Vol.2021-NL-248 No.25
Vol.2021-CVIM-226 No.25
2021/5/21

顔画像の属性編集

吉武 佑汰[1,a)]　延原 章平[1,2]　西野 恒[1]

概要：顔画像の属性を編集する手法として，敵対的生成ネットワーク（GAN）を用いたものが多く提案されている．特に近年では，無条件 GAN の潜在表現を操作することで編集を行う手法が注目されている．しかしこの手法は，例えば眼鏡を掛ける編集を行うとそれにつられて年齢が老けてしまうというように，編集で指示されていない特徴まで変更してしまうことが多い．そこで本研究では，編集対象ではない属性の変更を防ぐ属性維持損失と，骨格や姿勢の変更を防ぐ骨格損失によって訓練した深層ニューラルネットワークを用いて潜在表現を操作する．評価実験では，複数の人物の顔画像に対して様々な属性の編集を行い，その結果を評価することで本手法の有効性を示す．

1. 序論

顔画像の属性編集とは，性別や髪色といった人間の顔の属性を編集することをいう．顔画像は身の回りの様々な場面で活用されており，顔画像を簡単に，かつ自在に編集できるようにすることは多くの貢献をもたらす．一例として，デザイナーの支援が挙げられる．ポスター広告等に使われる顔画像が決定されるまでの過程では，何度も試作が行われ，その度にモデルの手配や再撮影等の膨大な作業が生じる．手持ちの画像を自在に編集できるようになれば，こうした作業が削減され，デザイナーはより創造的な活動に時間を使えるようになる．

顔画像の属性を編集する既存手法として，敵対的生成ネットワーク（GAN）を用いたものが多く挙げられる．特に近年では，単に本物らしい画像を生成するようにだけ訓練した無条件 GAN の潜在表現に変化を加えることで編集を行う手法が注目されている．潜在表現とは，GAN の生成器に入力するランダムノイズのことである．この手法は非常に高品質な結果を得られるが，編集で指示されていない属性や，顔の骨格，姿勢まで変更させてしまうという問題がある．この原因は，編集内容に対応する適切な変化を求めることが難しく，潜在表現に誤った変化が加えられるためである．

そこで本研究では，新たに導入する属性維持損失と骨格損失によって訓練した深層ニューラルネットワークを用いて潜在表現に加える変化を求める．属性維持損失は編集対象外の属性が変化した量を表し，ネットワークが編集対象

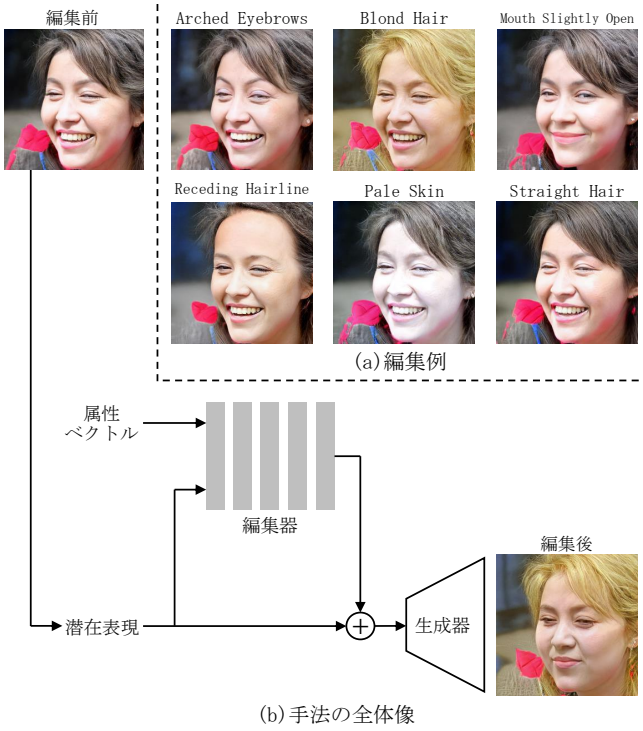

図 1　本手法から得られた編集結果と，手法の全体像．本手法では，画像の潜在表現に変化を加えることで編集を行う．加える変化は深層ニューラルネットワークである編集器により計算する．右下の画像は Blond Hair を付与して Mouth Slightly Open と Smiling を排除する，複数属性の編集を行った結果である．

でない属性を変化させることを防ぐ．骨格損失は目や鼻といった顔の特徴点の位置が変化した量を表し，ネットワークが骨格や姿勢を変化させることを防ぐ．提案する損失を用い，様々な顔画像と編集内容に対してネットワークを訓練することで，編集対象の属性だけを変更する正確な編集

1　京都大学大学院 情報学研究科
2　JST さきがけ
a)　yyoshitake@vision.ist.i.kyoto-u.ac.jp

が可能となる．なお本研究においては，各属性に加える編集内容を属性ベクトルで表現し，これをネットワークへの入力とする．

評価実験では，訓練したネットワークを用いて顔画像編集を行い，その結果を定性的，定量的に評価した．定性的な評価は，複数の人物の顔画像に対して様々な属性を編集することで行った．定量的な評価は，編集前と編集後の顔の類似度を指標として行った．加えて，属性維持損失と骨格損失を用いず訓練した場合と，用いて訓練した場合を比較することで，これらの損失による編集結果の改善を確認した．評価実験から，提案手法によって多様な属性を正確に編集できることが示された．

本研究は，属性ベクトルによって編集内容を指示し，編集対象の属性のみを編集することを可能とした．特に，複数の属性を同時に変更する複雑な編集も正確に行うことが可能となった．これは手軽で自在な顔画像編集の実現につながり，様々な創作活動の支援となることが期待される．

2. 関連研究

近年，画像を生成する強力な手法として，敵対的生成ネットワーク (Generative Adversarial Networks:GAN)[1], [2] が注目されている．GAN は，潜在表現と呼ばれるランダムノイズから画像を生成する生成器と，与えられた画像が実在する画像であるか，生成器により生成された画像であるかを見分ける識別器の二つのニューラルネットワークからなる．生成器と識別器を競わせる形で訓練することで，本物らしい画像を生成する生成器を得ることができる．より自然で高画質な画像を生成するよう改良された手法 [3], [4], [5] は，本物の画像と区別がつかないほど高品質な画像を生成できる．しかし，これらの手法は単に画像を生成するようにだけ訓練された無条件 GAN であり，画像編集等，生成画像の制御を必要とする作業はできない．

条件付敵対的生成ネットワーク (Conditional Generative Adversarial Networks:CGAN)[6] は，生成器への入力に条件を追加することで，生成する画像を制御できるよう改良した手法であり，様々な応用研究が行われている．例として，CGAN を用いた画像変換の手法 [7], [8] が挙げられ，これらの手法を用いることで顔画像を編集することが可能である．しかし，得られる編集結果の品質は無条件 GAN の生成画像と比較すると劣ったものである．

CGAN による画像変換は用いずに，潜在表現に変化を加えることで顔画像を編集する手法も提案されている．Shen らは，男性，女性といった二元的な属性についての境界面を潜在空間内で求め，境界面に対して垂直に潜在表現を動かすことで顔画像を編集する手法を提案した [9]．この手法は，使用する生成器自体が生成画像を制御する必要はなく，無条件 GAN の生成器を用いることで高品質な結果を得ることができる．しかし，この手法はしばしば編集対象でない属性まで変化させてしまうという問題がある．本研究では，対象外の属性を変化させないように訓練した深層ニューラルネットワークを用いることで，編集による意図しない属性の変化を抑えることができる．

3. 提案手法

本研究では，顔画像の任意の属性を編集するため，与えられた属性ベクトルにより画像の潜在表現に変化を加える深層ニューラルネットワークを導入する．このネットワークを，潜在表現を編集するネットワークとして，編集器と名付ける．編集器の訓練には，編集対象である属性の変更に対する損失（属性変更損失）に加え，編集対象外である属性の維持に対する損失（属性維持損失）と骨格や姿勢の維持に対する損失（骨格損失）を用いる．また，潜在表現から画像を生成するため，顔画像を生成するように事前訓練した無条件 GAN の生成器を用いる．

手法の全体像を図 1 に示す．編集器は，入力された属性ベクトルから，編集前の画像の潜在表現と編集後の画像の潜在表現の差分を出力する．差分を加えた潜在表現を生成器に入力することで，編集後の画像を得る．

本研究では編集対象として，二種類の画像を想定する．一つ目は生成器によって生成した，実在しない人物の顔画像（生成画像）である．生成画像は，その画像を得る際に生成器に入力した潜在表現に変化を加えることで編集する．二つ目は実在する人物の顔画像（実画像）である．実画像は，潜在空間へ投影することで得られた潜在表現に変化を加えることで編集する．

3.1 顔画像の潜在表現

潜在表現から顔画像を生成するため，無条件 GAN の生成器 G を用いる．GAN の生成器 G は潜在表現 $\boldsymbol{z}$ を入力として，画像 I を生成する．すなわち，

$$I = G(\boldsymbol{z}) \tag{1}$$

である．潜在表現 $\boldsymbol{z}$ は，潜在空間と呼ばれる空間内の確率分布からランダムにサンプリングされたベクトルであり，生成される顔画像の特徴に関する情報を持つ．入力する潜在表現 $\boldsymbol{z}$ が変化した場合，生成器 G が生成する画像 I も変化する．そのため，編集対象の画像を出力できる潜在表現 $\boldsymbol{z}$ に適切な変化を加え，生成器 G に入力することで，意図した変化が加えられた画像 I' を得ることができる．

実画像を編集する場合には，編集対象である実画像に最も近い画像を生成できる潜在表現を推定し，推定した潜在表現に変化を加えることで編集を行う．この推定は，生成器 G の重みを固定し，実画像と生成された画像の間の誤差が最小になるよう，生成器 G に入力する潜在表現を最適化することで行う．

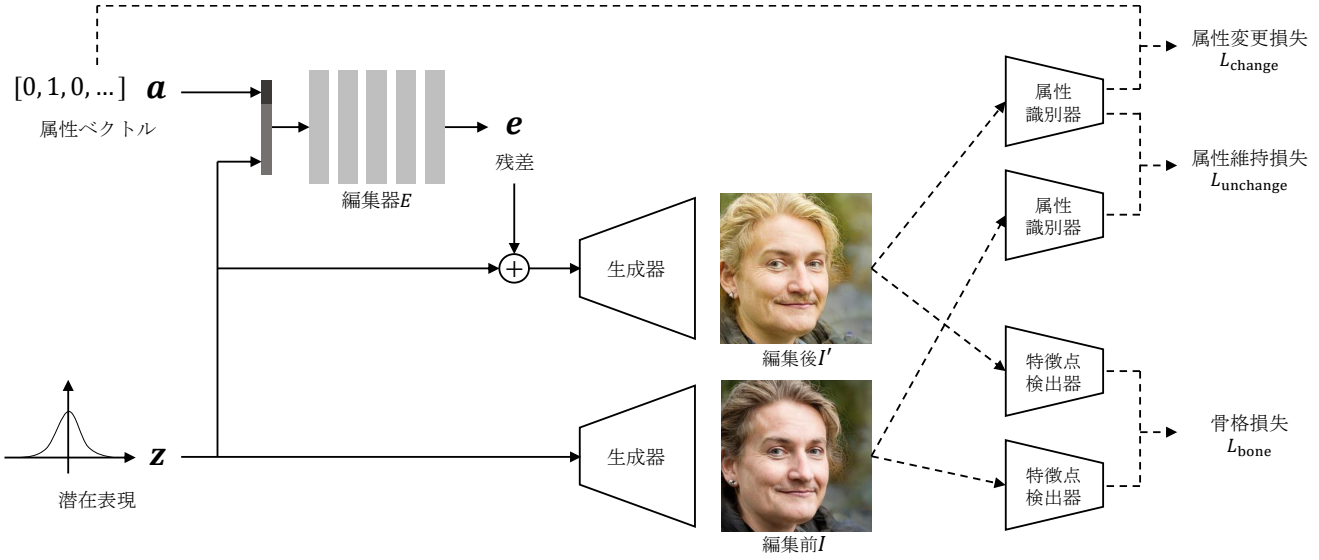

図 2 編集器の訓練の全体像．訓練中，編集器はランダムにサンプリングされた潜在表現に対し，ランダムに決定された属性ベクトルで指示される編集を行う．編集前後の画像から，属性変更損失と属性維持損失，骨格損失を計算して編集器の重みを更新する．

表 1 編集対象とした顔画像の属性の一覧．

5 o'Clock Shadow	Chubby	Pale Skin
Arched Eyebrows	Double Chin	Pointy Nose
Attractive	Eyeglasses	Receding Hairline
Bags Under Eyes	Goatee	Rosy Cheeks
Bald	Gray Hair	Sideburns
Bangs	Heavy Makeup	Smiling
Big Lips	High Cheekbones	Straight Hair
Big Nose	Male	Wavy Hair
Black Hair	Mouth Slightly Open	Wearing Earrings
Blond Hair	Mustache	Wearing Hat
Blurry	Narrow Eyes	Wearing Lipstick
Brown Hair	No Beard	Young
Bushy Eyebrows	Oval Face	

3.2 編集内容の表現

本研究では，編集内容を指示する入力として，属性ベクトル $\boldsymbol{a}$ を定義する．属性ベクトル $\boldsymbol{a}$ は髪色や性別を始めとする顔の属性についての編集内容を表す．属性ベクトルの各要素は一つの属性に対応しており，$\{-1, 0, 1\}$ のいずれかの値を持つ．この値は，編集によって対応する属性をどう変化させるかを表す．具体的には，編集によって対応する属性を付与する場合は 1，排除する場合は 0，変化させない場合は -1 が各要素の値となる．

編集対象とする具体的な属性を表 1 に示す．ここで，例えば Black Hair に編集することと Gray Hair に編集することを同時に指定するような，矛盾した属性ベクトルは入力として考慮しない．そのため，依存関係にある属性を編集する際にはその中の一つに対応する要素にのみ 1 を，残りの属性に対応する要素には 0 を割り当てる．

依存関係にある属性は，髪色を表す Black Hair，Blond Hair，Brown Hair，Gray Hair の四種類，前髪のスタイルを表す Bald，Bangs，Receding Hairline の三種類，髪質を表す Straight Hair，Wavy Hair の二種類である．加えて，髭を表す属性も，三種類の髭と 5 o'Clock Shadow と No Beard の六種類が依存関係にある．三種類の髭とは Goatee，Mustache，Sideburns のことで，これらの間に依存関係はない．そのため，5 o'Clock Shadow または No Beard を編集する場合には残りの五種類に対応する要素を 0 とし，三種類の髭のいずれかを編集する場合には 5 o'Clock Shadow と No Beard に対応する要素を 0 とする．

3.3 潜在表現の操作

入力された属性ベクトルを元に，潜在表現 $\boldsymbol{z}$ に適切な変化を加えることで編集を行う．潜在表現に加えられる変化は差分 $\boldsymbol{e}$ として表現され，編集器 E によって得る．編集器 E は全結合層からなるネットワークで，属性ベクトル $\boldsymbol{a}$ と潜在表現 $\boldsymbol{z}$ が連結されたベクトルを入力として受け取り，必要な差分 $\boldsymbol{e}$ を出力する．差分 $\boldsymbol{e}$ を加えた潜在表現 $\boldsymbol{z}'$ を生成器 G に入力することで，編集結果の画像 I' を得る．つまり，操作された潜在表現 $\boldsymbol{z}'$ は，

$$\boldsymbol{z}' = \boldsymbol{z} + \boldsymbol{e} = \boldsymbol{z} + E(\boldsymbol{z}, \boldsymbol{a}) \tag{2}$$

と表す事ができる．また，編集結果の画像 I' は，

$$I' = G(\boldsymbol{z}') \tag{3}$$

と表す事ができる．

3.4 編集器の訓練

編集器を正確な編集ができるように訓練するため，属性変更損失 $\mathcal{L}_{\text{change}}$，属性維持損失 $\mathcal{L}_{\text{unchange}}$，骨格損失 $\mathcal{L}_{\text{bone}}$ を導入する．属性変更損失 $\mathcal{L}_{\text{change}}$ により，編集対象の属

性に変更を加えるよう編集器を訓練する．属性維持損失 $\mathcal{L}_{\text{unchange}}$ により，編集器が編集対象外の属性に変化を加えることを防ぐ．骨格損失 $\mathcal{L}_{\text{bone}}$ により，編集器が目や鼻といった特徴点の位置を変化させることを防ぐ．編集器の訓練に用いる損失関数 $\mathcal{L}_{\text{total}}$ はこれらの重み付き和，

$$\mathcal{L}_{\text{total}} = \lambda_c \mathcal{L}_{\text{change}} + \lambda_u \mathcal{L}_{\text{unchange}} + \lambda_b \mathcal{L}_{\text{bone}} \tag{4}$$

で定義する．λ_c，λ_u，λ_b は各損失の比率を調節するハイパーパラメータである．

編集器の訓練の全体像を図 2 に示す．生成器，属性識別器，特徴点検出器には事前に訓練したものを用いる．編集器の訓練は，潜在空間の確率分布からランダムにサンプリングされた潜在表現 $\boldsymbol{z}$ と，ランダムに決定した属性ベクトル $\boldsymbol{a}$ によって行う．編集器は入力を受けて必要な差分 $\boldsymbol{e}$ を出力し，編集前の潜在表現 $\boldsymbol{z}$ に差分 $\boldsymbol{e}$ を加えて編集後の潜在表現 $\boldsymbol{z}'$ を求める．そして元の潜在表現 $\boldsymbol{z}$ と，編集後の潜在表現 $\boldsymbol{z}'$ を生成器に入力し，編集前と編集後の画像を生成する．得られた画像からそれぞれの損失を計算し，損失が小さくなるように編集器の重みを更新する．この時，生成器，属性識別器，特徴点検出器の重みは更新しない．

自在な編集を行うため，多様な組み合わせの属性ベクトル $\boldsymbol{a}$ に対応できるよう，編集器を訓練する必要がある．そのため，訓練中には編集する属性の数とその種類をランダムに決定した属性ベクトルを生成する．具体的には，編集可能な全属性の中から m 個の属性をランダムに選び出し，選び出した属性を付与するか削除するかをランダムに決定する．付与する属性には 1 を，排除する属性には 0 を割り当て，編集対象とならなかった属性には -1 を割り当てる．ここで，一度に編集対象とする属性の数 m が多いと学習が困難になるため，m はある上限値 m_u 以下となるようにする．また，3.2 節で述べた矛盾した属性ベクトル $\boldsymbol{a}$ を生成しないようにする必要がある．そのため，m 個の属性を選ぶ際，依存関係にある複数の属性は，その中のただ一つを編集対象に選び，編集対象となった属性には 1 を，残りの属性には 0 を割り当てる．

3.4.1 属性変更損失

編集器が属性ベクトルで指示された属性を変更するため，属性変更損失 $\mathcal{L}_{\text{change}}$ を導入する．ここで，編集器が加えた変更を評価するため，顔画像に含まれる属性を識別する属性識別器 f を導入する．属性識別器 f は顔画像を入力として受け取り，N 個の属性に関する識別結果 $\boldsymbol{p} \in \mathbb{R}^N$ を出力する．すなわち，

$$\boldsymbol{p} = (p_1, p_2, \ldots, p_N) = f(I) \tag{5}$$

である．p_i は i 番目の要素に相当する属性が入力した顔画像に含まれる確率を表す．

編集対象である属性の集合を T，編集後の画像における属性 i の識別結果を p'_i とする．この時，属性変更損失 $\mathcal{L}_{\text{change}}$ は識別結果 p'_i と属性ベクトル $a_i \in \{0,1\}$ の交差エントロピー誤差を全ての属性 $i \in T$ について平均した，

$$\mathcal{L}_{\text{change}} = \frac{1}{m} \sum_{i \in T} \{\mu_i a_i \log p'_i + \nu_i (1 - a_i) \log(1 - p'_i)\} \tag{6}$$

で定義する．m は編集対象である属性の総数を表す．また，μ_i と ν_i は属性ごとの属性変更損失 $\mathcal{L}_{\text{change}}$ の比率を表すパラメータである．

属性変更損失 $\mathcal{L}_{\text{change}}$ のパラメータ μ_i, ν_i を全ての属性で同じ値に設定して訓練した場合，一部の属性は加えられる変更が小さく，編集内容が結果に反映されない．そのため，属性 i の付与が反映されない場合には μ_i を，排除が反映されない場合には ν_i を大きく設定する．μ_i, ν_i が大きい属性 i を編集できなかった場合，そうでない属性と比較して属性変更損失 $\mathcal{L}_{\text{change}}$ が大きくなるため，編集器は属性 i を優先して編集できるように学習する．

3.4.2 属性維持損失

編集器が属性ベクトルで指示されていない属性を変更することを防ぐため，属性維持損失 $\mathcal{L}_{\text{unchange}}$ を導入する．編集前の画像における属性 i の識別結果を p_i とする．この時，属性維持損失 $\mathcal{L}_{\text{unchange}}$ は識別結果の絶対差分を全ての属性 $i \notin T$ について平均した，

$$\mathcal{L}_{\text{unchange}} = \frac{1}{M - m} \sum_{i \notin T} |p'_i - p_i| \tag{7}$$

で定義する．M は識別可能な属性の総数を表す．

3.4.3 骨格損失

編集の前後で顔の骨格や姿勢が変化することを防ぐため，骨格損失 $\mathcal{L}_{\text{bone}}$ を導入する．骨格損失 $\mathcal{L}_{\text{bone}}$ は特徴点検出器 h を用いて計算する．特徴点検出器 h は顔画像を入力とし，目頭や鼻先等の顔にある K 個の特徴点について，その位置を表すヒートマップ $\boldsymbol{H}_k$ を出力する．すなわち，

$$(\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_K) = h(I) \tag{8}$$

である．ヒートマップ $\boldsymbol{H}_k$ は二次元行列であり，k 番目の特徴点の存在確率が高い場所ほど大きな値となる．

編集前の画像に対する k 番目の特徴点のヒートマップを $\boldsymbol{H}_{\text{k}}$，編集後の画像に対する k 番目の特徴点のヒートマップを $\boldsymbol{H}'_{\text{k}}$ とする．この時，骨格損失 $\mathcal{L}_{\text{bone}}$ はこれらのヒートマップ $\boldsymbol{H}_{\text{k}}, \boldsymbol{H}'_{\text{k}}$ の平均二乗誤差を全ての特徴点について平均した，

$$\mathcal{L}_{\text{bone}} = \frac{1}{K} \sum_{k=0}^{K} MSE(\boldsymbol{H}'_k, \boldsymbol{H}_k) \tag{9}$$

で定義する．MSE は平均二乗誤差を計算する関数であり，二つの二次元行列 $\boldsymbol{x}, \boldsymbol{y}$ の $MSE(\boldsymbol{x} = (x_{ij}), \boldsymbol{y} = (y_{ij}))$ は，

$$MSE(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{MN} \sum_{m=0}^{M} \sum_{n=0}^{N} (x_{mn} - y_{mn})^2 \tag{10}$$

である．

4. 評価実験

提案した手法を用いて実際に顔画像編集を行い，得られた結果を定性的，定量的に評価した．定性的な評価は，複数の画像に対して様々な属性を編集し，編集前の画像と比較をすることで行った．定量的な評価は，編集を通して必要な特徴が保持されているどうかを確認するため，顔の類似度を指標として行った．また，属性維持損失と骨格損失を用いず訓練した場合と，用いて訓練した場合を比較することで，手法の妥当性を評価した．本実験では生成画像と実画像を編集対象とした．

4.1 実験条件

本実験で用いた生成器，属性識別器，特徴点検出器，編集器の詳細とそれぞれの学習条件は次の通りである．

4.1.1 モデルの詳細

生成器には, Karras らによって提案された StyleGAN2[5] を用いた．StyleGAN2 では，多次元標準正規分布から無作為にサンプリングされたベクトルを，Mapping Network に入力することで得られたベクトルを潜在表現としている．そのため，Mapping Network を通して得られた潜在表現に変化を加えることで編集を行った．本実験では，70,000 枚の人間の顔画像で構成されたデータセット, Flickr-Faces-HQ[4] により訓練したモデルを使用した．ここで，実画像の潜在表現は StyleGAN2 の論文中に示された手法を用いて推定した．

属性識別器には，He らによって提案された ResNet[10] を用いた．本実験では，顔画像に含まれる 40 種類の属性を識別できるように ResNet50 を訓練した．属性識別器の訓練の詳細は 4.1.2 節で説明する．

特徴点検出器には，Bulat らによって提案された Face Alignment Network (FAN)[11] を用いた．FAN は顔画像を入力として受け取り，68 ヶ所の特徴点の位置をヒートマップとして出力する．骨格損失の計算には，推定された 68 ヶ所の特徴点の内，左右の目頭，鼻先，前歯，顎の 5 ヶ所に対応するヒートマップを用いた．本実験では，顔画像に 68 ヶ所の特徴点の位置がラベル付けされたデータセット，300W-LP[12] により訓練したモデルを使用した．

4.1.2 属性識別器の学習条件

属性識別器の訓練は顔画像のデータセット，CelebA[13] を用いて次の条件で行った．CelebA は 202,599 枚の顔画像からなるデータセットであり，各顔画像には 40 個の属性ラベルが付与されている．属性ラベルはそれぞれ顔の属性に対応しており，画像が対応する属性を含む場合には 1 が，含まない場合には 0 が割り当てられる．

学習は CelebA の 162,770 枚の訓練用データに眼鏡に関するデータ拡張を加えた 315,019 枚の画像を用い，バッチサイズは 256 で 30 エポック行った．損失関数には，正解ラベルと出力結果の交差エントロピー誤差を用い，学習の際にはランダムな水平反転を行った．最適化は確率的勾配降下法によって行い，学習率は 0.1 に設定した．編集器の訓練には，各エポックで得られた重みの中で検証用データに対して最も良い性能を出したものを使用した．

眼鏡に関するデータ拡張は，訓練用データの中で眼鏡をかけていない画像に対して AttGAN[8] で Eyeglasses を付与する編集を行い，編集後の画像も訓練データに含めることで行った．編集に用いた AttGAN は，同じく CelebA で訓練したモデルである．

表 2 属性変更損失のパラメータ μ_i, ν_i.

属性	μ_i	ν_i	属性	μ_i	ν_i
Bags Under Eyes	1.5	1.5	Male	1.5	1.5
Big Lips	1.5	1.5	Narrow Eyes	1.5	1.5
Big Nose	1.5	1.5	Oval Face	2	2
Eyeglasses	25	5	Pointy Nose	1.5	1.5
High Cheekbones	1.5	1.5	Young	2	2

4.1.3 編集器の詳細と学習条件

編集器は，Shortcut Connection を持つ 8 層の全結合層で構成されたネットワークであり，活性化関数には Leaky ReLU が用いられている．Leaky ReLU のハイパーパラメータを 0.2 とした．ネットワークの入力層と隠れ層は 552 次元であり，潜在表現と属性ベクトルを連結したベクトルを入力として受け取る．また出力層は 512 次元であり，潜在表現に加えるべき差分を出力する．

本実験では，属性識別器が識別できる 40 種類の属性から，Wearing Necklace と Wearing Necktie を除いた，38 種類の属性を編集対象として編集器の訓練を行った．これらの属性を除いた理由は，StyleGAN2 の生成画像には首回りの領域が含まれないためである．

編集器の訓練はバッチサイズを 32 として 80,000 回行った．重みの最適化には Adam Optimizer を用い，学習率は 0.0001 に, その他のパラメータは $\beta_1 = 0.9, \beta_2 = 0.999$ に設定した．一度に編集対象とする属性の上限数 m_u は 3 とした．各損失のハイパーパラメータは $\lambda_c = 1, \lambda_u = 20, \lambda_b = 1000$ とした．表 2 に，属性変更損失 $\mathcal{L}_{\text{change}}$ のパラメータ μ_i, ν_i を 1 以外に設定した属性を示す．

4.2 評価指標

本実験では人物同一度 E_{ID} を導入し，編集を通して顔の特徴が維持されていることを評価した．人物同一度 E_{ID} は顔認証器 g により計算され，編集前と編集後における顔の類似度を示す．人物同一度 E_{ID} は，編集前の画像 I と編集後の画像 I' を顔認証器 g へ入力することで得られた特徴ベクトル $\boldsymbol{c}, \boldsymbol{c}'$ のコサイン類似度,

$$E_{\text{ID}} = \cos(\boldsymbol{c}', \boldsymbol{c}) = \cos(g(I'), g(I)) \tag{11}$$

で定義する．cos は 2 つのベクトルのコサイン類似度

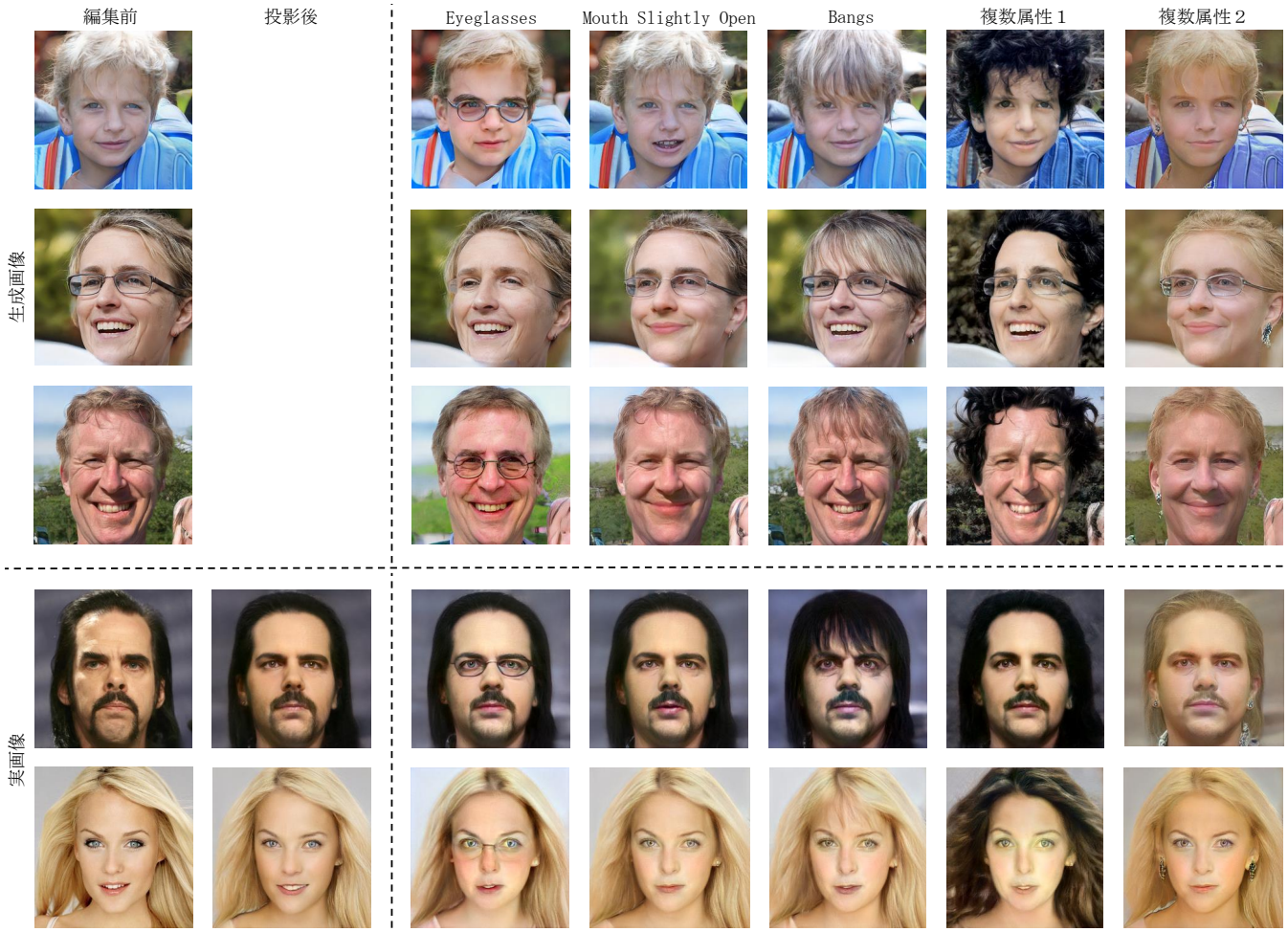

図 3 提案手法を用いて生成画像及び実画像を編集した結果．左端の各画像に対し，上に記載した属性を編集した．2 列目の投影後とは，実画像を潜在空間に投影して得た，潜在表現から生成した画像である．複数属性 1 は Black Hair と Wavy Hair を付与する，複数属性 2 は Blond Hair と Wearing Earrings を付与して Mouth Slightly Open を排除する編集をした結果である．多くの編集結果において，対象の属性のみが変化している．

を計算する関数であり，N 次元ベクトルについての $\cos(\boldsymbol{x} = (x_{ij}), \boldsymbol{y} = (y_{ij}))$ は，

$$\cos(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{n=0}^{N} x_i y_i}{\sqrt{\sum_{n=0}^{N} x_i^2} \sqrt{\sum_{n=0}^{N} y_i^2}} \tag{12}$$

である．人物同一度 E_{ID} が高いほど，編集前後の顔が類似していることを示す．本実験では，顔認証器 g に大規模な顔画像のデータセット，VGGFace2[14] により訓練した FaceNet[15] を用いた．

編集前後における編集対象とした属性の識別結果の差分 $p' - p$ によっても評価を行う．差分 $p' - p$ は編集器が編集すべき属性に加えた変更の大きさを表す．p は編集前の，p' は編集後の画像に対する識別結果であり，編集器の訓練に用いた属性識別器により計算する．

4.3 編集結果の評価

生成画像と実画像の二種類を編集対象として，編集性能の評価を行う．編集対象とした実画像は，顔画像データセット，CelebA-HQ[3] から取り出したものである．表 3 の上段に，生成画像と実画像を編集対象として，提案手法を用いて編集を行った場合の評価指標を示す．また表 3 の下段に，生成画像を編集対象として，属性維持損失または骨格損失の片方を除いて訓練した場合の評価指標の算出結果を示す．ここで，表 3 の値は，1,000 枚の画像に対し，全ての属性をそれぞれ単体で付与する編集をした場合について，各指標を計算して平均したものである．この 1,000 枚

表 3 各指標の計算結果．両指標とも，必ずしも値が高いほど良い結果であることを示す訳ではない．提案手法で生成画像または実画像を編集した場合（提案手法），属性維持損失か骨格損失の片方を用いずに訓練した編集器で生成画像を編集した場合（属性維持損失なし，骨格損失なし）に対して各指標を計算した．

	E_{ID}	$p' - p$
提案手法（生成画像）	**0.789**	0.493
提案手法（実画像）	0.510	0.414
属性維持損失なし	0.140	**0.967**
骨格損失なし	0.739	0.551

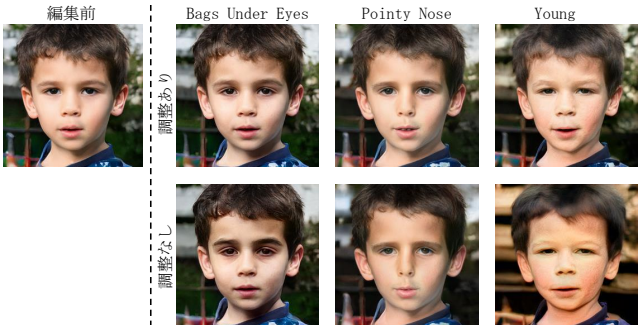

図 4 編集によって加えられる変化が弱い属性の例．また，対応する属性変更損失のパラメータ調整を行わなかった場合（調整なし）と，行った場合（調整あり）の比較を示す．

の画像は，編集で変更を加える属性 i の識別結果 p_i が 0.1 以下となるようにした．図 3 に，提案手法を用いて編集を行った結果を示す．

表 3 の人物一致度と図 3 から，提案手法が元の顔の特徴を維持しながら，対称の属性のみに変更を加えていることが分かる．特に，複数の属性を同時に指定する，複雑な編集も行えていることが分かる．

4.3.1 画像の種類と編集性能の関係

実画像は生成画像と比較して，編集結果が元画像から変化しない場合や，誤った変更が加えられる場合が多かった．表 3 から，実画像を編集した場合の方が人物同一度と識別結果の差分が小さく，定量的に良い結果が得られていないことが分かる．また，図 3 の 5 行目で Eyeglasses の編集に失敗しているように，定性的にも実画像に対する編集性能は低いことが分かる．

実画像の方が編集に失敗しやすい原因として，実画像を投影して得られた潜在表現の分布と，生成画像の潜在表現の分布が異なることが考えられる．実画像の潜在表現はノルムが 70 前後で，要素の最大値が 5 前後，最小値が −5 前後であった．一方で，生成画像の潜在表現はノルムが 15 前後で，要素の最大値が 3 前後，最小値が −1 前後であった．細かい値は重みによって異なるが，実画像は生成画像と比較して，潜在表現のノルムが 3〜5 倍大きな値となった．編集器の訓練には生成画像の潜在表現だけを用いるため，分布の異なる実画像の潜在表現に対応することが難しく，これが編集性能に差が生じる原因であると考えられる．

4.3.2 属性と編集性能の関係

属性によっても編集結果の品質には差があり，編集によって加えられる変更が小さいため，良い結果が得られない属性があった．これらの属性は，対応する属性変更損失のパラメータを調節することで，結果を改善することができる．図 4 に，変更が小さい属性について，パラメータを調節行った場合と行っていない場合の編集結果の比較を示す．図 4 から，パラメータを調節することでよりはっきりした編集が可能になることが分かる．

また，編集によって誤った変更が加えられる属性もあった．この原因として，属性識別器が属性が持つ本来の特徴ではない，誤った特徴を元に識別を行っていることが考えられる．編集器は，属性識別器によって計算する損失から学習するため，編集結果には属性識別器が捉えた特徴が反映される．例えば図 5 において，Chubby と Double Chin は編集結果に差が見られず，Double Chin の結果には二重顎がはっきりと現れていない．これは，Chubby と Double Chin の属性ラベルは同時に存在することが多く，属性識別器がこれらの属性が持つ特徴を同一のものとして認識したためだと考えられる．

図 5 編集によって誤った変化が加えられる属性の例．Chubby と Double Chin の編集結果が同じになっていることが分かる．

図 6 編集によって誤った変化が加えられる，Eyeglasses の編集結果．対応する属性変更損失のパラメータ調整と眼鏡に関するデータ拡張を行わなかった場合（調整なし，DA なし）とパラメータ調整のみを行った場合（調整あり，DA なし），両方を行った場合（提案手法）の比較を示す．

誤った変更が加えられる別の原因として，属性によって識別難易度に差があることが考えられる．例えば，眼鏡や帽子といった離散的で変化する領域の広い属性は，有無を識別が容易である．そのため，属性識別器はこれらの属性の特徴を細部まで捉えておらず，編集器が適切な変化を求めることが困難になると考えられる．実際に，図 6 において属性変更損失のパラメータの調整や，眼鏡に関するデータ拡張をすることなく訓練した場合，Eyeglasses の編集結果は目の周辺が変色しているだけある．図 6 に示すように，パラメータ調整に加え，属性識別器の訓練データにデータ拡張を行うことで Eyeglasses の編集結果を改善することができる．改善の理由は，データ拡張によって属性識別器が眼鏡の特徴を正確に捉えるようになったためであると考えられる．

4.3.3 各損失と編集性能の関係

図 7 に，属性維持損失または骨格損失の片方を除いて訓練した場合と，提案手法との編集結果の比較を示す．表 3 の人物同一度から，提案手法が最も編集前の顔の特徴を維持しながら編集を行えていることが分かる．属性識別器の識別結果は，提案した手法が最も悪い結果となっているが，これは編集器が元の顔の特徴が崩れるほど強い変化を加え

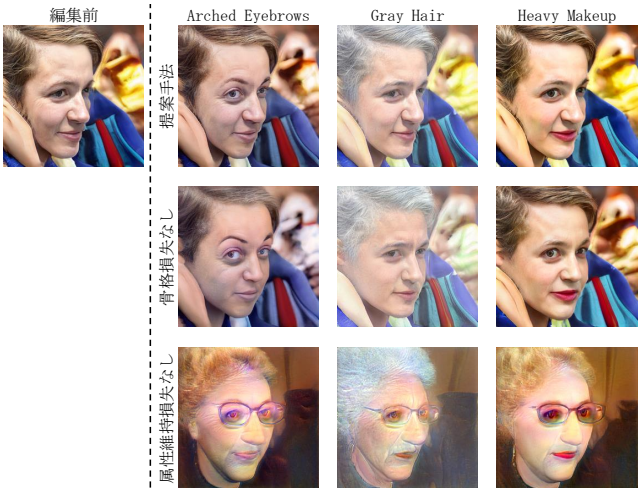

図 7 属性維持損失を用いずに訓練した場合（属性維持損失なし）と骨格損失を用いなかった場合（骨格損失なし），両方を用いた場合（提案手法）の比較を示す．

ているためである．図 7 に示す結果から，定性的には提案した手法が最も良く編集を行えていることが分かる．属性維持損失を取り除いた場合，編集結果の顔画像は編集前の特徴をほとんど維持できていない．骨格損失を除いた場合も，目や口といった顔のパーツの位置関係や，顔の姿勢が編集を通して変化している．

5. 結論

本研究では顔画像編集において，編集対象の属性のみ変化させ，関係のない特徴は編集前の状態を維持する正確な編集を実現する手法を提案した．正確な編集のため，本手法では属性維持損失と骨格損失を新たに導入し，編集器の訓練を行った．訓練した編集器を用い，複数枚の生成画像と実画像を編集対象として評価実験を行った．この実験により，属性維持損失を用いることで，編集器が編集対象外の属性を変化させることを防げることが示された．また，骨格損失を用いることで，編集器が顔の骨格や姿勢を変化させることを防げることが示された．

解決すべき今後の課題として，編集により誤った変化が加えられる属性を無くすことが挙げられる．正確な編集を行うためには，属性の特徴を正しく捉えるよう，属性識別器を改善する必要があると考えられる．また，別の課題として，実画像に対する編集性能を向上させることが挙げられる．実画像を編集することは生成画像を編集する以上に実用性が高い．しかし，現状の編集機では，実画像を編集すると誤った変更を加えてしまう場合や，変更を加えることができない場合があった．そのため，生成画像の潜在表現と実画像の潜在表現で編集結果の質に差が生まれる原因を解明し，解決する必要がある．

謝辞 本研究の一部は JSPS 科研費 17K20143, 20H05951, JST さきがけ JPMJPR1858 の助成を受けたものです．

参考文献

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y.: Generative Adversarial Nets, *Advances in Neural Information Processing Systems* (Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. and Weinberger, K. Q., eds.), Vol. 27, Curran Associates, Inc., pp. 2672–2680 (2014).

[2] Radford, A., Metz, L. and Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks (2015).

[3] Karras, T., Aila, T., Laine, S. and Lehtinen, J.: Progressive Growing of GANs for Improved Quality, Stability, and Variation, *CoRR*, Vol. abs/1710.10196 (online), available from ⟨http://arxiv.org/abs/1710.10196⟩ (2017).

[4] Karras, T., Laine, S. and Aila, T.: A Style-Based Generator Architecture for Generative Adversarial Networks, *CoRR*, Vol. abs/1812.04948 (online), available from ⟨http://arxiv.org/abs/1812.04948⟩ (2018).

[5] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J. and Aila, T.: Analyzing and Improving the Image Quality of StyleGAN, *CoRR*, Vol. abs/1912.04958 (online), available from ⟨http://arxiv.org/abs/1912.04958⟩ (2019).

[6] Mirza, M. and Osindero, S.: Conditional Generative Adversarial Nets, *CoRR*, Vol. abs/1411.1784 (online), available from ⟨http://arxiv.org/abs/1411.1784⟩ (2014).

[7] Isola, P., Zhu, J., Zhou, T. and Efros, A. A.: Image-to-Image Translation with Conditional Adversarial Networks, *CoRR*, Vol. abs/1611.07004 (online), available from ⟨http://arxiv.org/abs/1611.07004⟩ (2016).

[8] He, Z., Zuo, W., Kan, M., Shan, S. and Chen, X.: Arbitrary Facial Attribute Editing: Only Change What You Want, *CoRR*, Vol. abs/1711.10678 (online), available from ⟨http://arxiv.org/abs/1711.10678⟩ (2017).

[9] Shen, Y., Gu, J., Tang, X. and Zhou, B.: Interpreting the Latent Space of GANs for Semantic Face Editing, *CoRR*, Vol. abs/1907.10786 (online), available from ⟨http://arxiv.org/abs/1907.10786⟩ (2019).

[10] He, K., Zhang, X., Ren, S. and Sun, J.: Deep Residual Learning for Image Recognition, *CoRR*, Vol. abs/1512.03385 (online), available from ⟨http://arxiv.org/abs/1512.03385⟩ (2015).

[11] Bulat, A. and Tzimiropoulos, G.: How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230, 000 3D facial landmarks), *CoRR*, Vol. abs/1703.07332 (online), available from ⟨http://arxiv.org/abs/1703.07332⟩ (2017).

[12] Zhu, X., Lei, Z., Liu, X., Shi, H. and Li, S. Z.: Face Alignment Across Large Poses: A 3D Solution, *CoRR*, Vol. abs/1511.07212 (online), available from ⟨http://arxiv.org/abs/1511.07212⟩ (2015).

[13] Liu, Z., Luo, P., Wang, X. and Tang, X.: Deep Learning Face Attributes in the Wild, *CoRR*, Vol. abs/1411.7766 (online), available from ⟨http://arxiv.org/abs/1411.7766⟩ (2014).

[14] Cao, Q., Shen, L., Xie, W., Parkhi, O. M. and Zisserman, A.: VGGFace2: A dataset for recognising faces across pose and age, *CoRR*, Vol. abs/1710.08092 (online), available from ⟨http://arxiv.org/abs/1710.08092⟩ (2017).

[15] Schroff, F., Kalenichenko, D. and Philbin, J.: FaceNet: A Unified Embedding for Face Recognition and Clustering, *CoRR*, Vol. abs/1503.03832 (online), available from ⟨http://arxiv.org/abs/1503.03832⟩ (2015).