修 士 論 文

機械学習を用いた人物画像の表情変換

広島大学大学院先進理工系科学研究科

倪 志豪

令和５年９月

修 士 論 文

機械学習を用いた人物画像の表情変換

指導教官　中野浩嗣 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

M211488 倪 志豪

提出年月: 令和 5 年 8 月

概 要

近年，機械学習の発展に伴い，様々な場所で活用されるようになっている．特に，人物画像の処理研究は盛んで，その応用は多くのアプリケーションに搭載されている．

本研究では，敵対的生成ネットワーク (GAN) を用いて，入力されたポートレート画像の顔の表情を変換するためのモデルが提案されている．このモデルでは，一般的な表情変換アプリとは異なり，GAN と顔面動作符号化システム (FACS) を組み合わせ，喜怒哀楽などの標準的な表情に直接変換するだけでなく，顔のアクションユニット (AU) を個別に制御して特定のパーツの表情を微調整することも可能である．なお，既存のデータセットを処理し，ConvNeXt ネットワークを用いてオリジナルのモデルを改造する．それで，モデルから生成される画像の品質が向上し，モーフィングのような変換効果も実現された．

目 次

図 目 次

第1章　はじめに

1.1　研究の背景

近年，TikTok や Instagram などのソーシャルメディアの発達により，人々はこれらのソーシャルメディアに写真を投稿し，生活を共有することを楽しみ始めている．写真を編集するとき，アプリケーションはいくつかの追加機能を提供することが多い．例えば、明るさと色を変更できるフィルター，位置情報を表示するステッカーなどだ．これらのフィルターを使って，写真をクリエイティブに加工することに興味を持つ人も増えている．

また，近年の機械学習の発達により，画像処理の技術はますます発展している．その中でも特に注目されているのが，機械学習による表情変換である．市販の自撮り写真加工アプリは，ユーザーが入力した自撮り写真を，表情変換機能を使って他の表情に変換することができる (faceapp[2] など)．

1.2　研究の目的

既存の自撮り写真加工アプリでは，図 1.1 のように表情のスムーズに変換するケースが多く，良い効果が出ているが，またまた問題点がある．

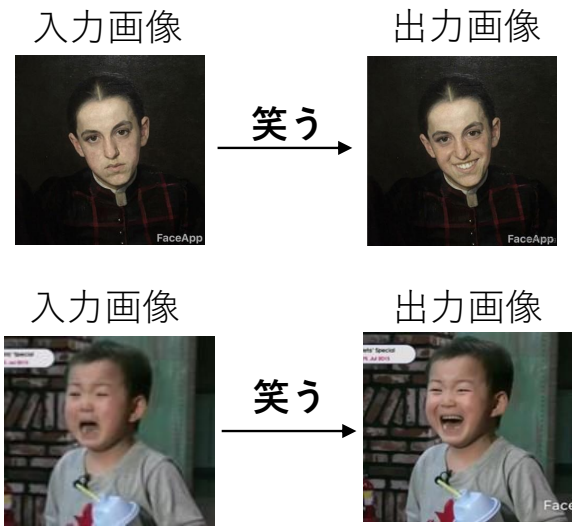

図 1.1: faceapp の表情変換機能

問題の一つ目は，これらの表情変換機能は一般的にスマイル表情にしか変えなくて，スマイルの大きさも調整できない．

問題の二つ目は，スマイル表情に変換するとき，その表情は顔全体の変化であるべきなのに，口元だけが個別に変換している．或いは，笑っている他人の写真の口元部分を複写し，元の画像に貼り付けることである．その結果，変換された画像には違和感が多く，目鼻立ちが本人に似ていないことである．

これらの問題を解決するために，本研究の目的は 3 つある．

一つ目は，全体的な表情の変化だけでなく，図 1.2 のように顔のどの部分の表情の大きさも個別に調整できるようにすることである．

図 1.2: 顔の単一部分の微調整

二つ目は，最終的な生成結果を直接示すのではなく，図 1.3 のように段階的な中間過程もモーフィングという形で示す．

図 1.3: モーフィングのような変換効果

三つ目は，より高品質な画像を生成することができる．

1.3　本論文の構成

本論文は 7 章で構成されており，第 2 章では本研究で導入した顔面動作符号化システムについて紹介し，第 3 章では本研究に関連する敵対的生成ネットワークについて説明する．第 4 章では本研究で実装したネットワークモデルと自分で設計した訓練方法について説明を行う．第 5 章では本研究で使用したデータセットと，自分で行った前処理方法について説明する．第 6 章では実験結果を示す．最後に，第 7 章では本論文のまとめを行う．

第2章　顔面動作符号化システム

表情をより正確に表現すると，個々の部分の微細な調整を実現するために，本研究では革新的な顔面動作符号化システムを導入した．このシステムは，顔の様々な筋肉部位を異なる区域に分割し，それらを符号化し，具体的な数値を対応する区域の筋肉の動きの度合いを制御する．このようにして，機械には理解できない表情が，理解できる符号化された形に変換される．この顔面動作符号化システムについては，本章で詳しく説明する．

2.1　FACS と AU

FACS (Facial Action Coding System) は顔面動作符号化システムである．人間の顔に現れる表情を，AU (Action Unit) と呼ばれる様々な顔面筋の動作の有無の組み合わせで機械判別可能な形に符号化するための仕組みである．

AU の定義については合計で 46 種類あるが，本研究では，表 2.1 に示すように 17 個の主要な AU 属性のみを選定した．

2.2　AU 属性の制御能力の調査

このセクションでは，AU 属性の一部がどのように画像の表情を制御しているか例を挙げて説明する．

2.2.1　単一の AU 属性の調査

前述した通り，顔面動作符号化システムは，顔を異なる部分に分割し，Action Unit で制御する．そして，AU 属性値の大きさは，その部分の顔面筋の動作状況を直接制御することができる．一般的に，AU 属性値は 0 から始まり，0 はその部分の筋肉の動作の度合いが 0 であることを示す．AU 属性値を増加させる過程で，対応する部分の筋肉の動作の度合いも徐々に強まっている．AU 値が大きすぎると生成された画像が崩れてしまうため，一般的に最大値は 2.0 に設定される．

表 2.1: AU 属性

AU 番号	AU の部位・動作
1	眉の内側を上げる
2	眉の外側を上げる
4	眉を下げる
5	上瞼を上げる
6	頬を持ち上げる
7	瞼を緊張させる
9	鼻に皺を寄せる
10	上唇を上げる
12	唇両端を引き上げる
14	えくぼを作る
15	唇両端を下げる
17	頤を上げる
20	唇両端を横に引く
23	唇を固く閉じる
25	顎を下げずに唇を開く
26	顎を下げて唇を開く
45	まばたく

例えば，AU4 は眉の上がり度合いを表し，図 2.1 に示すように，AU4 の値が 0 から 2.0 になるにつれて，画像の顔は眉が徐々に上がっている．

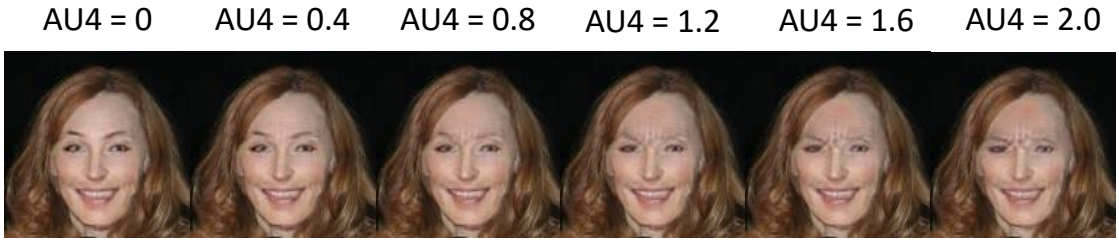

図 2.1: 単一の AU 属性の制御能力 (AU4 の例)

2.2.2 複数の AU 属性の共同効果

複数の AU 属性の値が同時に調整される場合，共同効果が発生する．

共同効果は，相乗効果と競合効果に分かれる．

相乗効果とは，複数の AU 属性が同じ表情と正の相関性を持つことである．これらの AU 属性の値を同時に増減させることで，表情をよりスムーズに変化させることができる．

例えば，図 2.2 に示すように，AU6，AU12 と AU25 はどちらも笑顔を制御する AU 属性であり，同時に増加させると表情はどんどん笑顔に近づいていく．これは相乗効果である．

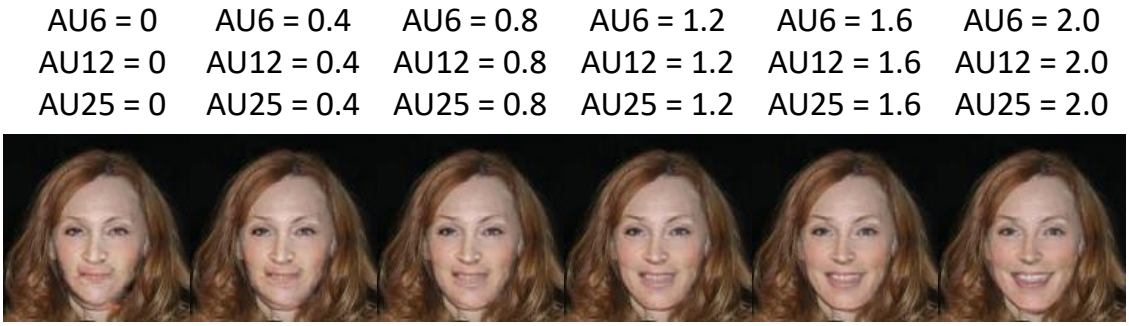

図 2.2: 複数の AU 属性の相乗効果

一方，競合効果とは，複数の AU 属性が相反する関係にあり，同時に増加すると，生成された画像の崩れが発生しやすいことである．

例えば，図 2.3 に示すように，AU23 は唇を固く閉じる，AU25 は唇を開くことであり，この二つ AU 属性の値を同時に増加させると，競合効果により，生成された画像の口の部分が崩れた．したがって，複数の AU 属性の値を変更する際には，競合効果が生じないように注意する必要がある．

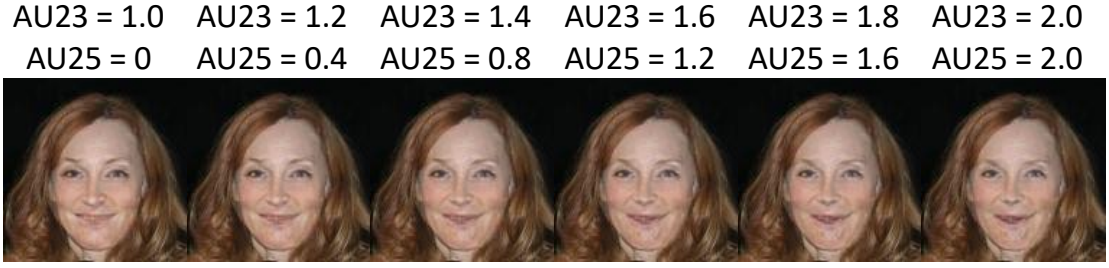

図 2.3: 複数の AU 属性の競合効果

2.3 異なる表情の AU 属性の合成式

異なる表情は，異なる顔面筋と関連している．例えば，笑顔は目と口元に関係し，悲しみは眉と鼻に関係する．そこで，異なる表情を異なる AU 属性の組み合わせとして理解することで，表情を数値で表すことが可能になる．表 2.2 に，8 つの標準的な表情の AU 属性の合成式を示す．

これら 8 つの標準的な表情に加え，異なる AU 属性を組み合わせることで，理論的には無数の表情を作ることが可能である．

表 2.2: 8 つの標準表情の AU 計算式

表情	AU 計算式
Natural	-
Happiness	AU6 + AU12
Sadness	AU1 + AU4 + AU15
Surprise	AU1 + AU2 + AU5 + AU26
Fear	AU1 + AU2 + AU4 + AU5 + AU7 + AU20 + AU26
Anger	AU4 + AU5 + AU7 + AU23
Disgust	AU9 + AU15 + AU16
Contempt	AU12 + AU14

第3章　敵対的生成ネットワーク

3.1　GAN

GAN[3] とは，Generative Adversarial Network（敵対的生成ネットワーク）と呼ばれる 2014 年に提案されたネットワークである．近年は画像生成の分野で傑出した貢献をしている．その特徴は，一つの生成器と一つの識別器からなる「敵対的」な生成ネットワークである．GAN の全体の構造は図 3.1 のように示す．

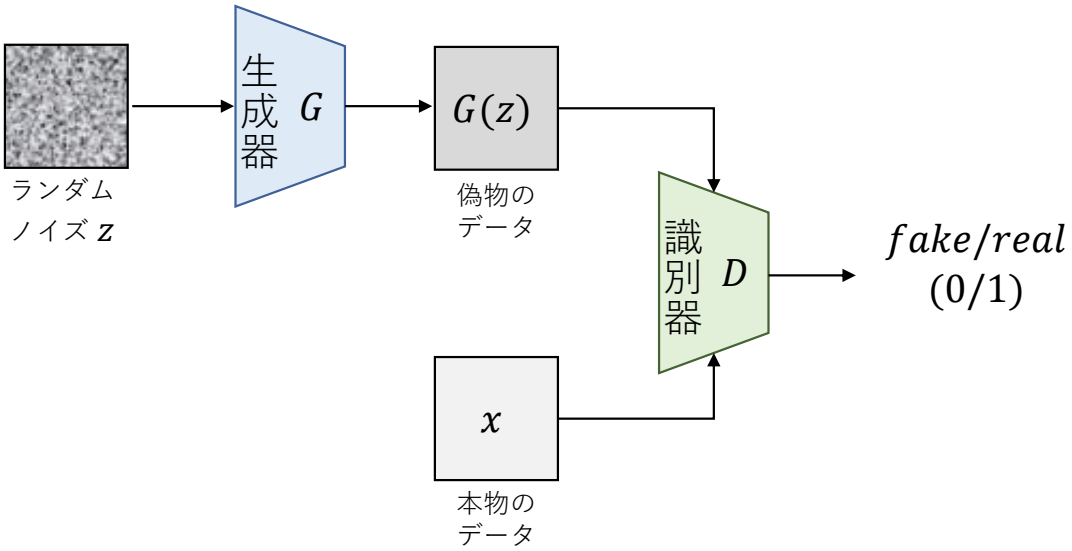

図 3.1: GAN の構造

具体的に言うと，生成器は入力されたランダムノイズ z を受け取り，偽の画像 $G(z)$ を生成する．そして，偽画像 $G(z)$ と真画像 x が識別器 D に入力され，識別器 D が真偽を判定する．

GAN の損失関数は式 3.1 のように示す．

$$V(D,G) = \min_G \max_D \left\{ \mathbb{E}_{x \sim P_{\text{data}}(x)} \left[\log D(x) \right] + \mathbb{E}_{z \sim P_z(z)} \left[\log(1 - D(G(z))) \right] \right\} \tag{3.1}$$

この過程では，生成器のタスクは，識別器を欺くことができる偽画像を生成し，識別器が偽画像にも高得点を与えるようにすることである．識別器のタスクは，真の画像と偽の画像を正確に区別し，真の画像に数値 1 傾向のスコアを与え，偽の画像に数値 0 傾向のス

コアを与えることである．このように，生成器と識別器を敵対的学習することで，生成器がより真の画像に近い画像が生成できるようになる．

3.2 CycleGAN

二つの画像間のスタイル変換のタスクを実行する場合，GAN は単方向生成であるため，性能はあまり良くない．

ランダムなノイズを入力として画像を生成する GAN に比べ，CycleGAN[4] は画像を入力として，別のスタイルの画像を生成する．なお，CycleGAN は 2 組の生成器と識別器を使用する．一組はドメイン X からドメイン Y への変換を実現し，もう一組はドメイン Y からドメイン X へのスタイル変換を実現する．この双方向変換の学習によって，2 つの異なるスタイルドメイン間の変換を可能し，生成される画像はより良い品質になる．

CycleGAN の全体の構造は図 3.2 のように示す．

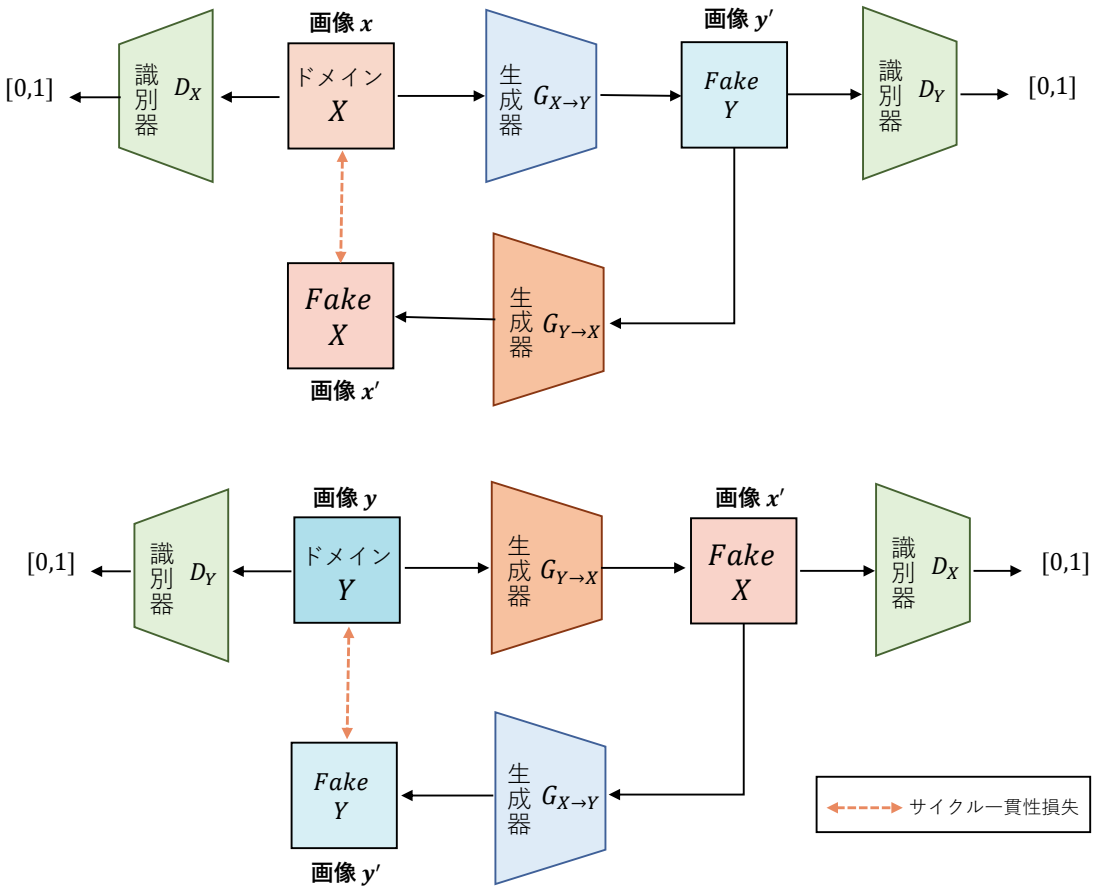

図 3.2: CycleGAN の構造

具体的には，まずドメイン X の画像 x を入力画像として，生成器 $G_{X\to Y}$ によってドメイン Y の偽の画像 $y^{'}$ を生成して，識別器 D_Y が画像 $y^{'}$ の真偽を識別する．真であればあるほど 1 に近くなり，偽であればあるほど 0 に近くなる．そして画像 $y^{'}$ は生成器 $G_{Y\to X}$ によってドメイン X の偽の画像 $x^{'}$ に還元される．画像 $x^{'}$ と入力画像 x は可能な限り近づけるべきである．

次に，ドメイン Y の画像 y を入力画像として，生成器 $G_{Y\to X}$ によってドメイン X の偽の画像 $x^{'}$ を生成して，識別器 D_X が画像 $x^{'}$ の真偽を識別する．真であればあるほど 1 に近くなり，偽であればあるほど 0 に近くなる．そして画像 $x^{'}$ は生成器 $G_{X\to Y}$ によってドメイン Y の偽の画像 $y^{'}$ に還元される．画像 $y^{'}$ と入力画像 y は可能な限り近づけるべきである．

CycleGAN の損失関数は，敵対性損失とサイクル一貫性損失の 2 つに分けられる．

敵対性損失とは，前節で説明した GAN の損失と同じで，学習過程で生成器と識別器が互いに敵対することによって生じる損失である．この過程において，生成器はターゲットドメインにできるだけ近い画像を生成して，識別器は各画像がどのドメインに属するかを正確に識別する．

CycleGAN は生成器と識別器のペアを 2 つ持つので，敵対性損失も 2 つある．これらはそれぞれ，式 3.2 に示す X ドメインから Y ドメインへの敵対性損失と，式 3.3 に示す Y ドメインから X ドメインへの敵対性損失である．

$$\mathcal{L}_{GAN}(G_{X\to Y}, D_Y, X, Y) = \mathbb{E}[\log(D_Y(y))] + \mathbb{E}[\log(1 - D_Y(G_{X\to Y}(x)))] \tag{3.2}$$

$$\mathcal{L}_{GAN}(G_{Y\to X}, D_X, Y, X) = \mathbb{E}[\log(D_X(x))] + \mathbb{E}[\log(1 - D_X(G_{Y\to X}(y)))] \tag{3.3}$$

CycleGAN の学習には敵対性損失だけでは不十分である．ドメイン X の画像 x を入力して，生成器 $G_{X\to Y}$ と生成器 $G_{Y\to X}$ による 2 回の変換後に得られる還元画像 x' は，x にできるだけ似ていることも望ましい．これはサイクル一貫性損失である．具体的な計算式を式 3.4 に示す．

$$\mathcal{L}_{Cycle} = \mathbb{E}|G_{Y\to X}(G_{X\to Y}(x)) - x| + \mathbb{E}|G_{X\to Y}(G_{Y\to X}(y)) - y| \tag{3.4}$$

したがって，CycleGAN の全損失関数 L_{total} は，式 3.5 に示すように，敵対性損失とサイクル一貫性損失を組み合わせたものになる．

$$\mathcal{L}_{total} = \mathcal{L}_{GAN}(G_{X\to Y}, D_Y, X, Y) + \mathcal{L}_{GAN}(G_{Y\to X}, D_X, Y, X) + \lambda\mathcal{L}_{Cycle} \tag{3.5}$$

3.3 StarGAN

CycleGAN は 2 つのドメイン間で画像変換する手法だが，複数のドメイン間でスタイルを変換するタスクを実行する場合，複数ドメインの任意 2 つのドメイン間で生成器と識別

器が必要である，その状況は複雑になる．図 3.3 に示すように，5 つのドメイン間でスタイル変換を行う場合，合計 20 組の生成器と識別器を用意する必要がある，パラメーターの数は非常に大きくなる．

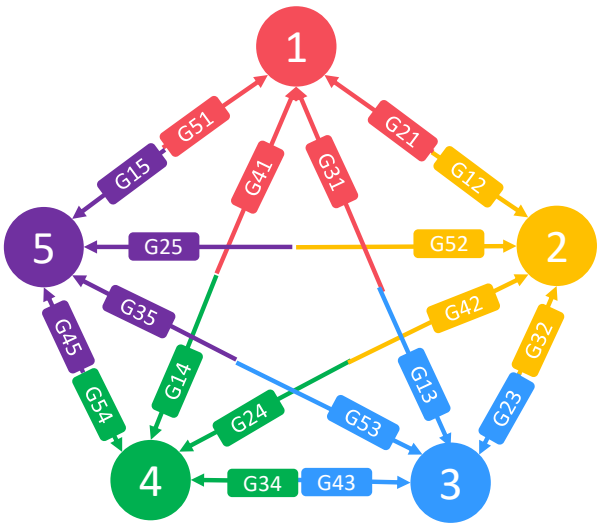

図 3.3: CycleGAN を用いた複数ドメインの変換

この問題を解決するために，StarGAN[1] が提案された．StarGAN は生成器の入力にターゲットドメインを表すラベルを付加し，識別器はソースドメインを分類することで単一の生成器で複数ドメインの画像変換が可能である．StarGAN の基本構造を図 3.4 に示す．中央の単一の生成器は，画像スタイルを周囲のさまざまなドメインに変換できる．

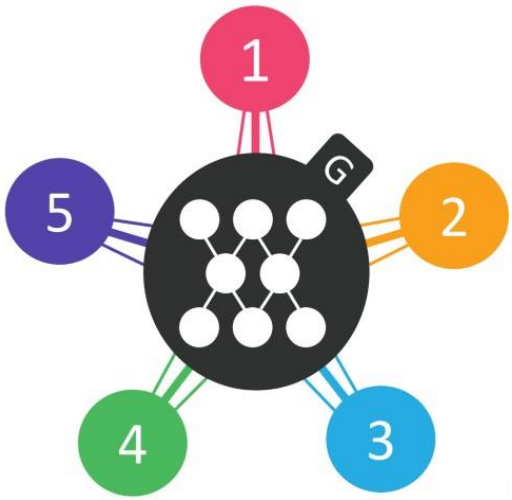

図 3.4: StarGAN の基本構造 [1]

これにより，複数のドメイン間で画像スタイルを変換するタスクを，一つの生成器と一

つの識別器だけで実現できるようになり，学習時間と GPU 資源の使用量が大幅に削減される．

例えば，異なる表情を変換する場合，各ドメインは 1 つの表情に対応し，値 0 対応するドメインの表情が表示されない，値 1 は対応するドメインの表情が表示されることを意味する．

図 3.5 に示すように,「無表情」「怒り」「笑」「悲しみ」「驚き」の 5 つの表情の相互変換を行う場合，5 つの表情に対応するドメイン数は 5 つになるはずである．

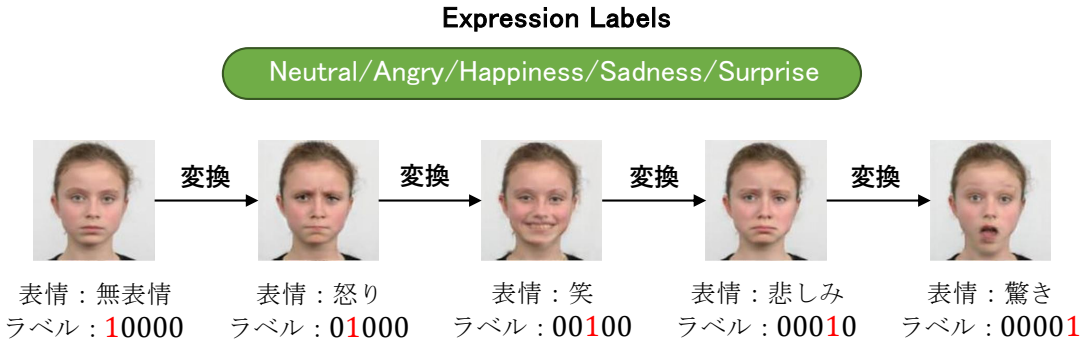

図 3.5: StarGAN を用いた表情変換

「無表情」という表情が示された場合，対応するラベルの値は 10000 である．最初の 1 は「無表情」が有効であることを示し，次の 4 つの 0 は他の 4 つの表情が無効であることを示す．

「怒り」の表情に変換すると,「無表情」の対応するドメインの値は 0,「怒り」の対応するドメインの値は 1 となり，ラベルの値は 01000 となる．

この過程で，StarGAN では 3 種類の損失を組み合わせた損失関数を用いる．敵対性損失 (Adversarial Loss)，ドメイン分類損失 (Domain Classification Loss) と再構成損失 (Reconstruction Loss) ．

敵対性損失とは，前節の説明と同じ，学習過程で生成器と識別器が互いに敵対することによって生じる損失である．StarGAN の敵対性損失は式 3.6 のようになる．

$$\mathcal{L}_{adv} = \mathbb{E}_x[\log D_{src}(x)] + \mathbb{E}_{x,c}[\log(1 - D_{src}(G(x,c)))] \tag{3.6}$$

画像ドメインの正確な変換を実現するために，StarGAN はドメイン分類損失を追加する．

式 3.7 は，生成器のドメイン分類損失である．生成器の学習目標は．生成された画像が目標ドメインとして分類されるように，生成された画像のドメイン分類損失を最小化することである．

$$\mathcal{L}_{cls}^{f} = \mathbb{E}_{x,c}[-\log D_{cls}(c|G(x,c))] \tag{3.7}$$

式 3.8 は識別器のドメイン分類損失である． 識別器の学習目標は，実画像の正しい分類を学習するために，実画像のドメイン分類損失を最小化することである．

$$\mathcal{L}_{cls}^{r} = \mathbb{E}_{x,c'}[-\log D_{cls}(c^{'}|x)] \tag{3.8}$$

StarGAN の再構成損失は，実は前節の CycleGAN のサイクル一貫性損失である．

つまり，生成器を 2 回使って，元画像をターゲットドメインの画像に変換し，次に変換された画像から元画像を復元し．最後に元画像と復元画像の L1 正規化距離を計算する．計算式は式 3.9 である．

$$\mathcal{L}_{rec} = \mathbb{E}_{x,c,c'}[||x - G(G(x,c),c^{'})||_1] \tag{3.9}$$

したがって，StarGAN の全損失は，式 3.10 と式 3.11 のように，これら 3 つの損失の組み合わせとなる．

$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls}^{f} + \lambda_{rec}\mathcal{L}_{rec} \tag{3.10}$$

$$\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls}^{r} \tag{3.11}$$

3.4 GANimation

GANimation[5] は，GAN と顔面動作符号化システムを組み合わせた，連続的な表情を生成する初めてのモデルである．モデルは完全に教師なしで訓練され，Attention メカニズムが提案される．このメカニズムにより，モデルは顔領域を集中し，背景部分を可能な限り変更しないようにすることができる．GANimation は StarGAN に基づいており，Generator を 1 つだけ使用し，複数の表情を生成できる．

異なる表現を異なるドメインとして考える StarGAN とは対照的に，GANimation は AU 属性をドメインとして扱い，ドメインの値は 0 と 1 に限定されない．代わりに，ドメインに対応する AU 動作の強度を示すために具体的な値が使用される．このようにして，GANimation は顔の個々のパーツの表情を調整することができる．

図 3.6 は，女性の写真と顔の表情の AU 属性ラベルを示している．

眉を上げたい場合は，眉の上がり具合を制御する AU 属性 AU4 の値を上げるだけで，図 3.7 のように表情が変換される．

変換前の画像

画像のAU属性ラベル

AU01_r	AU02_r	AU04_r	AU05_r	AU06_r	AU07_r	AU09_r	AU10_r	AU12_r	AU14_r	AU15_r	AU17_r	AU20_r	AU23_r	AU25_r	AU26_r	AU45_r
0.45	0.48	0	0	0.93	1.06	0	0.19	1.95	0.94	0	0.31	0	0	1.83	0.8	0

図 3.6: GANimation を用いた表情変換（変換前）

変換後の画像

画像のAU属性ラベル

AU01_r	AU02_r	AU04_r	AU05_r	AU06_r	AU07_r	AU09_r	AU10_r	AU12_r	AU14_r	AU15_r	AU17_r	AU20_r	AU23_r	AU25_r	AU26_r	AU45_r
0.45	0.48	2	0	0.93	1.06	0	0.19	1.95	0.94	0	0.31	0	0	1.83	0.8	0

図 3.7: GANimation を用いた表情変換（変換後）

第4章　提案手法

本研究では，ベースモデルは StarGAN と同じ構造を用いている，このモデルは教師なし学習に対して高い効果を得ることができる．そして，GANimation のモデルを参照して，人物顔の生成と変換タスクで優れた性能を発揮する生成器「Attention-based Generator」を使用して，顔面動作符号化システムの AU 属性を導入する．すなわち，本研究のモデルのドメインの数は，選定された AU 属性の数と同じ，17 個である．そこで，入力された 17 桁の AU 属性値の大きさにより，様々な表情を生成して，微調整を行うことができる．

なお，本研究では，ConvNeXt[6] を用いて，モデルを改善させることで，表情変換後の生成画像の品質を高めている．

4.1　Attention-based Generator

Attention-based Generator では，GANimation に提案されている．この生成器は図 4.1 に示すように，入力画像 I_0 を入力すると，出力層で 2 つに分割され，それぞれカラーイメージ C と Attention イメージ A が出力される．

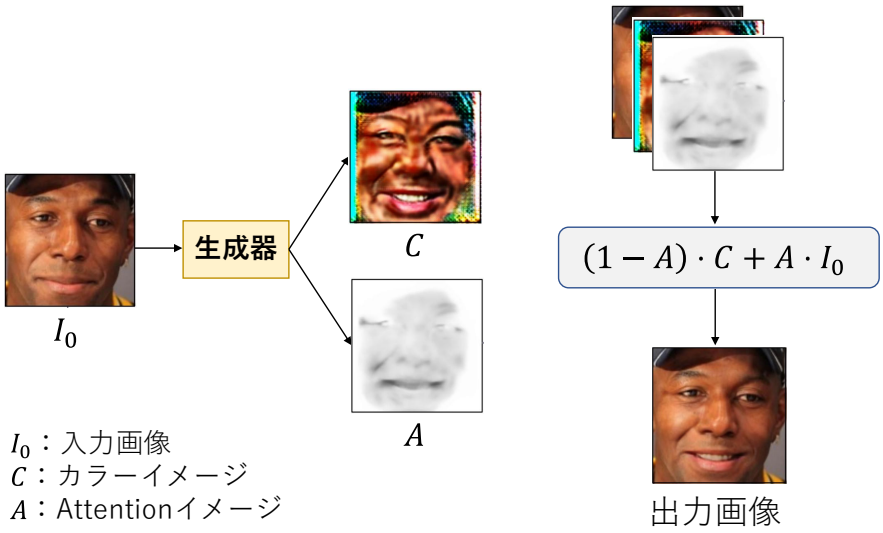

図 4.1: Attention-based Generator

カラーイメージ C は人間の肌に相当し，入力画像の顔の肌の色や特徴などの情報を保

存している．Attention イメージ A とは，表情の動きをコントロールする皮膚の下の筋肉や骨格のようなものであり，表情が変化するときに特定の部分に集中することができる．そして，I_0，C，A を図 4.1 の式のように合成して，最終的な出力画像を得る．このように Attention メカニズムを生成器に導入すると，生成器は入力人物画像の顔を集中して，背景部分の影響を無視することができ，生成された人物画像の目鼻の特徴は入力人物画像と同じとなる．

4.2 ConvNeXt Block

ConvNeXt は，2022 年に提案された新しいタイプの純粋な畳み込みニューラルネットワークであり，より高い精度に加えて，より速い推論を誇っている．本研究では，元モデルの ResNet 残差ブロックの代わりに，ConvNeXt 残差ブロックを使用する．これにより，生成される画像の品質を向上できる．本研究の ResNet 残差ブロックと ConvNeXt 残差ブロックの構造を図 4.2 に示す．

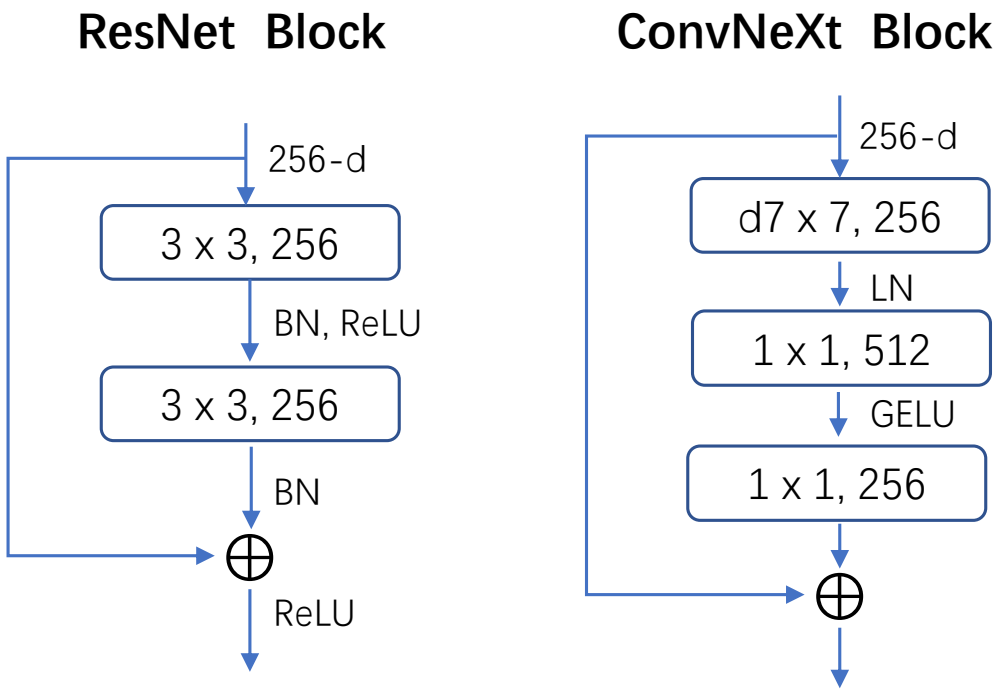

図 4.2: 残差ブロックの構造

4.3　モデルの仕組み

このセクションでは提案手法に用いられているネットワークモデルの仕組みについて説明を行う．

本研究では，生成器の構造は図 4.3 に示す．まずは最初の畳み込みブロックで，次に 2 層のダウンサンプリング層と 6 つの ConvNeXt 残差ブロックと 2 層のアップサンプリング層がある．最後に，出力層は，チャンネル数に応じて 2 つの部分に分割された．チャンネル数が 3 の出力はカラーイメージで，チャンネル数が 1 の出力は Attention イメージである．

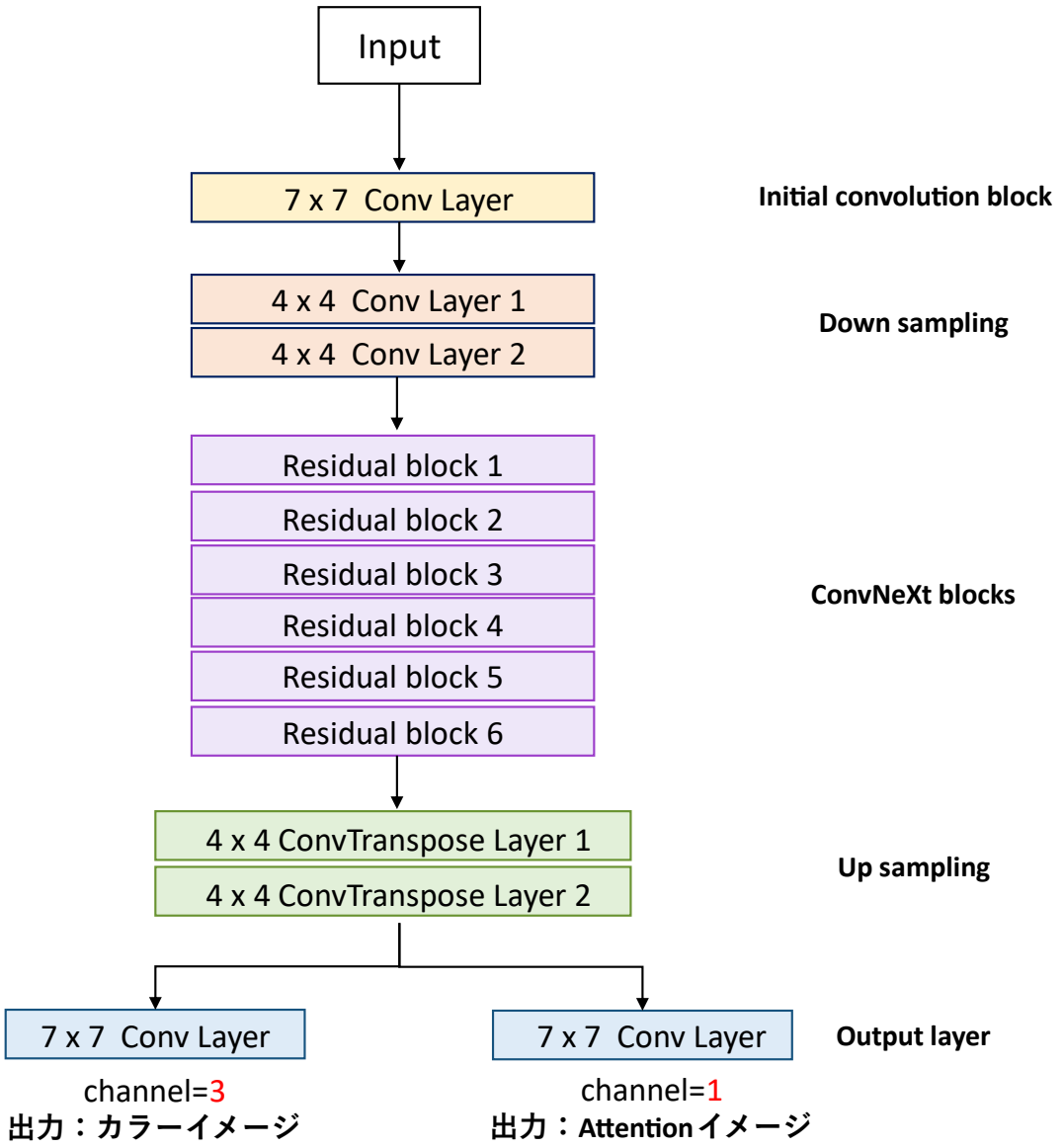

図 4.3: 生成器の構造

識別器の構造は図 4.4 に示す．最初は入力層，次は 6 層の隠れ層，最後の出力層はチャンネル数によって 2 つに分かれて，チャンネル数が 1 の部分は入力画像の真偽判定に，チャンネル数が 17 の部分は入力画像の 17 桁の AU 属性値の識別に使用される．

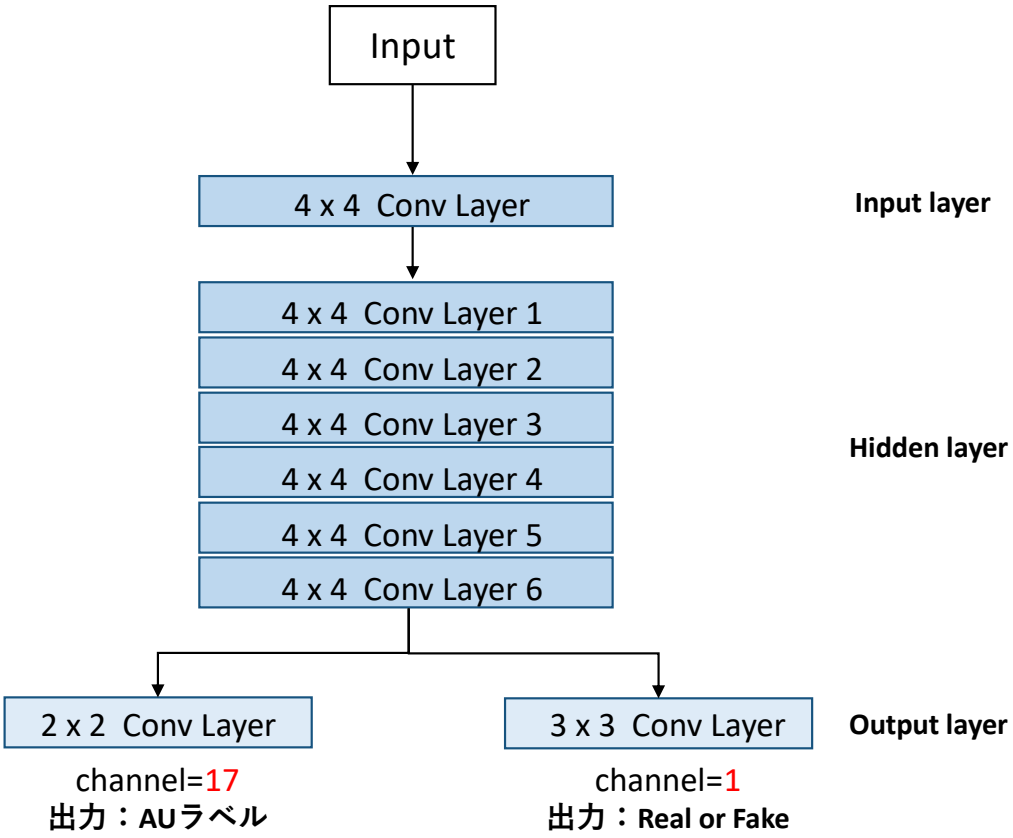

図 4.4: 識別器の構造

4.4 訓練の流れ

本研究のモデルの訓練の流れは図 4.5 に示す．全体の流れは 2 つのステップに分かれる．ステップ 1 は，目標表情の選択である．データセットからランダムに人物画像を選択して目標表情とする．そして，Openface[7] を使用して 17 桁の AU 属性値を抽出してターゲット AU 属性ラベルにする．ステップ 2 は，表情変換の訓練の流れである．まず，入力画像と目標表情の 17 桁の AU 属性ラベルを入力として，生成器を通じて変換された fake image を得る．Fake image が識別器により真偽と 17 桁のドメインの値を判定する．一方，入力画像の AU 属性ラベルと組み合わせて，生成器によって復元画像が得られる．

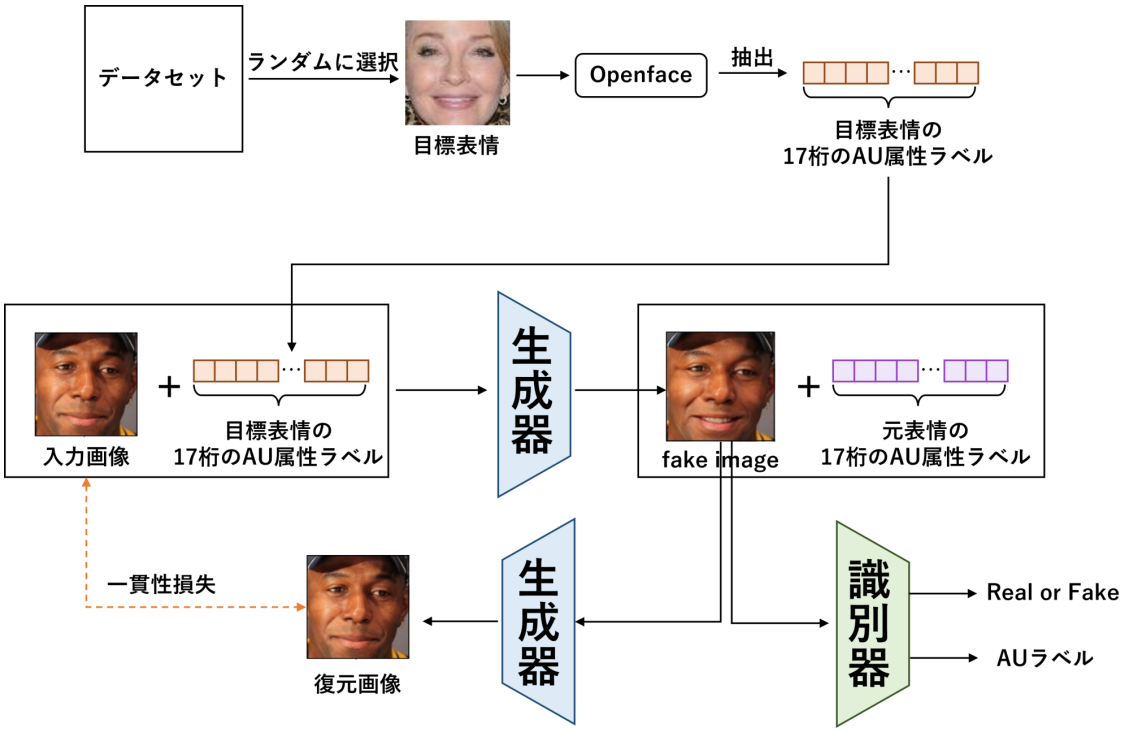
データセット
ランダムに選択
目標表情
Openface
抽出
目標表情の
17桁のAU属性ラベル
入力画像
目標表情の
17桁のAU属性ラベル
生成器
fake image
元表情の
17桁のAU属性ラベル
一貫性損失
復元画像
生成器
識別器
Real or Fake
AUラベル

図 4.5: 訓練の流れ

第5章　データセット

本研究で使用したデータセットは，CelebA[8] と EmotioNet[9] という 2 つのデータセットである．

5.1　CelebA データセット

CelebA データセットには，10,177 人の有名人の 202,599 枚のカラー写真が集められている．データセットに含まれる画像のほとんどは，有名人のイベントに参加する時のライブ写真や雑誌の写真であるため，年齢層が 20 代から 40 代が多く，表情は笑顔または無表情が多いである．CelebA データセットの一部を図 5.1 に示す．サイズは縦 218 ピクセル，横 178 ピクセルである．

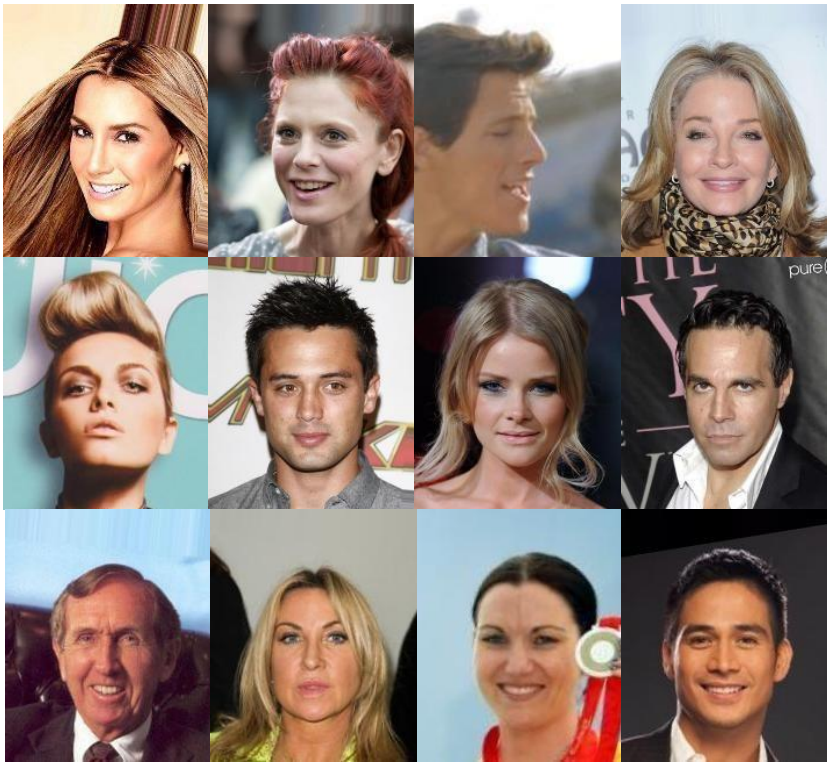

図 5.1: CelebA データセットの一部

5.2 EmotioNet データセット

EmotioNet データセットはオハイオ州立大学からのデータセットで，さまざまな年齢層，人種，表情の数百万枚の人物画像が含まれている．データセットを使用するには，メールでリクエストを送信する必要がある．データセットの内容は，画像のダウンロードアドレスを含む複数のテキストファイルである．本研究では，Python クローラーを用いて 346,004 枚の画像をダウンロードした．データセットの一部を図 5.2 に示す．

図 5.2: EmotioNet データセットの一部

CelebA データセットは基本的に大人の笑顔か無表情な顔であるのに比べ，EmotioNet データセットは格段に多様性に富んでいる．

5.3 前処理

前述の収集したデータセットでは，画像サイズを統一し，顔領域を集中するために，以下の 2 つのステップの前処理が必要である．

5.3.1 顔抽出

ステップ 1 は人物画像の顔の抽出である．図 5.3 に示すように，人物画像は顔認識ライブラリ Face Recognition[10] によって，顔を検出して，顔部分の正方形の位置を決めて，そしてカットする．カットした顔画像は，縦横 128 ピクセルにリサイズされる．この処理により，背景部分の影響を軽減し，顔の部分に焦点を当てた画像を得ることができる．

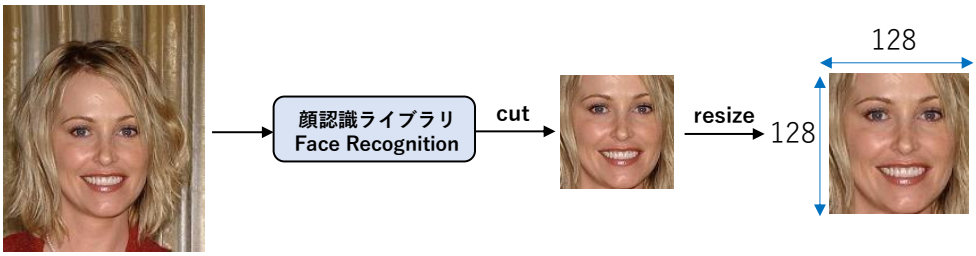

図 5.3: Face Recognition で顔抽出

5.3.2 AU 属性ラベルの作成

ステップ 2 は AU 属性ラベルの作成である．

本研究では，画像だけではなく，画像の AU 属性ラベルを組み合わせて学習処理を行う必要があるため，事前に AU 属性ラベルを作成しておく必要がある．

図 5.4 に示すように，カットした顔画像は Openface[7] を用いて 17 桁の AU 属性ラベルを作る．作成した AU 属性ラベルと画像は，最終的に完全なデータセットとなる．

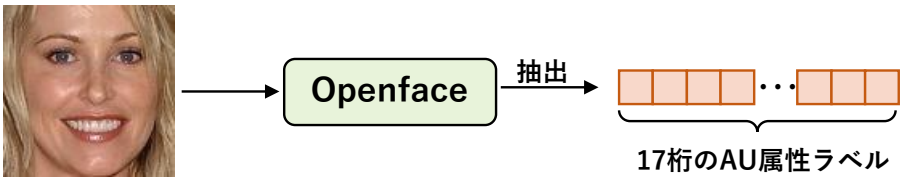

図 5.4: Openface で AU 属性ラベルを作る

5.4 顔抽出方法の改善

本研究では，筆者は元の顔抽出方法を最適化することで，異なる試しを行った．

最初の顔抽出方法を図 5.5 に示す．まず，縦 218 ピクセル，横 178 ピクセルの元画像の中心点を見つけて，縦横 178 ピクセルの正方形の画像をカットして，縦横 128 ピクセルのサイズにリサイズする．

しかし，この方法には 2 つの欠点がある．

まず，顔が常に画像の中心にあるとは限らない．顔が画像の上端や下端に近い場合，カットする時に顔の一部がカットされる可能性が高い．

二つ目は，カットした顔は画像全体の 50 %以下しか占めていない．訓練しながら微細な表情の特徴を学ぶのは難しい．

これらの欠点を解決し，カットした画像をより顔の部分にフォーカスさせるために，筆者は最初に図 5.6 に示すような顔抽出方法を設計した．

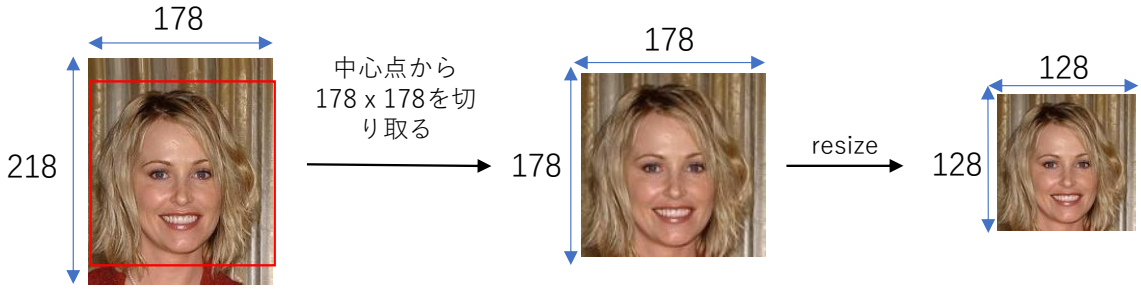

図 5.5: 元の顔抽出方法

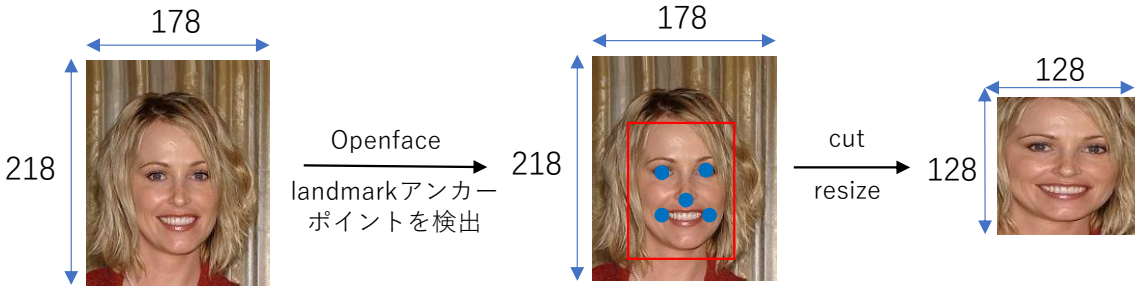

図 5.6: 自作の顔抽出方法 Ver1.0

この方法では，Openface ツールを使用して，画像の landmark アンカーポイントを検出する． landmark アンカーポイントは顔の左目，右目，鼻，左口角，右口角の 5 点の座標値である．そして，landmark アンカーポイントに基づいて顔部分をカットして，最後に縦横 128 ピクセルの正方形画像をリサイズする．

しかし，この方法でカットした顔画像は必ずしも正方形ではないため，顔画像を正方形にリサイズすると顔が引き伸ばされる．そこで，筆者は 2 回目の顔抽出手法最適化を行った．図 5.7 に示すように，Python の Face Recognition ライブラリを使用すると，顔の位置が検出され，自動的に正方形に正規化される．このようにしてカットされ，リサイズされた顔画像は，顔が画像の 90 %以上を占め，形が引き伸ばされることがない．

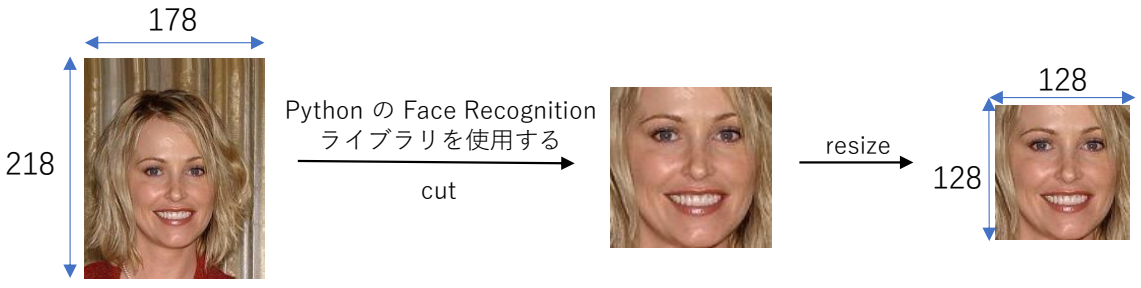

図 5.7: 自作の顔抽出方法 Ver2.0

顔抽出処理後のデータセットの一部を図 5.8 に示す.

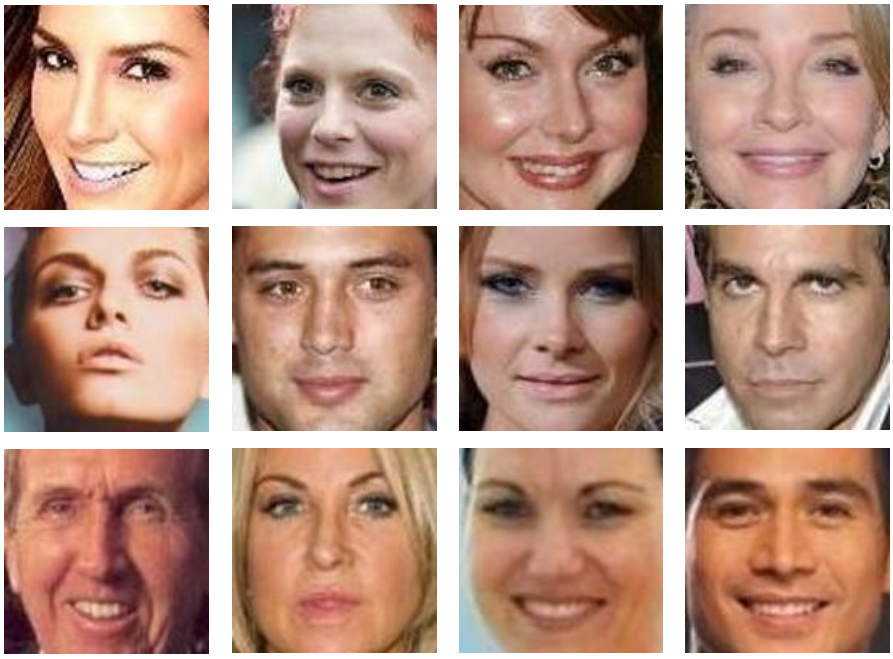

図 5.8: 顔抽出処理結果の一部

第6章　実験

本章では提案モデルを使ったCelebAデータセットとEmotioNetデータセット上の実験結果について紹介する．本実験では最適化アルゴリズムにはAdamを使用し，初期学習率は0.0001，エポック数は50である．

6.1　生成実験

生成実験の目標は，入力画像の表情を目標画像の表情に変換し，その中間生成結果を段階的に出力して，最終的にはモーフィングのような変換効果を得ることである．

ResNetとConvNeXtの生成効果を比較するために，本実験は操作変数法を使用して四つの対照実験を設計した．図6.1に示すようにCelebAデータセットとResNet残差ブロックを用いた実験1，図6.2に示すようにCelebAデータセットとConvNeXt残差ブロックを用いた実験2，図6.3に示すようにEmotioNetデータセットとResNet残差ブロックを用いた実験3，図6.4に示すようにEmotioNetデータセットとConvNeXt残差ブロックを用いた実験4である．

実験1と実験2の結果を比較すると，実験1の表情変換タスクはすべて成功したが，生成された画像の細かい部分，たとえば歯の部分は特にはっきりしなかった．歯と歯の間の隙間は目立たない．対照的に，実験2の結果はノイズが少ない．従って，CelebAデータセット上に，ResNetよりもConvNeXtの訓練効果が良いことが分かる．

実験3と実験4の結果を比較すると，実験3の結果にはノイズも見られたが，実験4はより高品質な画像が得られる．そして，実験4の結果の4行目によって，唇をすぼめた状態からニヤッと笑った状態への細かな変換も完璧に実現できるため，EmotioNetデータセットでも，ConvNeXtはResNetよりも性能が良いことが分かる．

6.2　評価実験

人間の目による評価には主観が含まれる可能性がある，ResNet残差ブロックとConvNeXt残差ブロックの効果をより客観的に比較するために，このセクションでは機械判定の方法を用いて生成画像の品質をスコア化する．

この評価実験では，EmotioNetデータセットを使用し，ResNetとConvNeXtがそれぞれ生成した3万枚の画像と，元のEmotioNetデータセット内の3万枚の画像を選択

図 6.1: 生成実験 1

する．評価方法は IS[11] (Inception Score) と FID[12] (Fréchet Inception Distance) を用いる．

IS の場合はスコアが高いほど，FID の場合はスコアが低いほど良い結果になる．評価実験の結果を表 6.1 に示す．いずれも ConvNeXt の生成結果はもっと良いことが分かる．

表 6.1: IS と FID の評価採点

Residual Block	IS	FID
ResNet	5.8021	5.9514
ConvNeXt	**5.1927**	**5.8933**

6.3 モデルの最適化実験

本研究ではモデルのさらなる最適化を試みた．本研究の中で置き換えた ConvNeXt 残差ブロックの構造を図 6.5 に示すが，中間の畳み込み層のチャンネル数が前層より 2 倍に拡張され，活性化関数 GELU と正則化関数 Layer Normalization がそれぞれ 1 回ずつ使用されている．

最適化し続けたい場合には，2 つの改造方向がある．一つは，図 6.6 に示すように，中間の畳み込み層のチャンネル数の拡張倍率を上げることである．例えば，4 倍拡張，6 倍

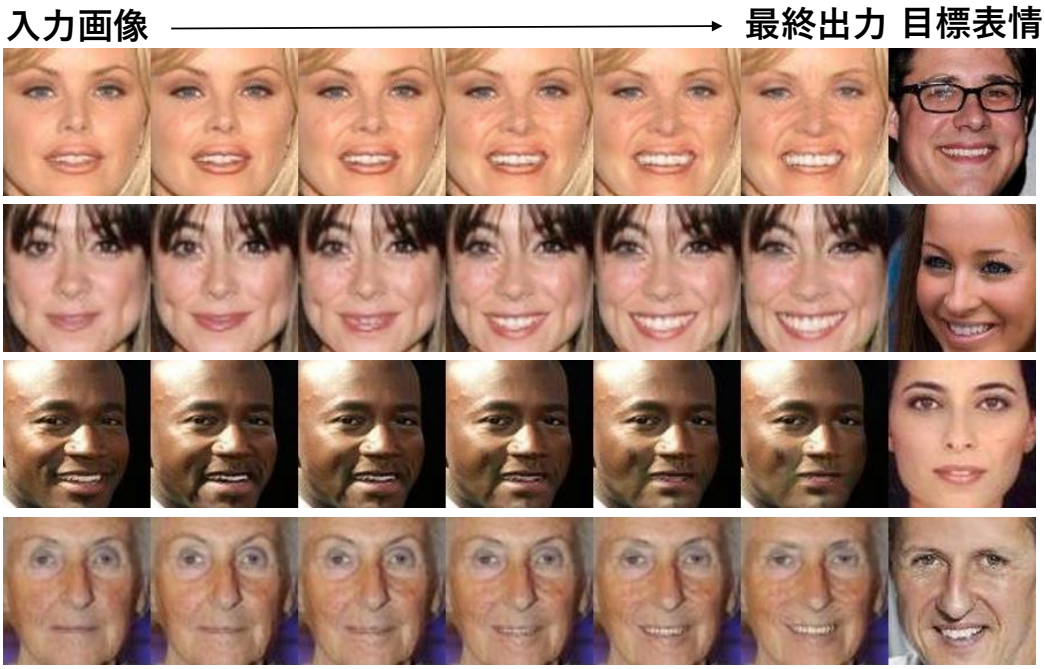

図 6.2: 生成実験 2

拡張などである.

もう一つは，図 6.7 に示すように，正則化関数 Layer Normalization と活性化関数 GELU の使用頻度を上げることである.

これらの改造を基に，訓練とテストが行われ，生成された結果は図 6.8 と図 6.9 に示す.

異なる最適化手法の生成結果を比較すると，目と口を閉じた表情に変換する場合,「中間層 6 倍拡張」モデルが最も良いことがわかる.

その理由は，他のモデルの生成結果では唇に小さな白い点が生じるが，実は歯が完全に除去されていないことによるノイズである.

IS と FID の評価結果を表 6.2 に示す. 生成実験の結果と合わせると，モデルの ConvNeXt 残差ブロックの中間の畳み込み層のチャンネル数を 6 倍に拡張したとき，最善の結果が得られることが分かる.

表 6.2: 異なる最適化手法の生成結果の機械評価採点

モデル	IS	FID
中間層 2 倍拡張	**5.1927**	5.8933
中間層 4 倍拡張	4.9073	5.8911
中間層 6 倍拡張	5.1881	**5.7718**
LN と GELU の使用頻度増加	5.1544	5.7792

図 6.3: 生成実験 3

図 6.4: 生成実験 4

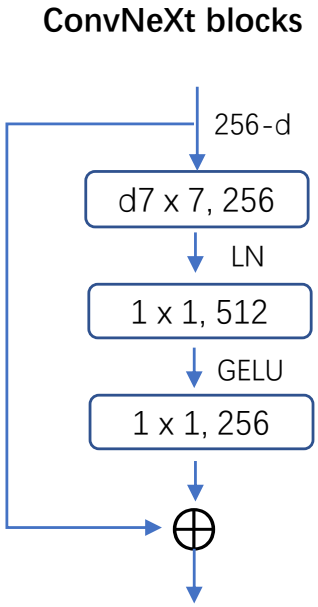

図 6.5: 元の ConvNeXt blocks の構造

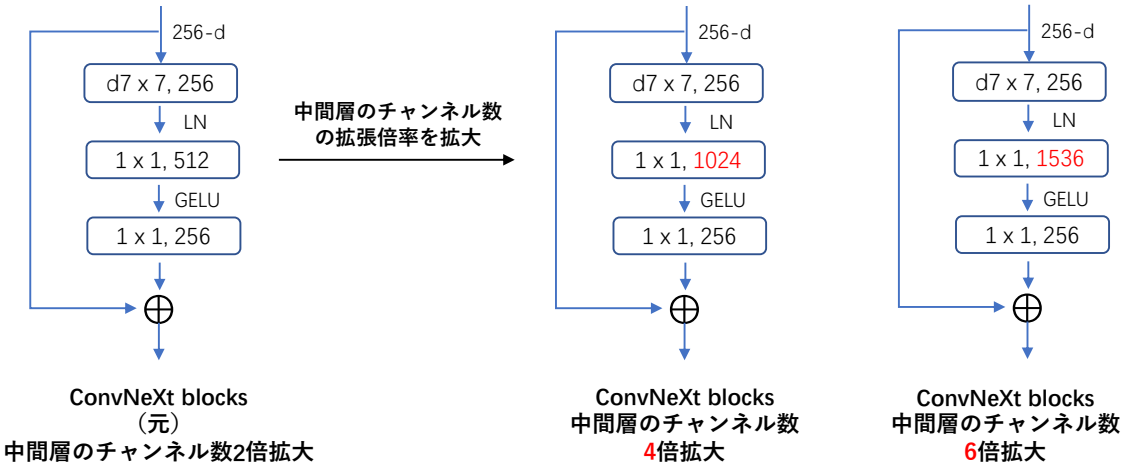

図 6.6: モデルの最適化手法 1

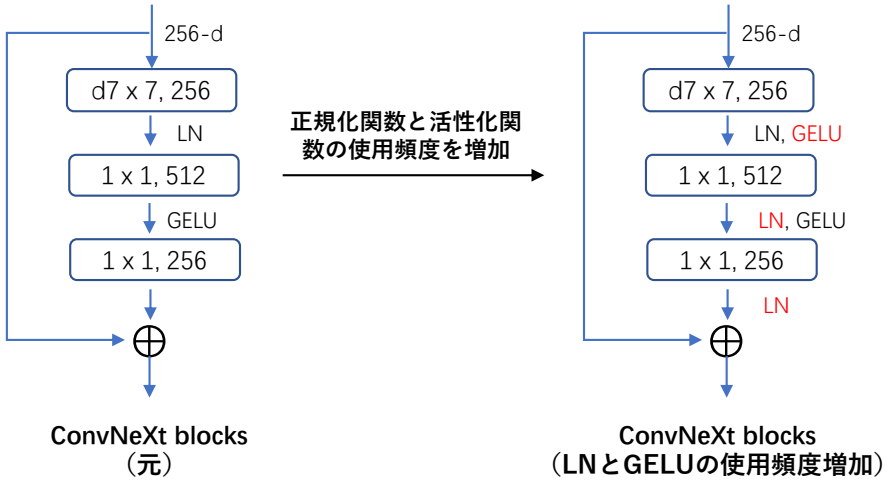

図 6.7: モデルの最適化手法 2

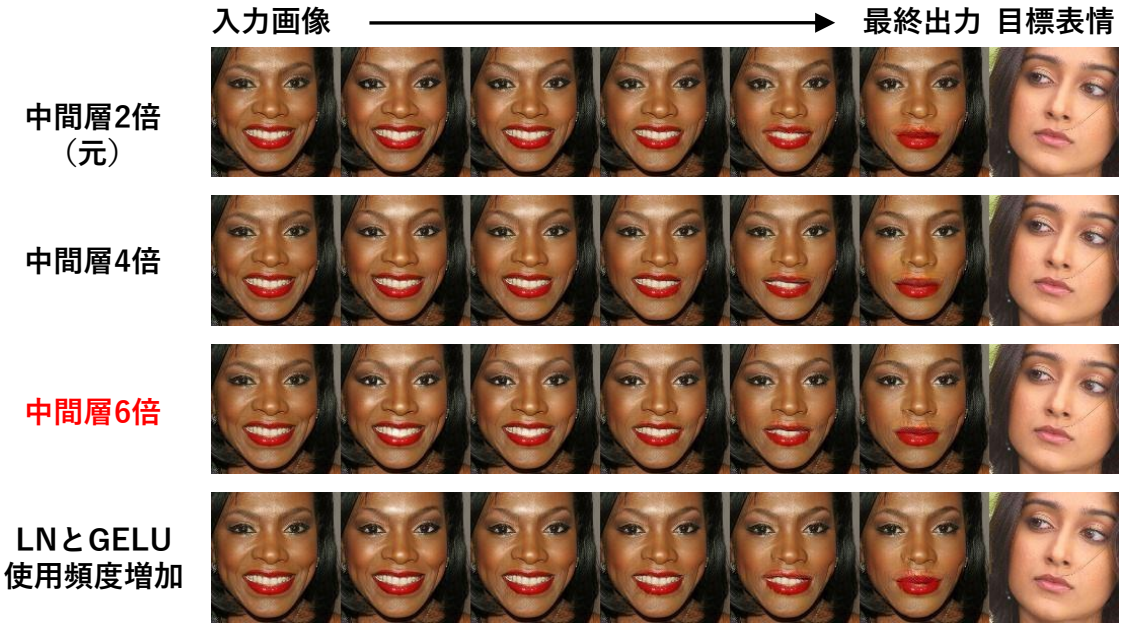

図 6.8: 異なる最適化手法の生成結果 1

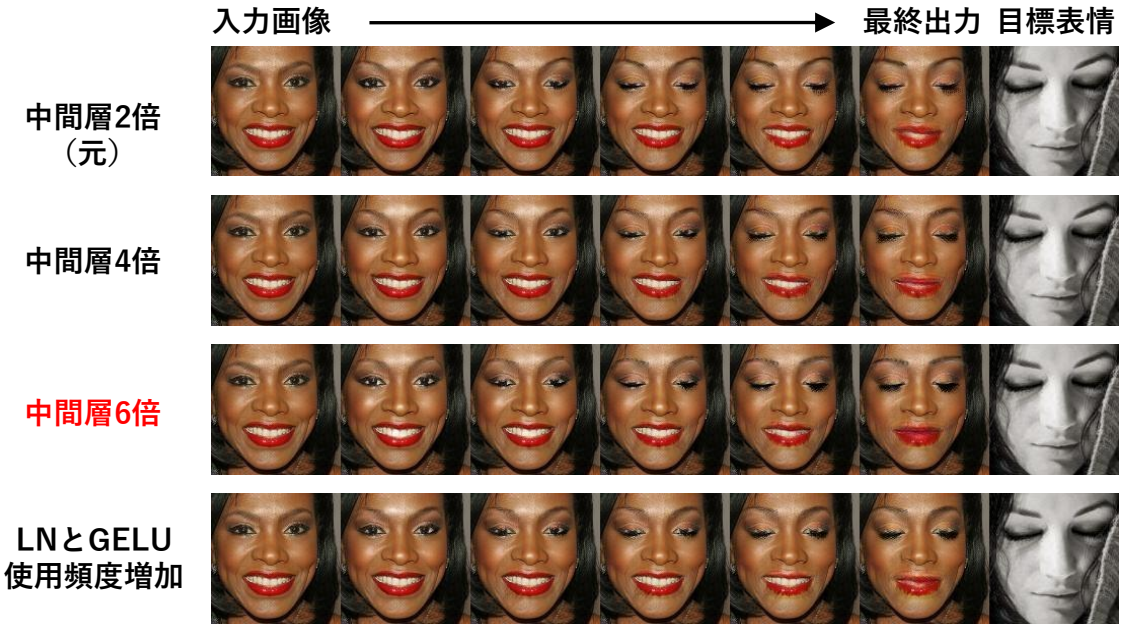

図 6.9: 異なる最適化手法の生成結果 2

第7章　まとめ

本研究では，人物画像の表情を自由に変換することを目的としている，画像処理タスクにおける表情変換をさらに発展させた．StarGAN のベースラインには，顔面動作符号化システムと組み合わせて，入力人物画像から多様な表情を生成できる GANimation を採用した．既存のデータセットに自身で処理して，ConvNeXt モデルを改善して，顔写真の単一領域の表情の微調整と段階的な変換効果を実現した．

本研究では，市販の自撮り写真加工アプリケーションとは異なり，表情を全体的に変換するのではなく，AU 属性値を制御することで，顔の異なる領域の各筋肉の変化を組み合わせて表情の変換を実現する．こうすることで既存の写真加工アプリを改善することができる．また，本研究はモデルの改良の継続の可能な方向を提供し，人間の目による主観的な評価と機械による客観的な評価を通じて，改善したモデルの有効性を確認していた．

今は人間写真から人間写真への表情変換だけで，人間の顔からアニメ風の表情変換もやってみたいと思っている．しかし，アニメ画像の顔の表情を抽出して学習させるのは非常に難しく，あまり良い実験結果にはならない．そのため，今後はこのモデルを改良し，人間写真の顔からアニメ画像の顔の表情を学習できることを目指す．

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中野浩嗣教授，伊藤靖朗教授，北須賀輝明准教授，高藤大介助教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] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.

[2] 顔写真加工アプリ, Available: https://play.google.com/store/apps/details?id=io.faceapphl=jagl=US.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza *et al.*, "Generative adversarial nets. arxiv website. arxiv. org/abs/1406.2661," *Published June*, vol. 10, 2014.

[4] 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*, 2017, pp. 2223–2232.

[5] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "GANimation: Anatomically-aware facial animation from a single image," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 818–833.

[6] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 976–11 986.

[7] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Openface: an open source facial behavior analysis toolkit," in *2016 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2016, pp. 1–10.

[8] Z. Liu, P. Luo, X. Wang, and X. Tang, "Large-scale celebfaces attributes (CelebA) dataset," *Retrieved August*, vol. 15, no. 2018, p. 11, 2018.

[9] C. Fabian Benitez-Quiroz, R. Srinivasan, and A. M. Martinez, "EmotioNet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5562–5570.

[10] D. S. Trigueros, L. Meng, and M. Hartnett, "Face Recognition: From traditional to deep learning methods," *arXiv preprint arXiv:1811.00116*, 2018.

[11] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*. PMLR, 2019, pp. 7354–7363.

[12] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.