修士論文要旨（2021 年度）
顔特徴に基づく GAN を用いたアニメ風顔画像の生成
Anime-Style Face Rendering Using GAN Based on Face Features

電気電子情報通信工学　劉　挙豪

1. まえがき

GAN[4]は、2 つのネットワークを競わせながら学習させるアーキテクチャとなっている。GAN は生成モデルの一種であり、データから特徴を学習することで、実在しないデータを生成したり、存在するデータの特徴に沿って変換できる。さらに 2015 年に、畳み込みニューラルネットワーク（CNN）で見られるような畳み込み層をネットワークに適用した DCGAN(Deep Convolutional GAN)[4]は、鮮明な画像生成を行うとともに、学習を安定化させる手法として提案された。

DCGAN に基づき、2017 年に、Zhu らはアンペアなデータセットを用いて学習し、画像のスタイルを相互変換することが可能な「CycleGAN」[5]を提案した。「CycleGAN」により、アンペアな画像で学習しているにも関わらず、元の画像の形状をほとんど保っている上、画像のスタイルを別のスタイルへ変換することができる。2018 年に、Chen らが提案した手法「CartoonGAN」[2]は「CycleGAN」のような循環しているアーキテクチャを使用しなく、一方向の画像生成ネットワークを構成し、「Content Loss」により「CycleGAN」のようにアンペアな画像で学習して元の画像内容を保持する上、画像をアニメ風に変換することができる。また、Chen らは 2019 年に、「CartoonGAN」の課題に対して改善して「AnimeGAN」[1]を提案した。

「AnimeGAN」は、「Grayscale style loss」と「Color reconstruction loss」を用いて実写の風景画像をアニメ風の風景画像へ変換することが得意だとなっている。しかしながら、「AnimeGAN」を用いて実写の顔画像をアニメ風顔画像へ変換し、生成されたアニメ風画像が期待しているような画像になっていない。生成されたアニメ風顔画像の中、鮮明な肌色と五官がうまく生成されなかった。また、実際の写真の中、例えば全身像の中、人間の顔が小さくなった場合に、その生成されたアニメ風の顔が崩れていた問題がある。

本研究では、「AnimeGAN」により肌色と顔特徴を鮮明に生成するための損失関数を提案し、実写の顔画像を用いて生成したアニメ風顔画像の品質を向上させることを目的とする。また、「GAN」モデルの高性能を保持する上、モデルを軽量化、学習時間を短縮する方法を検討する。

2. 提案手法

提案手法では、高品質なアニメ風顔画像を生成するために、損失関数「Skin Color Loss」と「Face Feature Loss」を提案した。また、「Huber 損失」を使用して従来の損失関数を改めて設計した。さらに、モデルを軽量化、学習時間を短縮するために、「MobileNet」[3]を用いて画像特徴を抽出する手法を提案した。

2.1 「Skin Color Loss」損失関数

本研究では、生成されたアニメ風画像中、人物の肌色を保持するために、「Skin Color Loss」という損失関数を提案する。以下の手順を沿い、「Skin Color Loss」を計算する。

① 実写画像と生成された偽画像の中、顔の領域を確定する。

② 実写画像と生成された偽画像の中、顔領域の肌色を、RGB 色空間モデルから YUV 色空間へ変換する。

③ 「Skin Color Loss」の数式は式 1 のように定義する。

$$L_{skin}(G,D) = E_{p_i \sim S_{data}(p)}[||Y(G(p_i)) - Y(p_i)||_H + ||U(G(p_i)) - U(p_i)||_H + ||V(G(p_i)) - V(p_i)||_H]$$

式 1

2.2 「Face Feature Loss」損失関数

本研究では、生成されたアニメ風画像中、人物の鮮明な五官を保持するために、「Face Feature Loss」損失関数という損失関数を提案する。以下の手順を沿い、「Skin Color Loss」を計算する。

① 実写画像と生成された偽画像の中、人物の顔領域を確定する。

② 実写画像と生成された偽画像の中、「ResNet101」を用いて顔特徴を抽出する。

③ 「Huber 損失」を用いて「本物の顔特徴」と

「偽顔特徴」の類似性を計算する。

④ 「Face Feature Loss」は式2のように定義している。

$$L_{face}(G,D) = E_{p_i \sim S_{data}(p) \sim q_i}[||RN(q_i) - RN(G(p_i))||_H]$$

式 2

2.3 損失計算方法

式3のように「Huber 損失」(Huber Loss：フーバー損失）とは、調整可能なパラメーターδ（デルタ）を例えば1.0とした場合、各データに対して「予測値と正解値の差（＝誤差、残差)」が0.0～1.0（δ）の範囲では「誤差の二乗値に0.5（＝1/2）を掛けた値」を計算し、1.0（δ）より大きい（外れ値になる可能性が高い）範囲では「誤差の絶対値から0.5を引いた値」を計算する関数のこと、もしくはその計算結果の総和をデータ数で割った値（＝平均値）を出力する関数を指す。

$$\text{HuberLoss}(a) = \begin{cases} \frac{1}{2}a^2, & |a| \le \delta \\ \delta(|a| - \frac{1}{2}\delta), & |a| > \delta \end{cases}$$

式 3

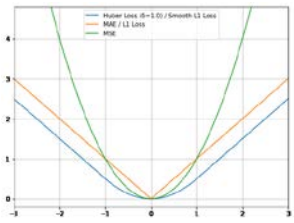

図 1 Huber 損失

図1のように、「Huber 損失」は、「L1 損失＝0」地点で「微分不可能」で、0 に近い場所でも勾配が大きいという弱点と、「L2 損失＝外れ値」に敏感という弱点を克服した。

「Huber 損失」を用いて「AnimeGAN」に使用されている損失関数[1]を式4のように変更させる。

$$L_{con}(G,D) = E_{p_i \sim S_{data}(p) \sim q_i}[||VGG_l(q_i) - VGG_l(G(p_i))||_H]$$

$$L_{gra}(G,D) = E_{p_i \sim S_{data}(p) \sim q_i, E_{x_i \sim S_{data}(x) \sim q_i}}[||Gram(VGG_l(G(p_i))) - Gram(VGG_l(x_i))||_H]$$

$$L_{col}(G,D) = E_{p_i \sim S_{data}(p) \sim q_i}[||Y(G(p_i)) - Y(p_i)||_H + ||U(G(p_i)) - U(p_i)||_H + ||V(G(p_i)) - V(p_i)||_H]$$

式 4

2.4 MobileNet

従来の「AnimeGAN」において、生成された偽画像の内容と実写画像の内容を一致するための「Content Loss」の中、偽画像内容の特徴マップと実写画像内容の特徴マップを抽出するために、「VGG19」を使用した。本研究では、「Skin Color Loss」と「Face Feature Loss」により品質がより良いアニメ風顔画像を生成する一方、計算コストと計算時間が上がる可能性がある。従って、本研究では、従来の「AnimeGAN」に使用した「VGG19」の他に軽量かつ高性能な「MoblieNet」[3]を使用することを提案する。

「VGG19」はモデルサイズが549MBとなり、正確率が71.3%、推理時間が一回につき4.4msとなっている。提案した「MobileNet」は、正確率が70.4%、「VGG19」モデルのと近似しているが、モデルサイズが14MB、推理時間が一回につき3.8msとなっている。つまり、「MobileNet」モデルは「VGG19」モデルより、モデルサイズが535MB、推理時間が一回につき0.6ms削減した。従って、提案手法に使用されている損失関を式5のように変更させる。

$$L_{con}(G,D) = E_{p_i \sim S_{data}(p) \sim q_i}[||VGG_l(q_i) - VGG_l(G(p_i))||_H]$$

式 5

3. 実験

3.1 データセット

実験には、実写の顔画像5000枚を「Content images」、アニメの顔画像1365枚を「Style images」とする。そして、学習用画像のサイズを全部「256 x 256」に変更する。また、前処理として実写の顔画像の背景を取り除いた。図2は、「Content images」と「Style images」の例である。

図 2 データセットの例

3.2 実験結果

実験において、事前に生成器の「Learning Rates」を0.00008、判別器の「Learning Rates」を0.00016、「学習 Epoch」を100を10と設定している。

3.2.1 Skin Color Loss

本実験において、「Skin Color Loss」を「AnimeGAN」にも使用されている損失関数と合わせ、式6を求めることでモデルの学習を行う。ここでは、ω_{adv}を300、ω_{con}を3、ω_{gra}を3.5、、ω

$_{col}$ を 10、ω_{skin} を 2.8 と設定している。

$$L(G,D) = \omega_{adv}L_{adv}(G,D) + \omega_{con}L_{con} + \omega_{gra}L_{gra}(G,D) + \omega_{col}L_{col}(G,D) + \omega_{skin}L_{skin}(G,D)$$

式 6

図 3 は「Skin Color Loss」損失関数を用いたモデルと従来のモデルで生成したアニメ風顔画像の例を示している。図のように、「Skin Color Loss」を用いたモデルで生成した偽画像の品質は、従来のモデルで生成した偽画像の品質より高くなっている。従来のモデルで生成した偽画像の肌色が真っ白になっており、「Skin Color Loss」を用いたモデルで生成した偽画像の肌色がピンク色が入っており、従来のよりよくなった。従って、損失関数「Skin Color Loss」が高品質なアニメ風顔画像の生成に適当であることが考えられる。

図 3　生成したアニメ風顔画像

3.2.2　Face Feature Loss

本実験において、「Face Feature Loss」を「AnimeGAN」にも使用されている損失関数と合わせ、式 7 を求めることでモデルの学習を行う。ここでは、ω_{adv} を 300、ω_{con} を 3、ω_{gra} を 3.5、、ω_{col} を 10、ω_{skin} を 2.8、ω_{face} を 2 と設定している。

$$L(G,D) = \omega_{adv}L_{adv}(G,D) + \omega_{con}L_{con} + \omega_{gra}L_{gra}(G,D) + \omega_{col}L_{col}(G,D) + \omega_{skin}L_{skin}(G,D) + \omega_{face}L_{face}(G,D)$$

式 7

図 4 のように、「Skin Color Loss」と「Face Feature Loss」両方ともに用いたモデルで生成した偽画像の品質は、「Skin Color Loss」のみを用いたモデルで生成した偽画像の品質より高くなっており、顔特徴が鮮明になっていることが分かった。従って、損失関数「Face Feature Loss」が高品質なアニメ風顔画像の生成に適当であることが考えられる。

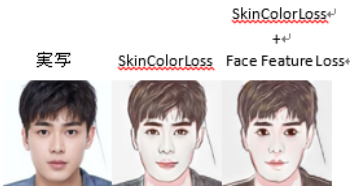

図 5　生成したアニメ風顔画像

3.2.3　損失計算方法を変更させる実験

本実験において、損失関数中の損失計算方法が学習時間と生成する偽顔画像の品質にどう影響するかを検討するために、従来の AnimeGAN において使用し損失計算方法「L１損失」、「L２損失」の利点と欠点を考慮し、「Huber 損失」を損失関数中に使用されたところに入れ替え、実験を行う。表 1 は、各損失計算方法を用いたモデルの学習時間を示している。

損失計算方法	Epoch	学習時間(h)
L１損失 L２損失	100	21.94
Huber 損失	100	23.19

表 1　学習時間

表 1 と図 5 のように、「Huber 損失」を用いたモデルの学習時間は、「L１損失」・「L２損失」を用いたモデルのより長くかかったが、「Huber 損失」を用いたモデルで生成した偽画像の品質が「L１損失」・「L２損失」を用いたモデルのよりかなりよくなっている。従って、学習時間と偽画像の品質を両方ともに考慮し、損失計算方法「Huber 損失」が高品質なアニメ風顔画像の生成に適当であることが考えられる。

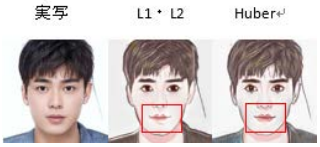

図 4　生成したアニメ風顔画像

3.2.4　CNN モデルを変更させる実験

本実験において、モデルの計算コストを軽減、計算時間を短縮するために、従来の AnimeGAN において使用した画像特徴を抽出する CNN モデル「VGG19」の利点と欠点を考慮し、「MobileNet」という高性能かつ軽量な CNN モデルを画像特徴を抽出する方法に入れ替え、実験を行う。表 2 は、各 CNN モデルを用いたモデルの学習時間を示す。

CNN モデル	Epoch	学習時間(h)
VGG19	100	22.36
MobileNet	100	19.92

表 2　学習時間

図 6 にように、「MobileNet」を用いたモデルで生成した偽画像の品質が「VGG19」を用いたモデルで生成した偽画像の品質に近似しているが、「MobileNet」を用いたモデルの学習時間は、「VGG19」を用いたモデルのより短くなっている。

従って、学習時間と偽画像の品質を両方ともに考慮し、画像特徴を抽出する方法「MobileNet」モデルが高品質なアニメ風顔画像の生成に適当であることが考えられる。

図 6　生成したアニメ風顔画像

3.3　実験評価

3.3.1　定性評価

図 7 のように、提案したモデルで生成したアニメ風顔画像は、肌色がよくなっており、五官が鮮明になっている。一方、従来のモデルで生成したアニメ風顔画像は、肌色が真っ白になっており、五官が鮮明になっておらず、特に目がアニメ風と言えないと考える。従って、定性評価により、提案した手法の有効性を検証した。

図 7　生成したアニメ風顔画像

3.3.2　定量評価

定量評価において、アンケート調査により提案した手法の有効性を評価する。具体的に、各モデルに対する定量評価において、悪い品質から高い品質まで、点数範囲を「0 点～10 点」と設定し、画像だと知らず回答者 10 人が画像の品質に対してスコアをつける。画像枚数合計「100 枚 x10 人 =1000」枚となる。表 3 はスコアの結果を示す。

モデル	平均得点 x100
AnimeGAN	64.67
本研究	82.09

表 3　評価得点

表 3 のように、本研究で生成した偽画像の得点は、AnimeGAN で生成した偽画像の得点より 17.42 点高くなっている。それにより、本研究で提案したモデルは AnimeGAN より高品質なアニメ風顔画像を生成することができる。従って、定量評価により、提案した手法の有効性をを検証した。

4.　おわり

本研究では、実写顔画像を高品質なアニメ風顔画像へ変換するために、AnimeGAN を用いて実験を行った。、本研究では、、提案した損失関数「Skin Color Loss」と「Face Feature Loss」の有効性を検証した。また、損失計算方法「Huber 損失」が偽画像の品質を向上させることができると分かった。さらに、「MobileNet」モデルを用いて偽画像の品質が変わらなかった上、モデルの学習時間が短縮されたことを検証した。

今後の課題について、より大規模のデータセットを用いて実験する必要がある。そのデータセット中、様々な背景がある顔画像と人物全身像を入れる必要がある。また、異なる画風のアニメ画像のデータセットを「Style images」とする必要がある。「Face Feature Loss」損失関数において、「ResNet101」モデルの他に高性能かつ軽量の CNN モデルを用いて実験する必要がある。

参考文献

[1] Chen, Jie, Gang Liu, and Xin Chen. "AnimeGAN: A novel lightweight gan for photo animation." International Symposium on Intelligence Computation and Applications. Springer, Singapore, 2019.

[2] Chen, Yang, Yu-Kun Lai, and Yong-Jin Liu. "Cartoongan: Generative adversarial networks for photo cartoonization." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[3] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).

[4] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

[5] Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." Proceedings of the IEEE international conference on computer vision. 2017.