んに行われている。特に、敵対的生成ネットワーク (GAN) は、高い精度で出力を行うことができるネットワークとして広く用いられている。本研究では、GAN を用いて、人物の顔写真をイラスト風に変換する。イラスト画像は、特徴が大きくデフォルメされていたり、バリエーションも人間の顔より少ない場合が多い。そのため、そのようなイラストに類似したスタイルの画像を生成することは、従来の GAN では難しい場合が多い。本研究は、StyleGAN2 を用いた教師なし画像変換の手法を用いることで、イラスト風への高品質な画像変換が可能であることを示す。また、StyleGAN2 に特化したエンコーダーを組み合わせることで、短時間で高品質な画像変換を行う手法を提案する。

目次

第1章	はじめに	1
1.1	研究の背景	1
1.2	研究の目的	1
1.3	本論文の概要	2
第2章	関連研究	3
2.1	StyleTransferによる画風変換	3
2.2	敵対的生成ネットワーク	5
2.2.1	GAN	5
2.2.2	CycleGANを用いた教師なし画像変換	6
2.2.3	StyleGAN2を用いた教師なし画像変換	8
2.2.4	pixel2Style2pixel	15
2.3	拡散モデルを用いた画像変換	16
第3章	提案手法	17
第4章	実験	19
4.1	実験の詳細	19
4.2	データセット	20
4.2.1	顔写真データセット	20
4.2.2	顔イラストデータセット	21
4.3	実験	22
4.3.1	データセットに関する実験	22
4.3.2	pSpに関する実験	24
4.3.3	レイヤースワッピングに関する実験	25
第5章	まとめ	28

目次

1.1	本研究で行う画風変換の概要	2
2.1	StyleTransfer の全体図	3
2.2	画像変換ネットワークを用いた StyleTransfer の全体図	4
2.3	GAN の全体図	6
2.4	pix2pix の全体図	7
2.5	CycleGAN の構造	7
2.6	PGGAN の学習	9
2.7	StyleGAN の Generator の全体図	10
2.8	スタイルミキシング (論文 [1] 内の画像を引用して作成)	11
2.9	レイヤースワッピング (論文 [2] 内の画像を引用して作成)	12
2.10	StyleGAN2 を用いた教師なし画像変換の概要 (論文 [3] 内の画像を引用して作成)	13
2.11	StyleGAN2 を用いた教師なし画像変換の学習	14
2.12	StyleGAN2 を用いた教師なし画像変換の推論	15
2.13	pSp の全体像	16
3.1	提案手法の手順	18
4.1	実験に用いた StyleGAN2 の構造	19
4.2	FFHQ データセットの一部	21
4.3	All-Age-Faces データセットの一部	21
4.4	人物の顔イラストのデータセットの一部	22
4.5	データセットに関する実験の Inversion 結果	23
4.6	データセットに関する実験の画像変換結果	24
4.7	pSp を用いた実験結果	25
4.8	レイヤースワッピングの実験結果	27

第1章 はじめに

1.1 研究の背景

機械学習を用いた画像変換に関する研究は、近年盛んに行われている。敵対的生成ネットワーク (GAN) は、2014年に発表されて以降急速に発展し、高精度な画像変換を可能にした。本研究では、画像変換の中でも、入力画像のスタイルを変える画「風」変換を取り上げる。GANを用いて画風変換を行う研究には、人間の顔写真を扱ったものも多く存在する。それらの多くは、線画 [4, 5], 肖像画 [3, 6], カートゥーン風 [3, 7, 6, 2], アニメ風 [3, 8, 9, 10] 等への変換を試みている。

一般に、画像の構造はほぼ同じで細部の色を変えることで相互に変換できるような場合は、画像のドメイン間の距離が近い。例えば、顔写真と肖像画のドメイン間の距離や、馬の画像とシマウマの画像のドメイン間の距離は近いと言える。顔写真も肖像画も、顔の輪郭や目の形などはほとんど同じであり、細部の色を変えることで相互に変換しやすい。ドメイン間の距離が近い場合は、StyleTransfer[11] や CycleGAN[12] といった既存の手法で比較的高精度な画像変換が可能である。しかし、ドメイン間の構造の変化が大きい場合は、既存の手法では依然として課題がある。例を挙げれば、顔写真と車の画像や、馬の画像と人の画像は、ドメイン間の距離が大きい。

本研究では、GANを用いて、人物の顔写真をイラスト風に変換する。イラストにもさまざまな種類があるが、本研究では、いらすとや風の画像を生成することを目標とする。いらすとやは、無料でイラストを提供するウェブサイトである。いらすとやで描かれている人物の顔イラストは現実世界をデフォルメして描かれており、顔写真との変化が大きい。そのため、いらすとや風に変換する研究はこれまであまり行われてこなかった。

1.2 研究の目的

本研究では、GANを用いて、人物の顔写真をイラスト風に変換する。本研究の概要を図 1.1 に示す。GANの中でも特に高性能な StyleGAN2[13] をベースとし、イラスト風への変換を高い精度で行うことを目指す。加えて、StyleGAN2に特化したエンコーダーを組み合わせることで、短時間で高品質な画像変換を行う手法を提案する。

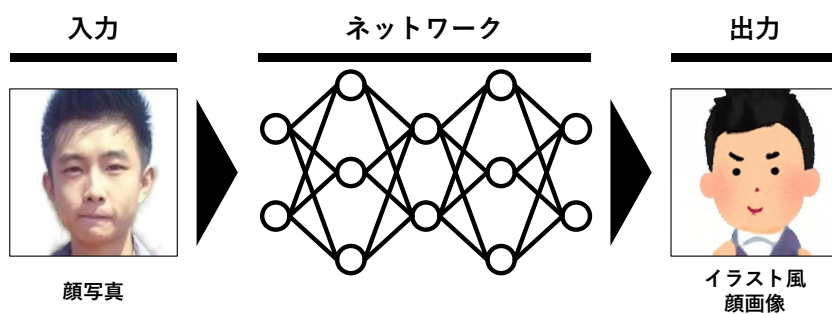


図 1.1: 本研究で行う画風変換の概要

1.3 本論文の概要

本論文は、次のように構成されている。2章では、関連研究を紹介する。3章では、提案手法について述べる。4章では、実験の詳細やデータセットについて触れた後、様々な実験と結果を示す。最後に、5章で本論文のまとめを行う。

第2章 関連研究

2.1 StyleTransfer による画風変換

StyleTransfer[11] は、ニューラルネットワークを用いた画風変換手法であり、2016年に発表された。コンテンツ画像とスタイル画像の2つの入力から1つの画像を生成する。コンテンツ画像から画像全体の構造を引き継ぎ、スタイル画像から画像のスタイルを取り込んだ画像を生成できることが特徴である。学習には訓練済みのVGG16[14]の中間出力を用いている。StyleTransferの全体図を図2.1に示す。

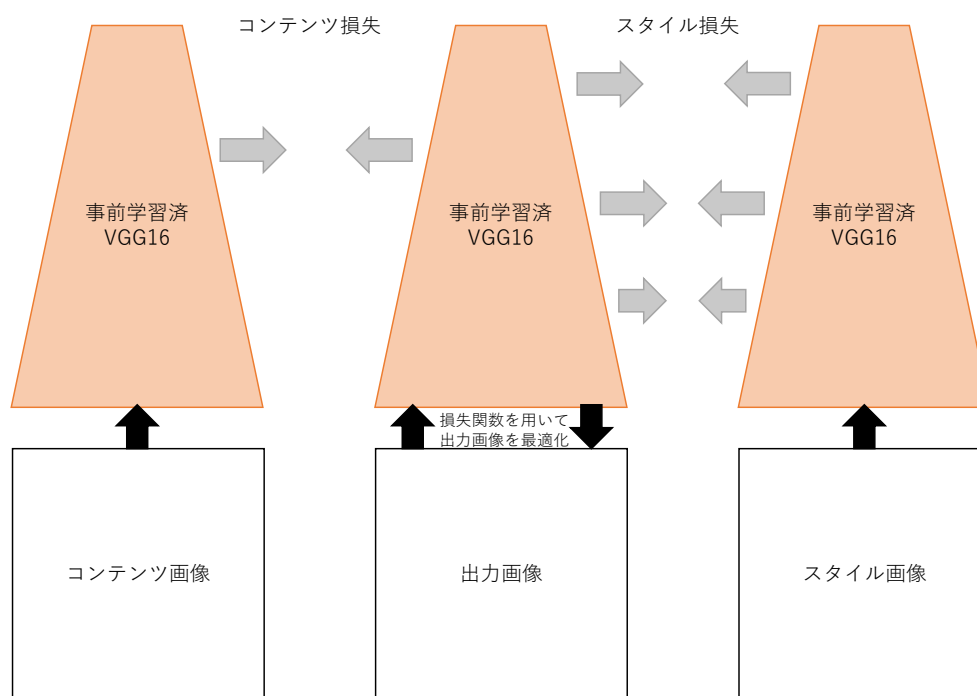


図 2.1: StyleTransfer の全体図

StyleTransferでは、2つの損失関数を定義している。一つは、コンテンツ損失である。この損失は、コンテンツ画像と生成画像をそれぞれ訓練済みのVGG16に入力し、特定の中間出力を抜き出して平均二乗誤差を計算したものである。学習が進むにつれてこの損失が小さくなると、生成画像はコンテンツ画像と似たような構造を持つ画像を生成するようになる。そしてもう一つの損失関数は、スタイル損失である。スタイル損失の計算には、

スタイル画像と生成画像を用いる。コンテンツ損失の計算時と同様に、2つの画像をそれぞれ訓練済みのVGG16に入力する。スタイル損失は、特定の間出力を抜き出してグラム行列の差分を取り、その平均二乗誤差を計算したものである。このスタイル損失によって、生成画像はスタイル画像の画風を模倣した画像を生成するようになる。

学習初期の生成画像は入力画像に全く関係のない画像を生成するが、前述した損失関数を用いた最適化を行うことで、コンテンツを保ったままスタイルを変換することができる。しかしこの手法の最大の問題点は、最適化に時間がかかることにあった。

2016年にJohnsonらによって論文が発表された[15]。この論文で提案された手法は、入力画像ごとに最適化を行うのではなく、予め画像変換ネットワークを訓練しておき、そのネットワークに推論させる。訓練時の損失関数はStyleTransferで用いていたものと同じである。全体図を図2.2に示す。

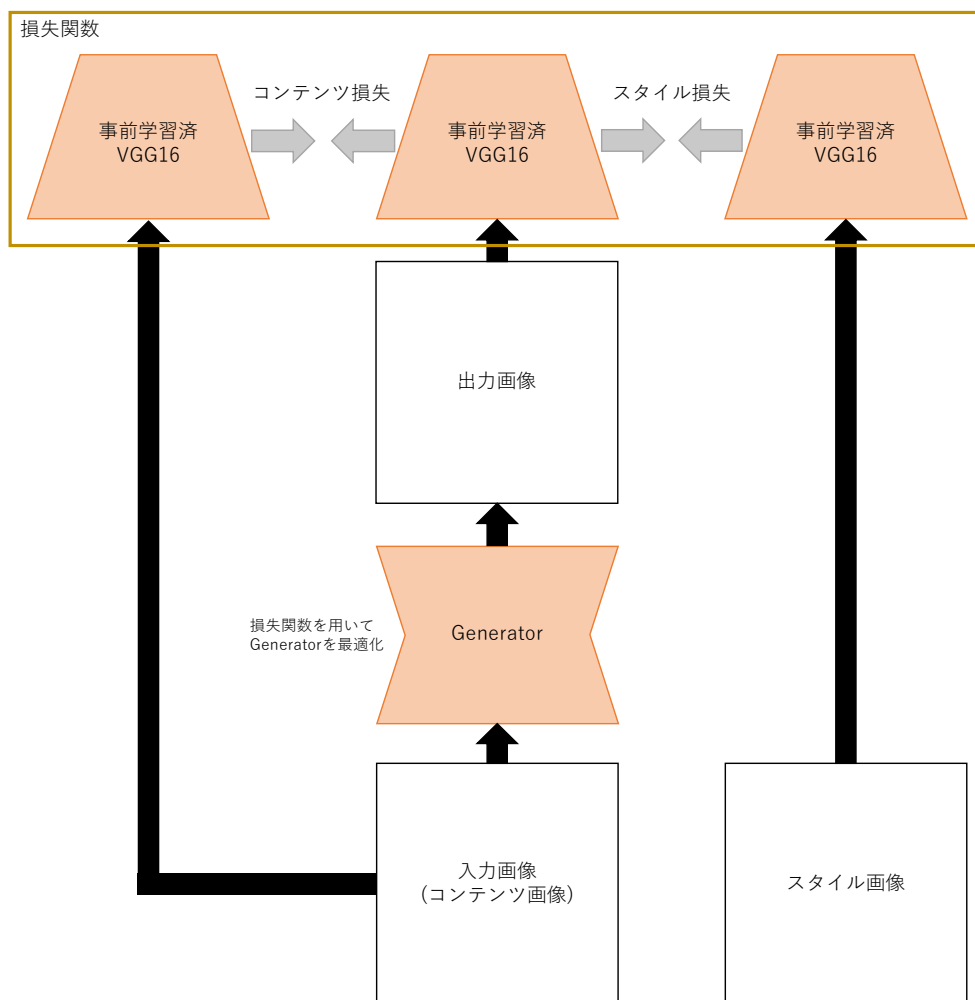


図 2.2: 画像変換ネットワークを用いた StyleTransfer の全体図

画像を Generator に入力すると、出力画像が得られる。出力画像と入力画像、スタイル画像を用いて損失を計算し、Generator は損失が小さくなるように学習を行う。入力画像

からコンテンツを引き継ぎ，スタイル画像からスタイルを引き継ぐような画像を生成すると損失が小さくなる．学習時は，スタイル画像に用いる画像のスタイルは統一しておく必要がある．推論時は，スタイル画像は不要となり，Generator に画像を入力するだけでスタイル変換が可能となる．この手法によって生成された画像は最適化手法による生成画像と同程度の品質を保ち，この手法は最適化手法と比較して推論時間を大幅に短縮した．

StyleTransfer による画風変換は，コンテンツの位置や形を変えずにスタイル変換を行う場合に最適である．例えば，写真から絵画風に変換する場合や，絵画を別の画風に変換するといった場合が挙げられる．しかし，スタイルの変換時にコンテンツの輪郭も大きく変化させる必要がある場合は，この手法では限界がある．例えば，写真をイラスト風やアニメ風に変換するような場合は，コンテンツ自体の変化が必要になるため，StyleTransfer での再現は難しい．

2.2 敵対的生成ネットワーク

敵対的生成ネットワーク (GAN)[16] は，2014 年に提案されたネットワークであり，画像変換の分野で頻繁に使用されている．GAN には様々な派生モデルがあり，今なお発展が著しい研究分野である．この節では，まずはじめに最も基本的な GAN の全体像について触れる．その後，CycleGAN[12] や StyleGAN[1] といった，よく知られている GAN の派生モデルについて解説する．本研究では，StyleGAN2[13] を Generator に用いている．

2.2.1 GAN

GAN は，入力された値を基に新たな画像を生成する Generator と，画像を識別する Discriminator から構成される．GAN の全体図を，図 2.3 に示す．Generator の入力，ノイズか画像であることが多い．Generator は Discriminator が誤って本物と判断してしまうような画像を生成し，Discriminator は入力された画像が正解画像か生成された画像なのかを正しく識別するように学習を進める．このように，Generator と Discriminator が競い合いながら学習を行うことによって，学習後の Generator から目的の画像に近い出力を得ることができる．

一般的に，GAN の Generator は，畳み込みや正規化，活性化関数などの一連の処理を一つのブロックとし，そのブロックを積み上げることで作成する．このブロックは「畳み込みブロック」と呼ばれる．Generator の畳み込みブロックは，解像度が小さい順に並んでいる．Generator は，入力を全て低解像度の畳み込みブロックに入力し，アップサンプリングを行いながら段階的にチャンネル数を減らし解像度を上げていくことで，高解像度の画像を出力する．例えば，Generator に入力されたデータは， 4×4 の畳み込みブロックに入力される．そしてその出力は， 8×8 にアップサンプリングされ， 8×8 の畳み込みブロックに入力される．その出力は， 16×16 にアップサンプリングされ， 16×16 の畳み込みブロックに入力される．これを何度か繰り返し，最終的には，例えば 256×256 の

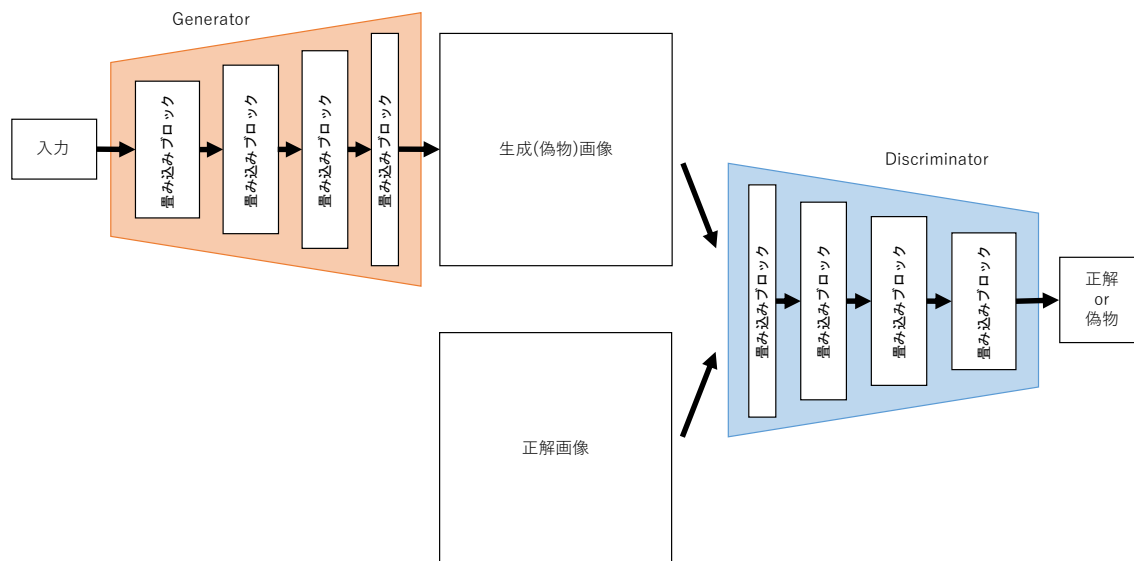


図 2.3: GAN の全体図

解像度のカラー画像を出力する。他方、Discriminatorは、Generatorを逆さまにしたような構造になっている。入力画像の情報量を段階的に圧縮していき、最終的には、入力が正解画像か偽物画像なのかを識別する。

2.2.2 CycleGAN を用いた教師なし画像変換

CycleGAN[12]は、pix2pix[17]を発展させたGANである。pix2pixは画像から画像への変換を実現したモデルである。pix2pixの全体図を図2.4に示す。構造は基本的なGANと変わらないが、Generatorはノイズではなく画像から画像を生成する。Discriminatorには、ドメイン X とドメイン Y の画像のペアを入力する。Discriminatorは、入力されたドメイン X の画像を参考に、ドメイン Y の画像がGeneratorによって生成された偽物画像か、それとも正解画像かを判定する。

このように、pix2pixは学習にペアデータセットを利用している点が特徴である。ペアデータセットとは、入力画像1枚と正解画像1枚をペアとして、それを大量に集めたものである。ペアデータセット作成のためには、訓練やテストに用いる全ての入力画像に対して適切な正解画像を用意する必要がある。そのため、ペアデータセットを用意するのは困難な場合が多く、ペアデータセットを用意できない場合はpix2pixを用いることができない。

CycleGANは、ペアデータセットなしで画像から画像への変換を可能にしたモデルである。CycleGANは図2.5のように、2つのGeneratorと2つのDiscriminatorで構成される。2つのGeneratorは、片方がドメイン X からドメイン Y への変換を行い(G_{xy})、もう片方がその逆のドメイン Y からドメイン X への変換を行う(G_{yx})。2つのDiscriminatorは、それぞれドメイン X 、ドメイン Y を担当する。ドメイン X のDiscriminatorである

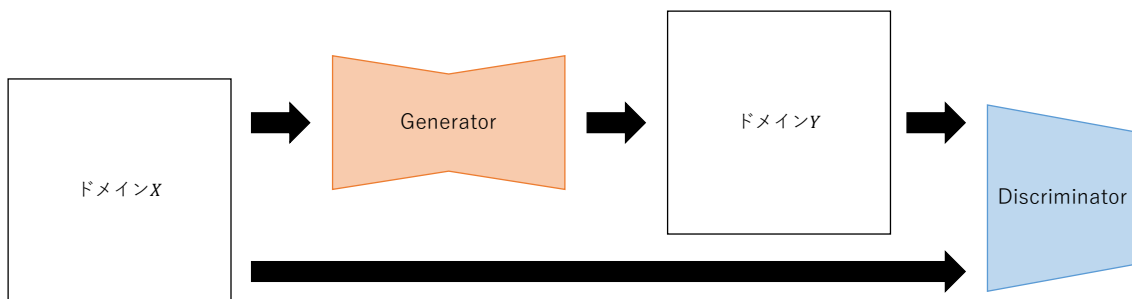


図 2.4: pix2pix の全体図

D_x には、本物のドメイン X の画像か、もしくは G_{yx} で生成された偽物の画像が入力される。Discriminator は、本物と偽物を正確に見分けるように学習を行う。一般的には、2つの Generator は同じネットワーク構造を持つ。同様に、2つの Discriminator も同じネットワーク構造を持つ。ただし重みは共有せず、別々のインスタンスとして学習を行う。ドメイン X からドメイン Y への変換とドメイン Y からドメイン X への変換は同時に訓練される。ドメイン X の画像 x はドメイン Y へ変換され、さらにドメイン X に逆変換される。この時、変換前後の画像をなるべく一致させることで、画像の内容を保つことができる。また、生成画像をドメイン Y の Discriminator で本物画像と判断されるように学習することで、よりドメイン Y に近い画像を生成することができる。

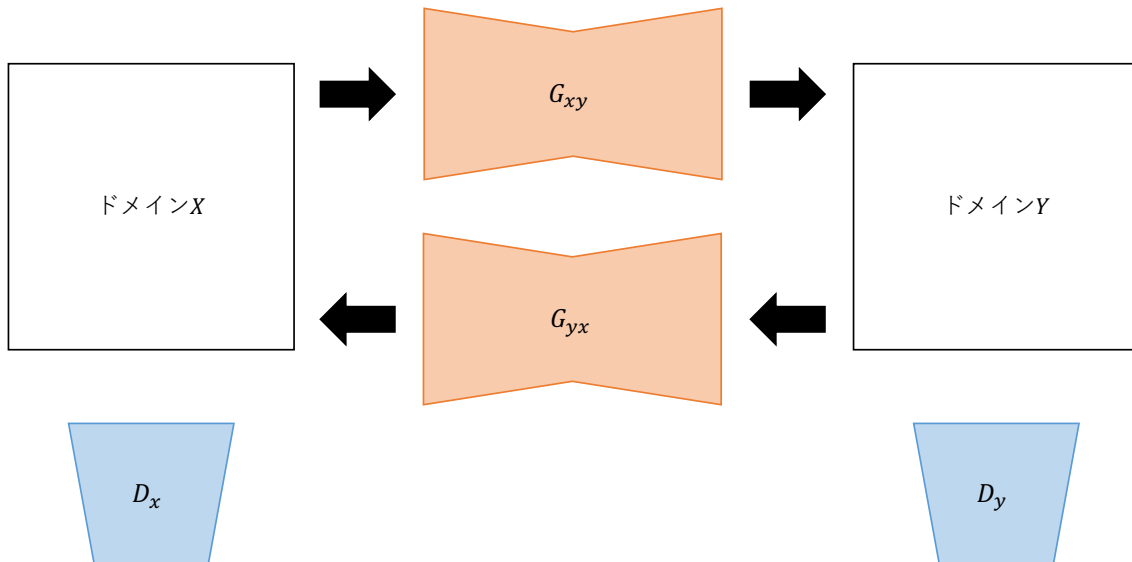


図 2.5: CycleGAN の構造

CycleGAN の登場により、教師なしの画像変換が可能になった。CycleGAN は、例えば馬の画像とシマウマの画像のような、2つのドメイン間の共通部分が多い場合に有効な手法である。一方で、形が大きく変わるような変換には向いていないことが知られている。加えて、CycleGAN は2つの Generator を同時に学習する必要があるが、学習がアンバラ

ンスになりやすく、生成画像の多様性が損なわれるモード崩壊が起こりやすい。これは、ドメイン X からドメイン Y へのマッピングと、ドメイン Y からドメイン X のマッピングという2つのタスクを同時に学習することが困難であることを示している。

2.2.3 StyleGAN2 を用いた教師なし画像変換

StyleGAN2[13] は2020年に発表された。Progressive Growing of GANs(PGGAN)[18] や StyleGAN[1] の影響を強く受けているため、ここでは順を追って PGGAN と StyleGAN について解説したのち、StyleGAN2 について述べる。

PGGAN

Progressive Growing of GANs(PGGAN)[18] は、NVIDIA が2017年に発表した GAN である。この GAN の最も大きな特徴は、GAN の学習方法を改善した点にある。図 2.6 にその概略を示す。これまでの GAN では、Generator と Discriminator の全ての畳み込みブロックを同時に学習していた。しかし PGGAN では、Generator と Discriminator の低解像度の畳み込みブロックのみを最初に学習する。ある程度学習が収束したら、次の畳み込みブロックも加えて学習を行う。このように、学習が進むにつれて、段階的に高解像度の畳み込みブロックを学習に追加してゆく。これは、最初から難易度の高い問題を解くのではなく、最初は簡単な問題から解き始めて、徐々に難しい問題へと移っていくカリキュラム学習に似ている。この学習方法は、学習の高速化・安定化に加えて、高品質の画像生成を可能にした。この学習手法は Progressive Growing と呼ばれる。

StyleGAN

StyleGAN[1] は、PGGAN の著者が2019年に発表したネットワークである。PGGAN をベースラインとし、主に GAN の Generator 部分に改良を加えることで、従来の GAN と比較して高性能化を達成した。StyleGAN の Generator は、Mapping network と Synthesis network の2つのネットワークから構成されている。全体図を図 2.7 に示す。

Mapping network は全結合層8層で構成されている。入力は512次元ベクトル、中間出力、最終的な出力は共に512次元ベクトルである。この層は、入力ノイズを非線形変換することで、潜在空間をときほぐし、特徴の抽出を容易にする役割を担う。

他方、Synthesis network は従来の Generator にあたる部分である。従来は入力ノイズを最も解像度の低いブロックに1度だけ入力していたが、StyleGAN では、Generator の畳み込みの後に毎回入力を行う。例えば Generator で 1024×1024 の解像度の画像を生成する場合には、計18回の入力を行う。そして Synthesis network への入力には、Mapping network の出力を用いる。Mapping network によって解きほぐされたノイズ w を、全結合層に通してスタイル特徴量を抽出。その値を用いて AdaIN を施すことで、入力ノイズか

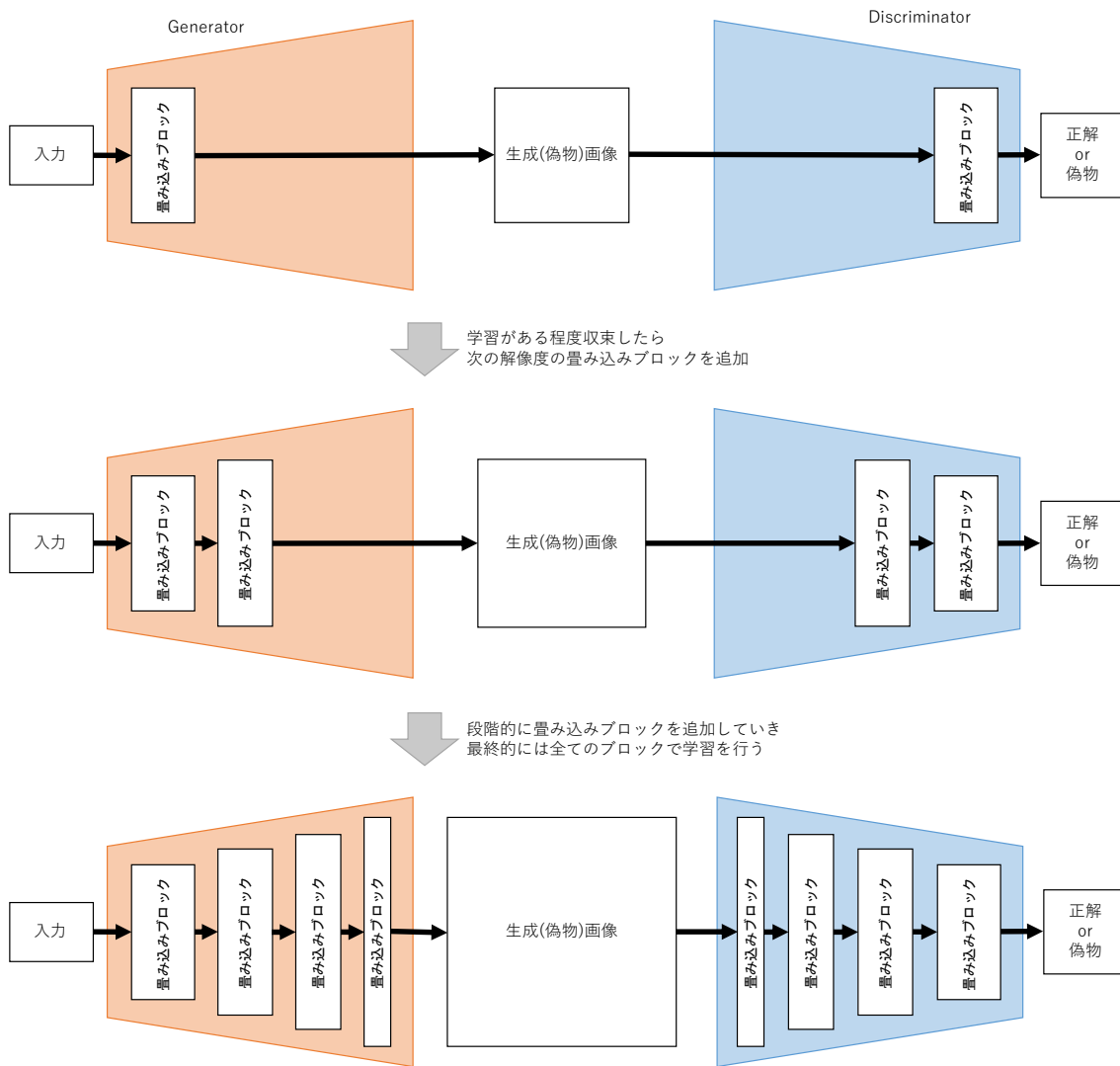


図 2.6: PGGAN の学習

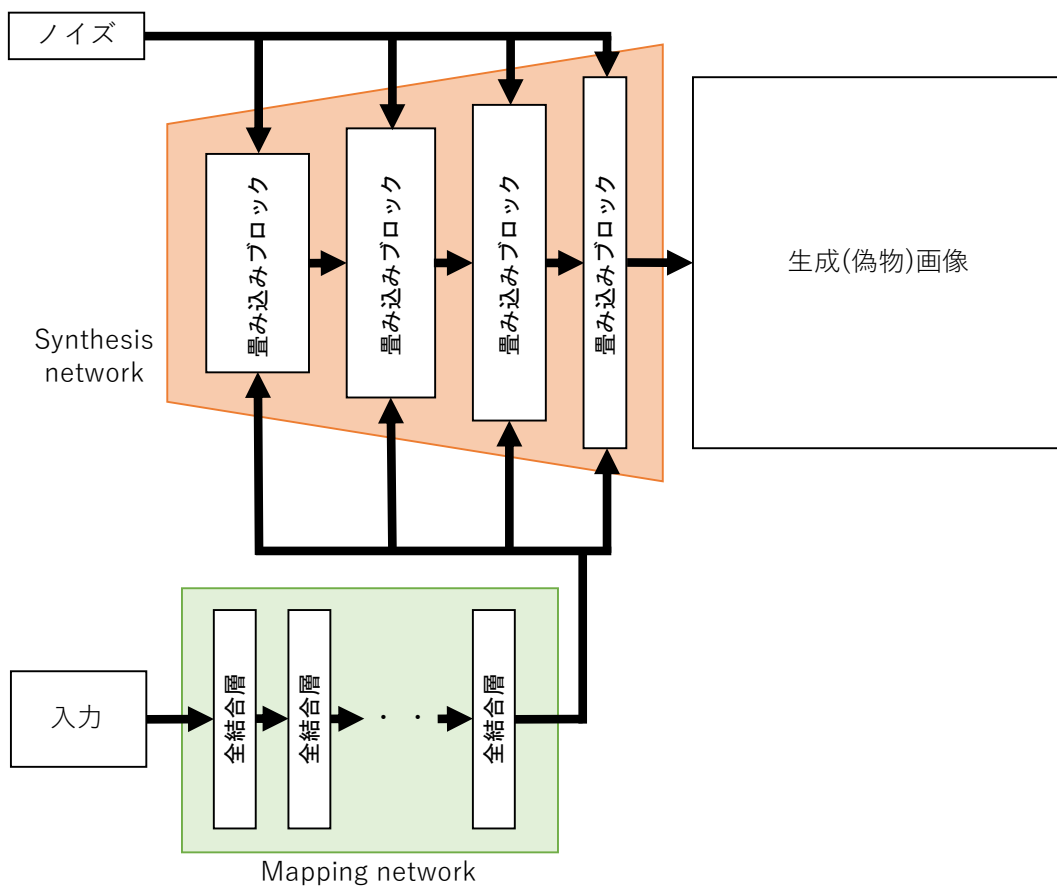


図 2.7: StyleGAN の Generator の全体図

らの特徴量を Synthesis network に入力することができる。各畳み込み層での入力を個別に変更することで、粗い特徴（顔の向き、顔の形）から細かい詳細（髪や肌の色）まで、解像度に応じた特徴量の供給を実現している。

スタイルミキシング

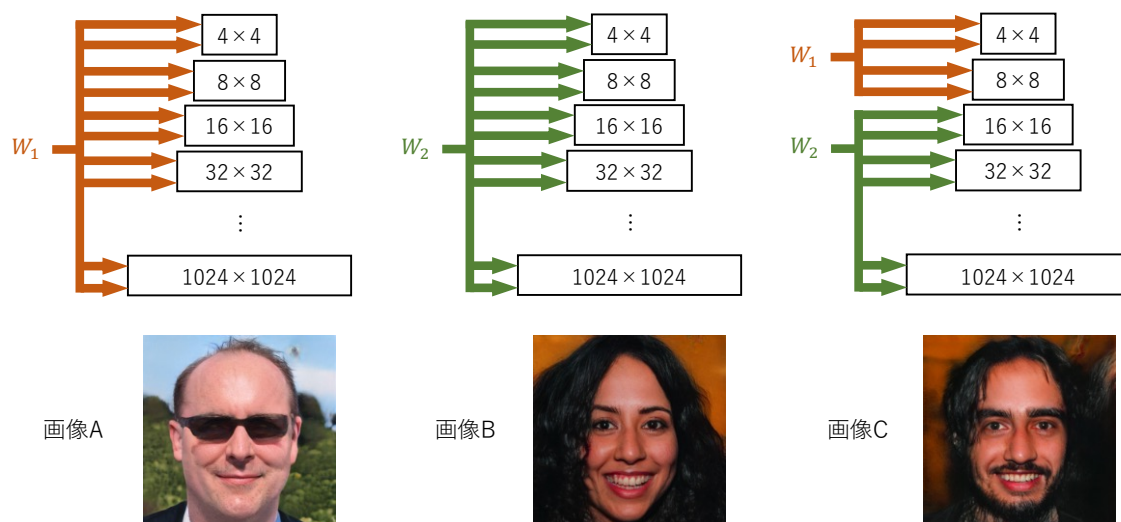


図 2.8: スタイルミキシング (論文 [1] 内の画像を引用して作成)

スタイルミキシングは、StyleGAN の論文 [1] 内で提案された教師なしの画像変換手法である。StyleGAN の登場によって、Generator の畳み込みブロックごとにスタイル情報を入力することが可能になった。スタイルミキシングは、この StyleGAN の特徴を利用している。スタイルミキシングの例を図 2.8 に示す。学習済みの StyleGAN に、潜在表現 W_1 を入力すると、画像 A が生成されたとする。同様に、潜在表現 W_2 を入力すると、画像 B が生成された。このとき、 4×4 と 8×8 のブロックに W_1 を、その他のブロックに W_2 を入力するとどのような画像が生成されるだろうか。低解像度側に画像 A、高解像度側に画像 B を表現するスタイル情報を入力すると、画像 A の構造を保ちつつ、画像 B のスタイルを取り入れた画像 C が生成される。ここでの画像の構造とは、被写体の形や大きさ、画像内での位置を意味する。StyleGAN の論文内では、スタイルミキシングを用いることで、異なる 2 人の顔画像を組み合わせたような画像を生成することができることを示している。

レイヤースワッピング

レイヤースワッピングは、2019 年に Pinkney らが発表した論文 [2] 内で提案した教師なし画像変換手法である。それぞれ異なるドメインで学習した 2 つの Generator の層を入れ替

えることで画像変換を行う。例えば、図 2.9 のように、ドメイン A を生成する GeneratorA と、ドメイン B を生成する GeneratorB が存在するとき、低解像度の畳み込みブロックは GeneratorA のものを、高解像度の畳み込みブロックは GeneratorB のものを用いることで画像変換を行う。この場合、画像全体の構造はドメイン A から引き継がれ、スタイル情報はドメイン B から得られる。これは、低解像度ブロックでは画像全体の構造を生成し、高解像度ブロックではより細かい部分を生成するという GAN の特徴を利用した手法である。この手法は、スタイルの変化に伴うコンテンツの位置や形の変化に比較的強いという特徴がある。一方で、生成画像の質は、どの場所の層を入れ替えるかによって変化するため、目的の画像変換結果を最も良く出力できるような層の構成を見つける必要がある。

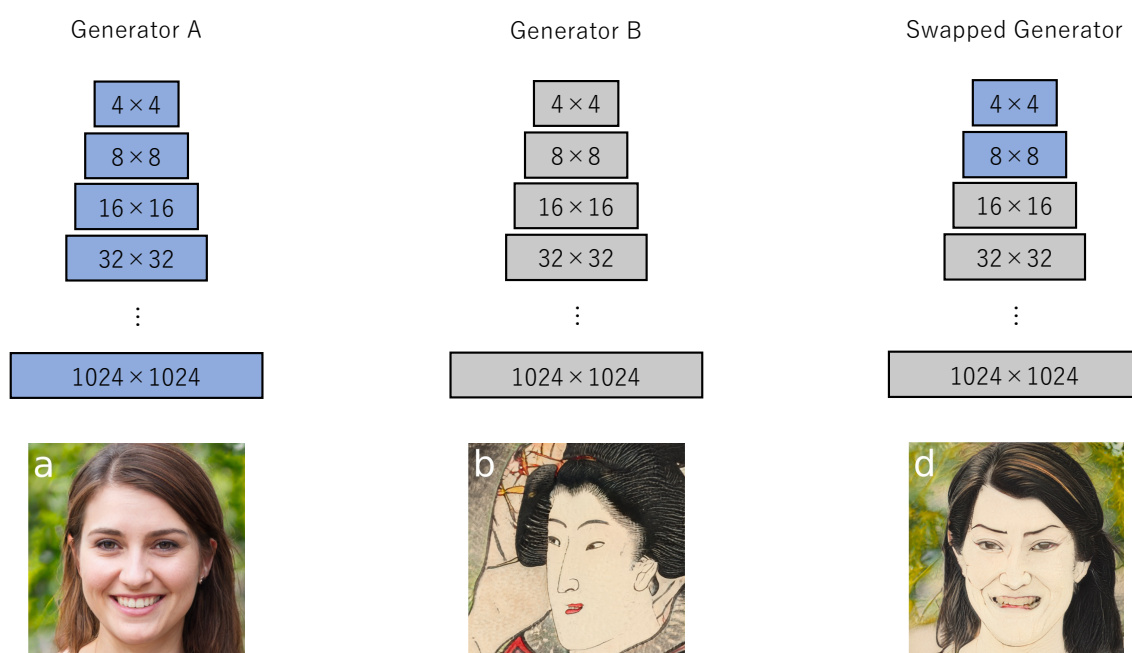


図 2.9: レイヤスワッピング (論文 [2] 内の画像を引用して作成)

StyleGAN2

StyleGAN2[13] は 2020 年に発表された。著者は前述の PGGAN や StyleGAN の著者と同じである。StyleGAN2 は StyleGAN をベースとしており、StyleGAN の Generator 部分の構造や損失関数に改良を加えることで、StyleGAN で発生していた問題を解消した。

StyleGAN では、AdaIN と呼ばれる手法を用いてスタイル特徴量を入力していたが、これは局所的なノイズの原因となっていた。そのため、AdaIN を用いるのではなく、畳み込みの重みに対してスタイル特徴量の入力と正規化を行うことで、局所的なノイズを減らしている。また、ベースラインの StyleGAN では、段階的に高解像度の畳み込みブロックを追加する Progressive Growing を採用していたが、これは、出力画像の多様性を低下さ

せる原因となっていた。そのため、この Progressive Growing を取りやめ、最初から全ての畳み込みブロックを学習するように変更した。その代わりに、Generator の出力は最終層の出力のみではなく、各解像度のブロックの出力を足し合わせることにした。それにより自然な画像が生成できるようになった。損失関数にも変更を加えている。StyleGAN の評価を行うために導入した評価指標である Perceptual Path Length(PPL, 知覚的経路長) を損失関数に加えることで、生成画像の品質が向上した。

StyleGAN2 を用いた教師なし画像変換

2022 年、Jialu Huang らが StyleGAN2 を用いた教師なしの画像変換手法を発表した [3]。StyleGAN2 を用いて、主にファインチューニングとレイヤースワッピングを用いたスタイル変換を行う。この手法は、GAN の中でも優れた性能を示している StyleGAN2 を用いているため、生成画像の質が高い。加えて、レイヤースワッピングを用いているため、スタイル変換時の位置や形の変化に強い。本研究では、この手法をベースに実験を行った。

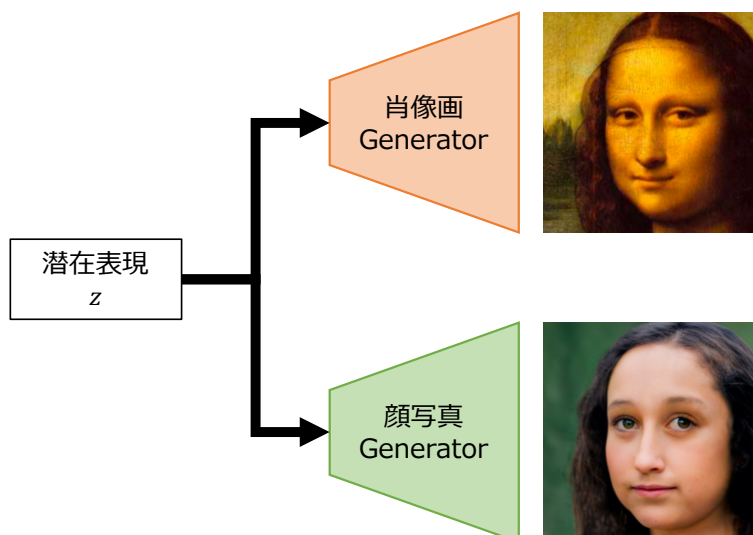


図 2.10: StyleGAN2 を用いた教師なし画像変換の概要 (論文 [3] 内の画像を引用して作成)

この手法では、同一のコンテンツを表現することのできる潜在表現が存在すると仮定している。その潜在表現は、どんな Generator に入力しても同一のコンテンツを生成できる。概要を図 2.10 に示す。例えば、一つの潜在表現を肖像画 Generator に入力するとモナリザの肖像画が出力されるとする。このとき、その潜在表現を顔写真 Generator に入力するとモナリザを写真風にした画像を生成する。このように、一つの潜在表現を別の Generator に入力しても、アイデンティティを保持した画像が出力されるような仕組みを考える。この仕組みを実現するため、それぞれの Generator を別々に学習するのではなく、一つの Generator を元に別の Generator をファインチューニングする手法を用いる。こうすることで、二つの Generator に関連性を持たせることができる。

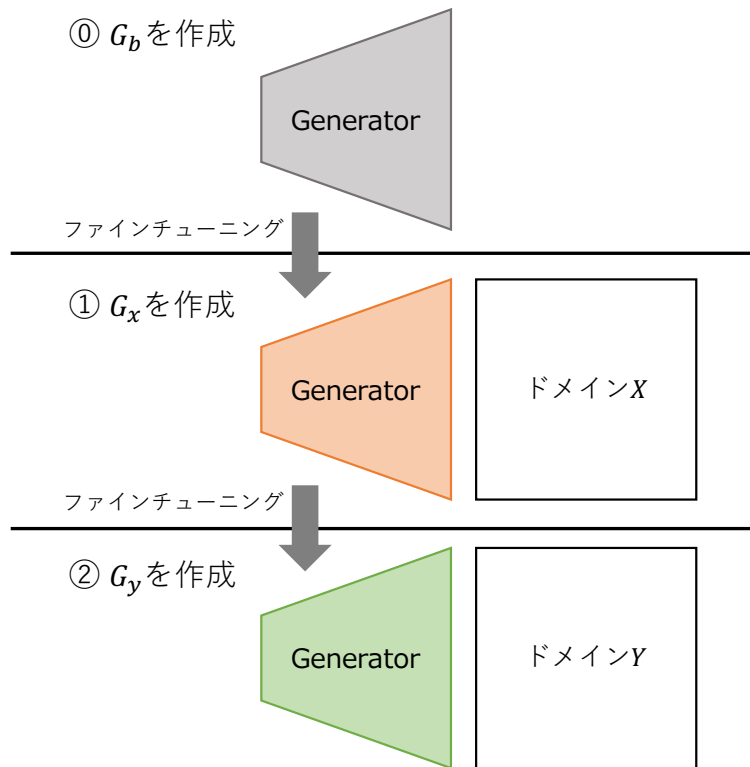


図 2.11: StyleGAN2 を用いた教師なし画像変換の学習

ドメイン X からドメイン Y への画像変換を行う場合の学習時の手順を図 2.11 に示す。まず、ベースとなる Generator として G_b を準備する。 G_b は、顔写真 70000 枚からなる FFHQ データセット [1] で事前学習済みの StyleGAN2 である。次に、 G_b に対してソースドメインのデータセットを用いてファインチューニングを行い、完成した Generator を G_x とする。ただし、ソースドメインが顔写真である場合は、ファインチューニングを行わず、 G_b をそのまま G_x として良い。そして、作成した G_x をベースとして再度ファインチューニングを行う。ターゲットドメインのデータセットを用いる。完成した Generator を G_y とする。一連のファインチューニングにより、ソースドメイン側の Generator である G_x と、ターゲットドメイン側の Generator である G_y が完成した。

推論時の手順を図 2.12 に示す。まず、 G_x と変換したい画像 x から入力 z を見つける。StyleGAN の出力から入力を求めるこの手法は GAN Inversion と呼ばれる。GAN Inversion にはいくつかの手法があるが、Jialu Huang らの論文 [3] では最適化手法を用いている。Inversion によって z を見つけた後、 z を G_y に入力し、画像 y を生成する。このとき、 G_y に対してレイヤスワッピングを行う。 G_y の低解像度側の畳み込みブロックを、 G_x のものに置き換える。そうすることで、ドメイン X のコンテンツを保ちつつ、ドメイン Y のスタイルを取り入れた画像を高精度で生成することができる。

この手法は、教師なし学習である。ドメイン X の Generator を学習する際、ドメイン Y のデータは必要ない。同様に、ドメイン Y の Generator を学習する際、ドメイン X の

③ G_x と変換したい画像 x から入力 z を求める

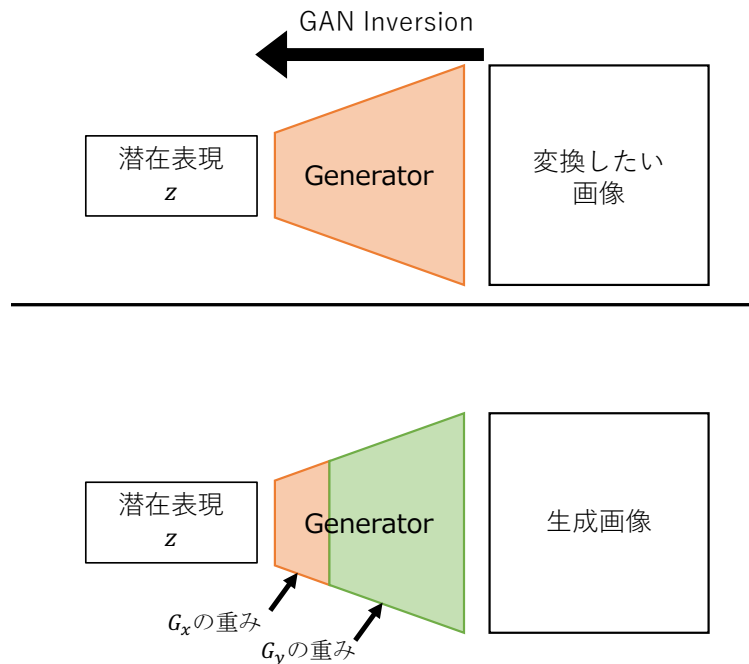


図 2.12: StyleGAN2 を用いた教師なし画像変換の推論

データは必要ない。そのため、ペアデータセットを必要としない。また、CycleGAN のようなサイクル構造を持たないため、複数の Generator を同時に学習する必要がない。1 回の学習で単一の Generator の学習に注力できるため、CycleGAN よりも高精度な画像生成が実現できる。加えて、入力に用いる潜在表現にノイズを混ぜることで、容易にマルチモーダル変換への拡張を行うことができる。

2.2.4 pixel2Style2pixel

GAN Inversion に関する研究は数多くあるが、その手法は基本的に 3 つに大別される。最適化手法、エンコーダーを作成する手法、そして最適化手法とエンコーダーを組み合わせた手法である。最適化手法は最も古典的な手法である。推定に時間がかかるが、精度は高い。エンコーダーを作成する手法は、推定に時間がかからない一方で精度は低い。その 2 つを組み合わせた手法は、多少時間がかかるが、精度は悪くない。

2021 年に発表された pixel2Style2pixel(pSp)[19] は、画像の潜在表現を画像から直接推論するエンコーダーを用いる。全体像を図 2.13 に示す。エンコーダーと StyleGAN を直列に並べる。エンコーダーは画像を入力すると潜在表現を出力する。StyleGAN は潜在表現を入力すると画像を生成する。StyleGAN には学習後のものを用意し、重みを固定する。入力画像と出力画像が一致するように学習を行う。こうすることで、エンコーダーは、入

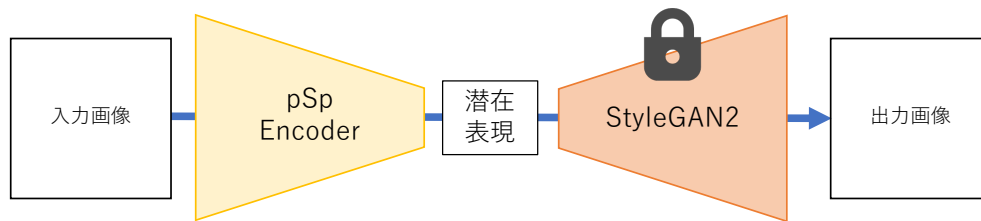


図 2.13: pSp の全体像

力画像の情報からその画像に対応する潜在表現を出力するようになる。pSpを用いることで、画像から画像への変換を高速で行うことができる。

2.3 拡散モデルを用いた画像変換

2022年12月、AI Picasso 株式会社は、いらすとや風のイラストを生成する「AI いらすとや」をリリースした [20]。これは、拡散モデル [21] で学習済みのモデルをベースラインとし、いらすとやのデータセットを用いてファインチューニングを行うことで、従来よりも多様な画像生成を実現したモデルである [22]。拡散モデルを用いることで、人間の顔だけでなく、例えば建物や飲食物など、被写体を問わず様々な写真をいらすとや風に変換することができる。しかし、大きく3つの問題がある。一つ目は、全体的に出力画像の品質が低いことである。入力によっては、画像をうまく生成できない場合が多く発生する。機械学習による画像生成の問題点として、画像の品質と多様性はトレードオフの関係であることが指摘されてきた。拡散モデルはGANに比べて多様な画像を生成可能である。しかしその品質には改善の余地がある。二つ目は、推論に時間がかかるということである。拡散モデルは、入力した情報から画像を生成する際、反復処理を行いながら少しずつ画像を生成していく必要がある。そのため、GANと比較すると画像を生成するのに時間がかかる。三つ目は、入力としてテキストを与える必要があるということである。一般に、顔写真の特徴を捉えた画風変換を行う場合には、対象の顔写真を直接入力した方が簡単である。テキストで入力するためには、前処理として対象の顔写真をテキストに変換する必要がある。しかしそれが、学習時間の長期化や、出力画像の品質の低下を招く可能性がある。

本研究は、変換の対象を顔写真に限定し、顔写真から顔イラスト、すなわち画像から画像への画風変換について扱ったため、拡散モデルは用いていない。

第3章 提案手法

本研究では、StyleGAN2を用いた教師なしの画像変換の手法 [3](以下、既存手法と呼ぶ)をベースとして実験を行う。既存手法は複数の Generator を同時に学習する必要がないため、各 Generator が1度に1つずつ学習する。そのため、Generator は自身の学習により注力可能となった。加えて、ファインチューニングとレイヤースワッピングにより、2つの Generator に関連性を持たせつつも自由度の高さを確保した。それらの理由から、高精度な画像変換が実現した。しかし、この既存手法には欠点も存在していた。推論時に時間がかかるという問題である。既存手法では、変換したい画像に対応する潜在表現を求める処理に1分以上を要していた。これは、潜在表現を求めるには学習済みの Generator の重みを固定し、最適化手法を用いて1000回の反復処理を行う必要があったからである。本研究では、既存手法に pSp のエンコーダーを導入することで、高速で高精度な画像変換を行う手法を提案する。提案手法は既存手法と遜色ない精度を保ちつつも、ほぼリアルタイムでの画像変換を実現した。提案手法の手順を図 3.1 に示す。提案手法は3ステップの学習からなる。既存手法と同様に G_x の作成、 G_y の作成を行ったのち、pSp のエンコーダーの学習を行う。pSp のエンコーダーの学習では、重みを固定した G_x を用いる。入力画像と出力画像が一致するように学習を行うことで、エンコーダーは入力画像から潜在表現を推定するようになる。推論時は、pSp のエンコーダーと、レイヤースワッピングを行った G_y を重みを固定して用いる。変換したい画像をエンコーダーに入力し、得られた潜在表現を G_y に入力することで、画像変換を行う。画像の潜在表現をエンコーダーによって1回で推論するため、画像ごとに1000回の反復処理を行う既存手法と比較して、推論時間の大幅な短縮が可能となった。

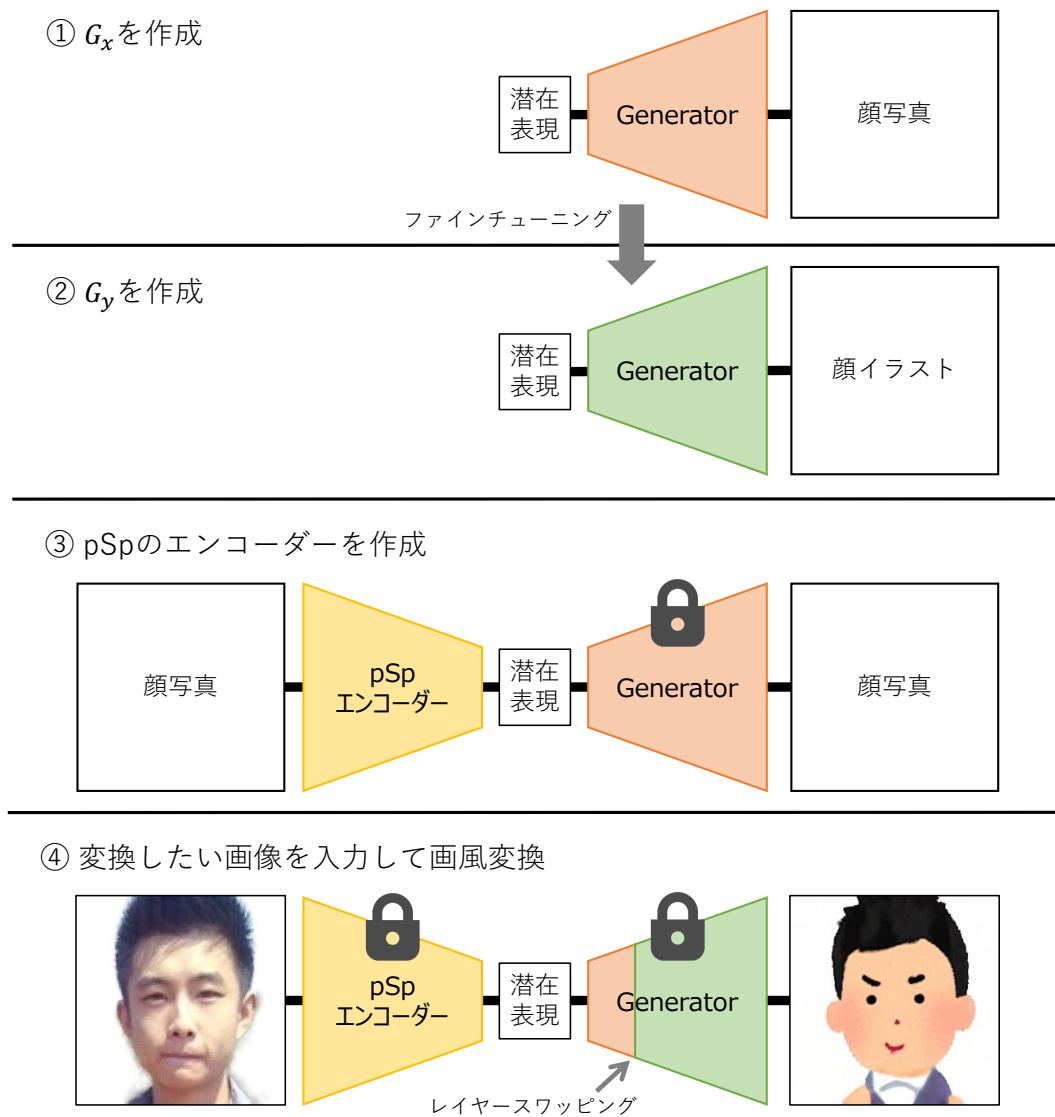


図 3.1: 提案手法の手順

第4章 実験

4.1 実験の詳細

既存手法では、ファインチューニングとレイヤースワッピングが画像変換の重要な役割を担っている。本研究では、今回のデータセットにおけるそれらの手法の検証と、提案手法の評価のための実験を行った。

Generatorとして用いたのは、StyleGAN2である。ただし、後述する顔イラストデータセットの画像サイズを考慮し、解像度が 4×4 から 256×256 のブロックまでを使用した。解像度が 512×512 と 1024×1024 のブロックは使用していない。Mapping networkは全結合層8層とした。実験に用いたStyleGAN2の構造を図4.1に示す。

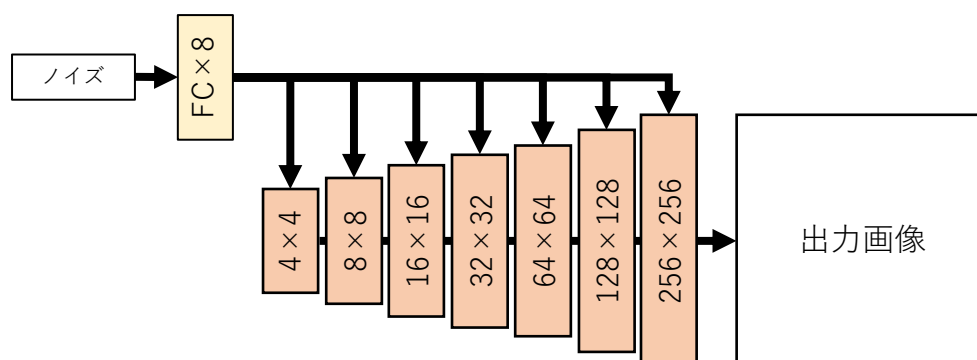


図 4.1: 実験に用いた StyleGAN2 の構造

ソースドメイン側の Generator(G_x)は、ベースとなる Generator を元にファインチューニングを行うことで作成した。効果的な G_x の作成については、4.3.1 節で検証する。ターゲットドメイン側の Generator(G_y)は、後述する顔イラストデータセットを用いて、 G_x をファインチューニングすることによって作成した。

G_x や G_y のファインチューニングでは、StyleGAN2 の Mapping network の重みは固定し、学習対象から外した。イテレーションは 12000、バッチサイズは 8 で統一した。

最適化手法は、学習率は 0.1、イテレーションは 1000 とした。optimizer には Adam を使用した。

pSp のエンコーダーには、ResNet50 ベースのエンコーダーを用いた。エンコーダーの学習において、バッチサイズは 8、イテレーションは 300000、学習率は 0.0001 とした。optimizer には Ranger を使用した。pSp のエンコーダーを導入することにより、推論時間

の大幅な短縮が期待される。4.3.2節では、エンコーダーの導入による効果と、出力画像の品質について、既存手法と比較して検証する。最適化手法、pSpのエンコーダー共に、推論を行ったのはStyleGAN2の512次元の w 空間である。

学習後の推論には、後述するAll-Age-Facesデータセットのテスト用画像を用いた。イラスト風画像の生成に関して、特に記述がない場合は1対1変換の結果を掲載している。最適化手法もしくはpSpのエンコーダーによって推論された潜在表現をレイヤースワッピング後の G_y に入力し、画風変換を行った。レイヤースワッピングは、デフォルトで 4×4 と 8×8 のブロックをソースドメイン側のGeneratorの重みに入れ替えた。この妥当性については、4.3.3節で検証する。既存手法では、得られた潜在表現をStyleGAN2に入力する際、 16×16 のブロック以降に入力する潜在表現に対して、潜在表現全体の平均値に寄せるという操作を行っている。この操作を行うことで、出力画像の安定性が増す。本研究でも同様の操作を行っている。

4.2 データセット

本研究で使用したデータセットは、人物の顔写真のデータセットと、人物の顔のイラスト画像のデータセットである。

4.2.1 顔写真データセット

人物の顔写真のデータセットとして、2種類を使用した。

FFHQ データセット

Flickr-Faces-HQ(FFHQ)データセットは、StyleGANの論文[1]で発表された顔写真データセットである。FFHQは、画像70000枚からなり、年齢、民族、画像の背景などのバリエーションが豊富である。写真共有のためのコミュニティウェブサイトであるFlickrから収集され、アルゴリズムによって顔の検出と切り抜きが行われた。許可された画像のみで構成されている。各画像の解像度は 1024×1024 であるが、本研究では 256×256 にリサイズして使用した。FFHQデータセットの一部を図4.2に示す。

All-Age-Faces データセット

All-Age-Faces(AAF)データセット[23]はアジア人を中心とした2歳~80歳の13322枚の顔写真で構成されているデータセットである。本研究で使用するイラストデータセットは日本人を描いた画像が多いため、顔写真データセットとしてアジア人を中心とした本データセットを採用した。顔の検出と切り抜きを行い、 256×256 ピクセルに揃えた。顔が検出できた13306枚の画像のうち、ランダムに選んだ15枚をテスト用画像データセッ



図 4.2: FFHQ データセットの一部

ト, 残った 13291 枚を訓練用画像データセットとした. 事前学習済みの IndexNet[24] を用いて, 背景を削除した. データセットの一部を図 4.3 に示す.



図 4.3: All-Age-Faces データセットの一部

4.2.2 顔イラストデータセット

いらすとや [25] は, 数多くのイラストを無料で提供しているサイトである. イラストレーターのみふねたかし氏が運営している. 2012年1月から9年間に渡り毎日イラストが追加されてきたが, 2021年2月以降は不定期でイラストが追加されている. 様々なイラストがワンストップで入手でき, イラストの画風に統一感があるため利便性が高い. 加えて, 商用, 非商用問わず無料で利用できることから, 公的機関などでも使用されている. このサイトには膨大な数のイラストが存在する. 描かれている人物も, 人種, 性別, 年齢によって様々なバリエーションが存在する. また, 同じようなシチュエーションのイラストでも, 人物は少しずつ異なる特徴を持つように描かれている場合が多い. 本研究では, このサイトから 26071 枚の画像を収集した. OpenCV[26] を用いて収集した画像から顔の検出と切り抜きを行い, 256×256 ピクセルに揃えた. 人物のイラストとして収集したのは 10823 枚である. データセットの一部を図 4.4 に示す.



図 4.4: 人物の顔イラストのデータセットの一部

4.3 実験

4.3.1 データセットに関する実験

この節では、顔写真をいらすとや風に変換するのに適したソースドメイン側の Generator (G_x) の作成手法を探るための実験を行った。

実験

G_x の作成に最も適したデータセットとして、以下の3つのデータセットを候補とした。1つ目は Flickr-Faces-HQ (FFHQ) データセット [1], 2つ目は All-Age-Faces (AAF) データセット [23], 3つ目は背景削除処理を行った AAF データセットである。

学習済みモデルとして、FFHQ データセットで 550000 イテレーションの学習を行ったモデル [3] を用いた。これを G_b と表す。以下の3つの実験を行い、結果を比較した。

実験 1-1

G_b をそのまま G_x とした。

実験 1-2

G_b に対して、AAF データセットを用いてファインチューニングを行い、 G_x とした。

実験 1-3

G_b に対して、事前に背景削除処理を施した AAF データセットを用いてファインチューニングを行い、 G_x とした。

Inversion 結果

まず、それぞれの G_x に対して、最適化手法を用いて GAN Inversion の結果を比較した。テスト用画像それぞれに対して、最適化手法により潜在表現を求めた。求めた潜在表現を

G_x に入力することで、元の顔写真をどの程度復元できるか検証した。目標画像に近い画像が生成されるほど精度が高い結果であると言える。結果を図 4.5 に示す。最も上の行が入力画像であり、次の行からそれぞれ実験 1-1、実験 1-2、実験 1-3 の画像の復元結果である。各列が同じ目標画像に対応している。



図 4.5: データセットに関する実験の Inversion 結果

実験 1-1 では、特に目元の復元が難しいことが分かった。これは、FFHQ データセットに欧州の顔画像が比較的多く含まれているためであると考えられる。アジア人の顔写真が比較的多く含まれている AAF データセットを用いた実験 1-2 では、実験 1-1 と比較して目元の復元の精度が高かった。実験 1-3 は、実験 1-2 と比較して、全体的に復元精度が高まった。実験 1-3 は、画像の背景を削除したデータセットで学習した。それにより、Generator は顔と背景の関係を学習する必要がなくなり、顔の部分に集中して学習を進めることができたのではないと思われる。

画風変換結果

次に、それぞれの G_x に対してファインチューニングを用いて G_y の作成を行い、画風変換の結果を比較した。結果を図 4.6 に示す。一番上の行が入力画像である。各列がそれぞれの入力画像に対応している。この結果から、3つの実験のうち、実験 1-3 が最も良い結果を出力できていることがわかる。実験 1-1 は、どの生成画像も顎のあたりに濃い肌色のエリアが存在する。これには、2つの原因がある。一つは、顔写真データセットと、顔イラストデータセットの口の位置の差異が大きいこと。二つ目は、いらすとやの顔イラストは、鼻が極端に小さく描かれることである。顔イラスト Generator は、顔写真 Generator

をファインチューニングすることで学習を行う。顔写真 Generator では鼻を生成していたはずの位置に、いらすとやの極端に小さい鼻と大きな口がある。そのため、結果的に、写真の鼻の特徴量から、イラストの大きな口を生成するようになってしまう。実験 1-2 は、この問題を多少は吸収できている。西欧の顔よりもアジアの顔の方がいらすとやの一般的な顔に近いのではないかとと思われる。しかし実験 1-2 では、背景にノイズが発生した。これは背景のある画像で学習していたため、Generator が顔の部分と背景の部分の関係を学習してしまった結果だと考えられる。実験 1-3 は、顔写真データセットの背景を削除したため、どんな背景であっても関係なく顔画像の生成に専念した学習ができていると思われる。そのため、学習後の Generator に背景画像付きの画像を入力しても、ノイズの少ない画像を生成できる。実験 1-3 が最も良い結果となった。

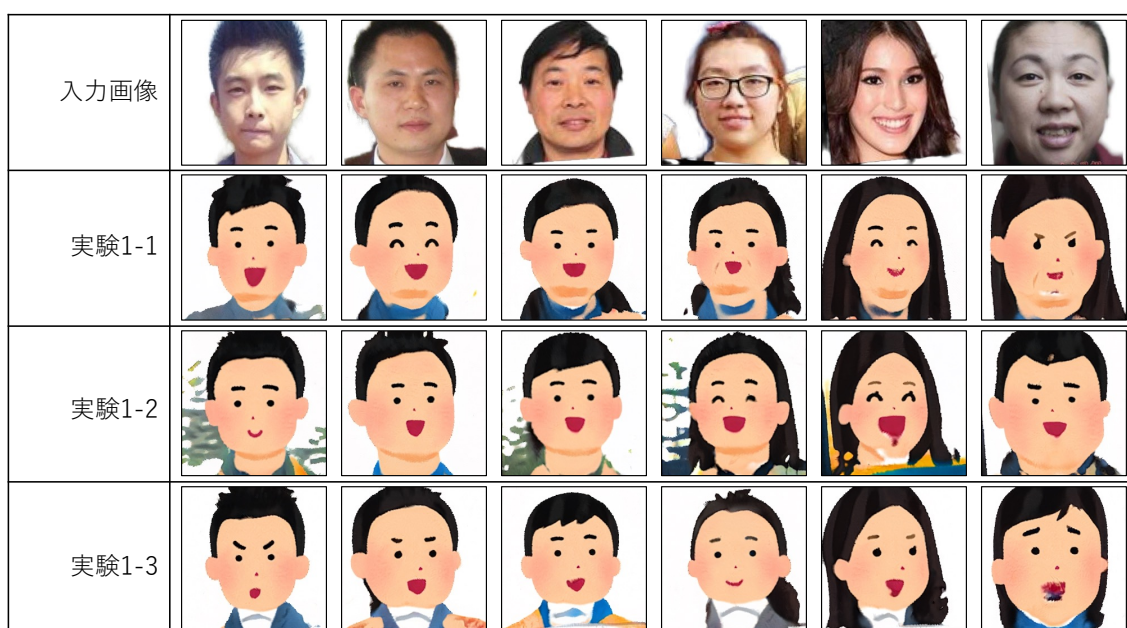


図 4.6: データセットに関する実験の画像変換結果

4.3.2 pSp に関する実験

この節では、前節の実験 1-3 をベースラインとし、その手法に pixel2Style2pixel(pSp)[19] を導入した場合の効果について評価を行う。既存手法と同様に G_x の作成、 G_y の作成を行ったのち、pSp のエンコーダーの学習を行った。その後、pSp のエンコーダーと G_y を用いて推論を行った。実行時間と生成画像の品質について、既存手法と比較する。

推論に要した時間は 0.01 秒であった。既存手法では、Inversion に 75 秒程度が必要だったため、提案手法は推論時間を大幅に短縮した。pSp を用いた実験結果を図 4.7 に示す。一番上の行から、入力画像、Inversion 画像、出力画像である。Inversion 画像は、学習済みの pSp エンコーダーによって推定された潜在表現を顔写真 Generator に入力して得られ

た画像である。入力画像に近ければ近いほどよい。出力画像は、推定された潜在表現をレイヤースワッピング後の顔イラスト Generator に入力して得られた画像である。Inversion 画像は、既存手法と比較して遜色ない結果となった。右から3枚目の画像はメガネの復元ができていているという点で、実験 1-3 よりも復元精度が向上した。一方で、画像によっては既存手法の方が結果が良くなる場合もあった。例えば、最も右の画像に関しては実験 1-3 による Inversion 結果の方がより元画像を復元できている。次に、画像変換の結果について比較する。pSp を導入した場合の変換結果は、時々ノイズが生成される場合があるものの、全体的に見れば実験 1-3 による結果と比較して遜色ない結果となった。画像によっては、pSp を導入した場合の方が良い結果になった。

この実験により、提案手法は生成画像の品質をほとんど保持しつつ、推論時間を大幅に短縮できることが分かった。



図 4.7: pSp を用いた実験結果

4.3.3 レイヤースワッピングに関する実験

次に、レイヤースワッピングに関する実験を行った。推論時の Generator は基本的には G_y を用いるが、 G_y の低解像度のブロックのいくつかは、 G_x の重みを利用する。これがレイヤースワッピングである。レイヤースワッピングを用いた場合、生成画像の質はどの場所の層を入れ替えるかによって大きく変化するため、目的の画像変換結果を最も良く出力できるような層の構成を見つける必要がある。ここでは、どのブロックを G_x と入れ替えると結果が良くなるのかについて検討を行った。以下の5つの実験を行い、結果を比較した。

実験 3-1

レイヤースワッピングを行わず、 G_y をそのまま使用した。

実験 3-2

G_y の 4×4 のブロックのみを G_x の 4×4 のブロックに置き換えた。

実験 3-3

G_y の 4×4 から 8×8 のブロックまでを G_x の当該ブロックに置き換えた。

実験 3-4

G_y の 4×4 から 16×16 のブロックまでを G_x の当該ブロックと置き換えた。

実験 3-5

G_y の 4×4 から 32×32 のブロックまでを G_x の当該ブロックと置き換えた。

実験結果を、図 4.8 に示す。レイヤースワッピングを行わなかった実験 3-1 では、生成画像の品質は安定しているが、入力画像の特徴の反映が弱い。実験 3-2 や実験 3-3 は、入力画像の特徴をより反映できており、ノイズも少ない。実験 3-2 と実験 3-3 を比較すると、実験 3-3 の方が、より入力画像の特徴を捉えた出力ができています。実験 3-4 や実験 3-5 は、生成画像がイラスト風から離れてしまい、品質が低下した。これらの実験により、実験 3-3 の結果が最も良いことが分かった。入力画像の特徴を捉えながらも生成画像のノイズが少ないのは、 4×4 のブロックと 8×8 のブロックのみを置き換える場合である。




























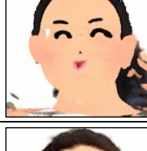
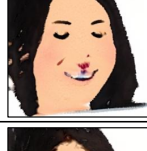
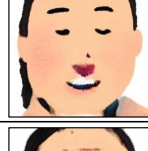






入力画像						
実験3-1						
実験3-2						
実験3-3						
実験3-4						
実験3-5						

図 4.8: レイヤースワッピングの実験結果

第5章 まとめ

本研究では，GANを用いて，人物の顔写真をイラスト風に変換した．イラストデータセットには，いらすとやを用いた．いらすとやで描かれている人物の顔イラストは顔写真との変化が大きいので，いらすとや風への変換は比較的難易度が高い．本研究では，GANの中でも特に高性能なStyleGAN2をベースとし，教師なし画像変換を行った．実験によって，本研究の画風変換に対して，ファインチューニングとレイヤースワッピングの最も良い設定を明らかにした．加えて，StyleGAN2に特化したpSpのエンコーダーを組み合わせることで，短時間で高品質な画像変換を行う手法を提案した．この手法により，既存手法による画風変換と比較して精度を維持しつつ，ほぼリアルタイムでの画風変換が可能となった．

一方で，課題もある．現在の手法では，メガネなどの装飾品を含む顔写真を入力した場合，変換後の画像では装飾品を生成することができない．生成画像の品質を維持しつつも，装飾品を含めた入力画像の特徴を反映できるようになると，より画風変換のレベルを上げることができるであろう．

謝辞

本研究においては、ご多忙の中、熱心に指導をしてくださった伊藤靖朗教授、中野浩嗣教授、高藤大介助教、亀井清華准教授に深く感謝致します。研究を進めるにあたり、とても貴重なご意見をいただきました。

また、研究しやすい環境を作ってくくださった研究室の皆様にも、心より感謝しています。

参考文献

- [1] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.
- [2] J. N. Pinkney and D. Adler, “Resolution Dependent GAN Interpolation for Controllable Image Synthesis Between Domains,” *arXiv preprint arXiv:2010.05334*, 2020.
- [3] J. Huang, J. Liao, and S. Kwong, “Unsupervised Image-to-Image Translation via Pre-Trained StyleGAN2 Network,” *IEEE Transactions on Multimedia*, vol. 24, pp. 1435–1448, 2022.
- [4] R. Yi, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, “APDrawingGAN: Generating Artistic Portrait Drawings From Face Photos With Hierarchical GANs,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 10 743–10 752.
- [5] R. Yi, M. Xia, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, “Line Drawings for Face Portraits From Photos Using Global and Local Structure Based GANs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3462–3475, 2021.
- [6] G. Song, L. Luo, J. Liu, W.-C. Ma, C. Lai, C. Zheng, and T.-J. Cham, “AgileGAN: Stylizing Portraits by Inversion-Consistent Transfer Learning,” *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–13, 2021.
- [7] J. Back, “Fine-Tuning StyleGAN2 For Cartoon Face Generation,” *CoRR*, vol. abs/2106.12445, 2021. [Online]. Available: <https://arxiv.org/abs/2106.12445>
- [8] M. J. Chong and D. Forsyth, “JoJoGAN: One Shot Face Stylization,” in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 128–152.
- [9] R. Wu, X. Gu, X. Tao, X. Shen, Y. Tai, and J. Jia, “Landmark Assisted CycleGAN for Cartoon Face Generation,” *CoRR*, vol. abs/1907.01424, 2019. [Online]. Available: <http://arxiv.org/abs/1907.01424>

- [10] J. Kim, M. Kim, H. Kang, and K. Lee, “U-GAT-IT: Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization for Image-to-Image Translation,” *arXiv preprint arXiv:1907.10830*, 2019.
- [11] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image Style Transfer Using Convolutional Neural Networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.
- [12] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2223–2232.
- [13] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and Improving the Image Quality of StyleGAN,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020, pp. 8110–8119.
- [14] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *International Conference on Learning Representations*, 2015.
- [15] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual Losses for Real-Time Style Transfer and Super-Resolution,” in *European conference on computer vision*. Springer, 2016, pp. 694–711.
- [16] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Nets,” *stat*, vol. 1050, p. 10, 2014.
- [17] 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 (CVPR)*, July 2017, pp. 1125–1134.
- [18] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive Growing of GANs for Improved Quality, Stability, and Variation,” in *International Conference on Learning Representations*, 2018.
- [19] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, “Encoding in Style: A StyleGAN Encoder for Image-to-Image Translation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 2287–2296.

- [20] “いらすとや風のイラストを生成する「AIいらすとや」をリリース | AI Picasso 株式会社のプレスリリース,” 12 2022. [Online]. Available: <https://prtmes.jp/main/html/rd/p/000000002.000113219.html>
- [21] J. Ho, A. Jain, and P. Abbeel, “Denoising Diffusion Probabilistic Models,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.
- [22] Yasunori Ozaki, “Diffusion Model Specializing In An Illustrator,” **電子情報通信学会技術研究報告**, vol. 122, no. 181, pp. 61–66, 2022.
- [23] J. Cheng, Y. Li, J. Wang, L. Yu, and S. Wang, “Exploiting Effective Facial Patches for Robust Gender Recognition,” *Tsinghua Science and Technology*, vol. 24, no. 3, pp. 333–345, 2019.
- [24] H. Lu, Y. Dai, C. Shen, and S. Xu, “Indices Matter: Learning to Index for Deep Image Matting,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019, pp. 3266–3275.
- [25] “かわいいフリー素材集 いらすとや.” [Online]. Available: <https://www.irasutoya.com/>
- [26] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.