GAN による特徴を持つ画像の生成法の改良

2019SS021 市川翔大　2019SS068 肖銘睿
指導教員：小市俊悟

1　はじめに

本研究では，機械学習，特に敵対的生成ネットワーク（GAN）[1] と呼ばれる手法を用いて，指定された複数の特徴をもつ画像を生成する方法の改良を行う．昨今，ＧＡＮに関する研究の進歩には目覚ましいものがあり，本物の画像と遜色ない画像を生成することができるようになった．しかし，まだまだ改善が望まれる機能もある．

例えば，顔画像は複数の特徴を持つことが一般的であるが，指定されたすべての特徴をもつ顔画像データが少ないため，GAN であっても十分な学習が行えず，複数の特徴を持つ画像を生成することは容易ではない．そこで本研究では，同様の問題に取り組んでいた卒業研究 [7] を引き継ぎ，GAN に関する最近の結果も取り入れて，より良い生成法の開発を目指す．新しく取り入れようと考えているアイディアの一つは，学習中に生成された画像を学習データに追加し，より多様性のある画像を生成するというものである．また，そのほかにも，これまでの既存の卒業研究では，低解像度やノイズ画像に対して，特徴を備えていないことの識別が不十分であったと考えられることから，それらの画像に対しても確実に識別できるように，学習方法の改善も試みる．

2　GAN の説明

本研究では，敵対的生成ネットワークや，英語名で，GAN(Generative Adversarial Networks) と呼ばれる機械学習の手法を用いるので，GAN について，まず説明する．

GAN は，2014 年にモントリオール大学の Ian Goodfellow が提唱した機械学習アーキテクチャである．Generator と Discriminator と呼ばれる二つの深層ネットワークを用意し，それらが競い合うように学習を行う．以後，Generator を G, Discriminator を D と表記する．G は画像生成のためのネットワークで，ランダムなノイズ z を受け取り，このノイズから画像 G(z) を生成する．

D は，画像が「本物」がどうかを判断する識別ネットワークである．入力 x は画像，出力 D(x) は x が本物の画像である確率を表す．値が 1 であれば本物と，値が 0 であれば偽物と判断していることを意味する．

学習中，G の目標は，D を騙すためにあたかも本物であるような画像を生成することであり，D の目標は，G によって生成された画像と本物の画像を正確に判別しようとすることである．競合的な学習の結果として望む状態は，D が真偽について相当な能力を擁しながらも，G が D(G(z))=1 の意味で D を騙し，G が本物であるような画像を生成できるようになることである．

3　改良方法について

今日の GAN では，様々なネットワーク構造が提案されている．本研究では，そのようなネットワーク構造として，Conditional GAN[2] で提案されたものと，StyleGAN2[5] で提案されたものを利用する．前者のネットワーク構造を用いた生成法を生成法Ａと呼び，後者のネットワーク構造を用いて生成法を生成法 B と呼ぶ．この節では，生成法Ａと生成法Ｂについてそれぞれ説明する．

3.1　生成法 A

Conditional GAN のネットワーク構造を用いた生成法は大きく分けて 3 つの段階に分けられる．それらを Phase 1, 2, 3 と呼ぶことにする．

Phase 1 では，一つの特徴をもつ顔画像を生成できるような Generator G_1 を作成することが目的となる．特徴が一つであれば，十分な学習データもあるので，通常の GAN に従った競合的な学習を行うことで，G_1 を作成する．学習の過程で G_1 が作成した低解像度やノイズ画像と言えるような不明瞭な画像は，Phase 2 において，特徴を備えない画像として利用する．また，学習結果として得られた G_1 のパラメータの一部は，Phase 3 において，学習済みパラメータとして利用される．これにより，効率的な学習が行えると考える．

Phase 2 では，一つの特徴について，入力された画像が，それを備えるか否かを識別することができる識別器 D_2 を作成することを目的とする．Phase 1 の途中で得られた画像も利用していることにより，D_2 は不明瞭な画像は特徴を備えない画像として識別できることが期待できる．D_2 は，Phase 3 において，一つの特徴について，それを備える画像を安定的に生成するように仕向けるのに利用される．

Phase 3 では，Phase 1 で用いた G_1 に層を追加して得られる Generator G_3 について，Discriminator D_3 との競合関係の中で，二つの特徴を備える画像を生成できるように学習を進める．G_3 では G_1 と共通する部分については，G_1 のパラメータを引き継ぐ．G_3 の損失関数には，D_3 を騙すことを評価する項に加えて，Phase 2 で作成した D_2 に，一つの特徴を備えていることを評価させる項が含まれている．この後者の項の存在により，一つの特徴について生成できることを維持させつつ，もう一つの特徴も備える画像を生成できるようにする．その結果として，二つの特徴を備える顔画像を生成できるようにする．

3.1.1　各 Phase の損失関数

各 Phase で用いる損失関数を中心に説明する．下記では，z_i は乱数から生成されたノイズ画像を表す．

Phase 1 の損失関数　Phase 1 において，x_i は一つ目の特徴を持つ本物画像データを表す．Phase 1 で作成する D_1 の損失関数は，基本的な GAN でも用いられるような

$$\sum_{i=1}^{m}[(D_1(x_i)-1)^2+D_1(G_1(z_i))-0)^2]$$

とする．この損失関数は D_1 が本物データ x_i に対して $D_1(x_i)=1$ として，偽物データ $G_1(z_i)$ に対して $D_1(G_1(z_i))=0$ として，真偽を正しく識別するほど，小さくなる．一方で，G_1 の損失関数も，通常の GAN で用いられる

$$\sum_{i=1}^{m}(D_1(G_1(z_i))-1)^2$$

である．この損失関数は G_1 がノイズから生成する画像 $G_1(z_i)$ に対して，$D_1(G_1(z_i))=1$ として D_1 を騙すほど，小さくなる．

Phase 2 における損失関数　Phase 2 は，画像に指定された特徴があるかないかを判定させる識別器 D_2 を作成することが目的であるので，特徴を持つ画像データ x_i と特徴を持たない画像データ y_i を用意して，D_2 には次の損失関数を小さくするように学習をさせる．

$$\sum_{i=1}^{m}(D_2(x_i)-1)^2+\sum_{i=1}^{m}(D_2(y_i)-0)^2$$

とする．特徴の有無を正しく識別できるほど，この損失関数は小さくなる．特徴がない画像 y_i には，Phase 1 の学習途中で G_1 が作成したほぼノイズのような画像や，低解像度と言えるような画像も含む．

Phase 3 における損失関数　Phase 3 において，x_i は二つ目の特徴を持つ本物画像データを表す．このようなデータを用いることで，通常の GAN と同様に，二つ目の特徴については，学習の結果として，それを備えた画像を G_3 は生成できるようになると考える．したがって，一つ目の特徴を，いかにして G_3 に学習させるかがポイントとなる．Phase 3 における D_3 の損失関数は，

$$\frac{1}{\alpha_1+\alpha_2+\alpha_3}\times\Bigg\{\alpha_1\sum_{i=1}^{m}(D_3(x_i)-1)^2+\alpha_2\sum_{i=1}^{m}(D_3(G_3(z_i))-0)^2+\alpha_3\sum_{i=1}^{m}(D_3(G_3(z_i))-D_2(G_3(z_i)))^2\Bigg\}$$

である．係数 $\alpha_1,\alpha_2,\alpha_3$ については，調整が可能であるが，試行錯誤の結果として現状では，$\alpha_1=1.0,\alpha_2=1.0,\alpha_3=0.1$ としている．Phase 1 と同様に第 1 項と第 2 項より，真偽を正しく識別できるほど，この損失関数は小さくなる．また，第 3 項として識別器 D_2 の識別結果も組み込まれている．D_2 が特徴の有無を識別できることから，この項の影響により，D_3 も指定された特徴が画像に含まれているか否かを識別できるようになることが期待される．一方，G_3 の損失関数は，

$$\frac{1}{\beta_1+\beta_2}\times\Bigg\{\beta_1\sum_{i=1}^{m}(D_3(G_3(z_i))-1)^2+\beta_2\sum_{i=1}^{m}(f(D_2(G_3(z_i)))-1)^2\Bigg\}$$

である．係数 β_1,β_2 についても調整が可能であるが，試行錯誤の結果として現状では，$\beta_1=2.0,\beta_2=30.0$ としている．第 1 項が Phase 1 とも共通するように，この損失関数は D_3 を騙すことが 1 つの目的である．第 2 項に現れる関数 f については，定義が煩雑なので詳細は割愛するが，第 2 項を導入した目的は基本的に $D_2(G_3(z_i))=1$ となること，すなわち，偽物画像 $G_3(z_i)$ に対して，D_2 が一つ目の特徴を含むと識別することである．これにより，G_3 には，一つ目の特徴も備えた顔画像を生成するようにさせる．

3.1.2　AdaIN の追加

Phase 3 において，StyleGAN で使用される AdaIN 操作を組込む．AdaIN[8] とは，Adaptive Instance Normalization の略であり，コンテンツ画像 (入力画像) の特徴量の平均と分散をスタイルの特徴量の平均と分散に合わせるというものである．これを次のように用いる．Phase 3 において，epoch に応じて異なるデータセットを用いる．epoch とは学習の繰り返し回数を表すものであるが，epoch が 500 以下のときは，ひげのデータセットを利用し，それ以降はスキンヘッドのデータセットを使用する．そして，epoch が 500 のときの G_3 を Generator H として保存し，epoch が 500 を超えてからは，G_3 が生成した顔画像と H が生成した顔画像を AdaIN により，合成する．これにより，それぞれの特徴を持ち合わせた画像が生成できるのではないかと考える．

3.2　生成法 B

StyleGAN[6] のネットワーク構造を用いるが，本研究課題に StyleGAN2 を用いる初めてなので，この要旨では StyleGAN2 の簡単な説明と，なぜ StyleGAN2 を用いることを考えたかを説明する．

3.2.1　StyleGAN と StyleGAN2

2018 年に NVIDIA が ProGAN を提案するまで，GAN を用いても大きなサイズの画像を生成することは困難であった．そのような状況の中で，ProGAN が提案した解決策は，4 × 4 という非常に小さなサイズの画像を用いて Generator と Discriminator を学習させることからはじめて，画像サイズとネットワークを順次拡大していくことを繰り返すことで，最終的に高解像で大きなサイズの画像を生成するというものであった．このような手法は，StyleGAN へと引き継がれている．StyleGAN では，

ProGAN の手法がさらに発展している．図 1 に示すように，StyleGAN では，Generator は Synthesis Network と Mapping Network と呼ばれる 2 つのネットワークに分かれる [6]. このうち Mapping Network は 8 つの全結合層からなるネットワークである．この Mapping Network の出力をアフィン変換した後にスタイルとして，Synthesis Network の中で画像が拡大していく途中に複数回にわたって入力する．ここで，Mapping Network の出力をスタイルと呼ぶのは，顔であれば，それが髪型などの指定に関わる変数となるからである．スタイルを Synthesis Network に入力する際には，AdaIN 操作が利用される．StyleGAN の中では，その二つの画像の一方が Mapping Network が出力するスタイルに対応でなり，スタイルに則した画像を Synthesis Network が生成することになる．StyleGAN では，AdaIN 操作を用いていることが一つの特徴ではあったが，AdaIN 操作のために生じていると考えられる問題もあった．それは，「滴」と呼ばれる画像の一部に不鮮明な領域が生じる問題であり，その原因は，AdaIN 操作の中で行われる一種の正規化のためだと考えられた．そこで，AdaIN 操作を代替する方法が考案され，それを組み込んだ画像生成法として，StyleGAN2 が提案されている．

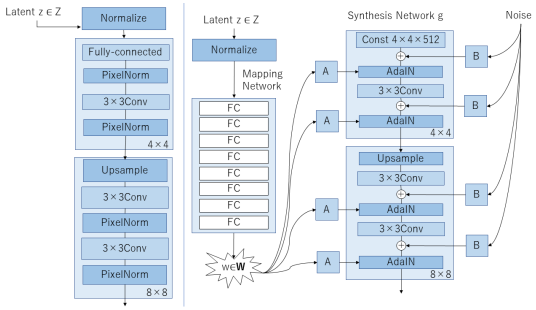

図 1　StyleGAN の構造

3.2.2　StyleGAN2 を用いる理由

StyleGAN2 を利用しようと考える理由の一つは，それらが Style Mixing と呼ばれる機能を持つからである [6]．前述のように，StyleGAN および StyleGAN2 では，Systhesis Network による画像の生成中に，Synthesis に対して，スタイルと呼ばれる変数の入力が複数回あり，それが Synthesis の生成する画像の大まかな特徴を指定する．さらに，二つのスタイルがあるとき，それらのスタイルを複数回入力する中で，適宜切り替えたり，スタイルそのものを合成して入力することで，生成する画像に二つのスタイルを反映させることも可能であるとされている．この機能を利用して，複数の特徴を持つ顔画像の生成を試みる．ただし，そのためには，適切なスタイルを作成できなければならない．スタイルは，それ自体，Mapping Network の出力であるため，スタイルの作成には工夫が必要である．

4　実行結果

4.1　生成法 A で生成した画像

図 2 は Phase 1 において，一つ目の特徴を持つ画像データとしてスキンヘッドのみのデータセットを用いて学習を行った結果，Generator G_1 が生成した顔画像である．Phase 1 そのものは通常の GAN で用いられるものでもある．図 2 からも，GAN を用いれば，本物データとしたデータ，すなわち，一つ目の特徴を備えたデータに十分に近いデータを生成することができることが確認できる．

図 2　Phase 1 で生成された 1 つの特徴を持つ顔画像

図 3 は，Phase 3 の学習の結果，Generator G_3 が生成した画像である．

Phase 3 では，本物データは，二つ目の特徴として，ひげのみのデータセットを用いている．本物データは，一つ目の特徴，すなわち，スキンヘッドであるということを必ずしも備えているとは限らない．したがって，通常の GAN では，ひげという特徴は備えたとしても，スキンヘッドでないような画像が多くなることは十分に考えられる．提案した手法では，G_3 の損失関数に Phase 2 で作成した D_2 による識別結果を組み込むことで，G_3 は一つ目の特徴を備えている画像を生成するように仕向けられているので，比較的二つの特徴を持つ顔画像が多く生成できているように思える．加えて，図 3 は試行回数が 910 回目であり，AdaIN の追加によって，学習の繰り返し回数が多いと，1 つ目の特徴が消滅するのを防ぐことに成功しているように思える．

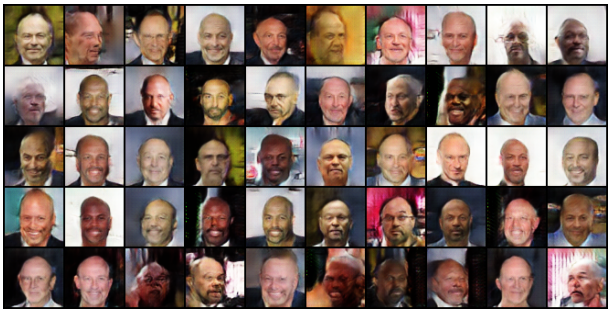

図 3　Phase 3 で生成された複数の特徴を持つ顔画像

また，生成法 A だけでは顔画像の鮮明度を上げるのは，困難だと感じ，画像の補完をしてくれる GPEN[3] という GAN を用いることにした．GPEN を利用することで高画質な顔画像の生成にも成功できた．図 4 がその画像である．すべての画像を掲載するのは要旨集の関係上難しいため，一人の顔画像を掲載する．

図 4　図 3 の画像の中でも複数の特徴がよく現れた顔画像

4.2　生成法 B で生成した画像

生成法 B については，StyleGAN2-ada-pytorch[4] を利用することができることをはじめに確認し，さらに簡単な実験を行った．図 5 はスキンヘッド顔画像のデータとひげ顔画像のデータを，ターゲット画像としてスタイルを作成したのちに，そのスタイルを元に Synthesis Network で画像を作成した結果である．StyleGAN2 の特徴通り，どちらもかなり高精細な画像を生成できており，さらに，ターゲットとしたデータの特徴がはっきりと現れていることを確認した．

図 5　Synthesis Network が生成したスキンヘッドの顔画像 (左) とひげの顔画像 (右)

一方，図 6 は図 5 を得るのに作成したスタイルを Style Mixing した際に Synthesis Network によって，生成された画像がである．スキンヘッドとひげを特徴とするスタイルを用い Style Mixing を行うことで，スキンヘッドでありながらひげの特徴を加えた，かなり高精細な画像を生成できた．

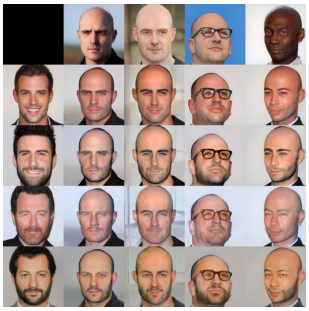

図 6　Style Mixing 機能を使った結果

5　おわりに

複数の特徴を持つ顔画像の生成に対して，生成法 A と生成法 B を用いて研究を進めてきた．研究結果として相性の良い複数の特徴を持つ顔画像の生成は成功できたのではないかと考える．まず，前者の GAN は，AdaIN の導入により一つ目の特徴を維持できたことで複数の特徴を持つ顔画像が生成できた．後者の GAN については，StyleGAN2-ADA アルゴリズムを利用したこで，複数の特徴を持つ高精細な顔画像を生成することができた．

参考文献

[1] 宮本圭一郎, 大川洋平, 毛利拓也：『PyTorch ニューラルネットワーク 実装ハンドブック』秀和システム,2018.

[2] 杜世橋：『現場で使える！ PyTorch 開発入門 深層学習モデルの作成とアプリケーションへの実装』翔泳社,2018.

[3] Tao Yang, Peiran Ren, Xuansong Xie, Lei Zhang: GAN Prior Embedded Network for Blind Face Restoration in the Wild, arXiv:2105.06070 [cs.CV], 2021.

[4] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, Timo Aila: Training Generative Adversarial Networks with Limited Data arXiv:2006.06676 [cs.CV], 2020.

[5] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila: Analyzing and Improving the Image Quality of StyleGAN, arXiv:1912.04958 [cs.CV], 2019.

[6] Tero Karras, Samuli Laine, Timo Aila: A Style-Based Generator Architecture for Generative Adversarial Networks arXiv:1812.04948 [cs.NE], 2019.

[7] 内田翔太朗：『特徴識別器を組み込んだ GAN による複数の特徴を持つ顔画像の生成』．南山大学 理工学部 2021 年度　卒業論文, 2022.

[8] Xun Huang, Serge Belongie: Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization arXiv:1703.06868 [cs.CV], 2017.