Deep learningによる複数人物画像のヒュージョン

Fusion of Multiple Person Images using Deep learning

高木 一宏 † 岡留 剛 †
Kazuhiro Takagi Takeshi Okadome

概 要

画像生成で使われる Generative Adversarial Network(GAN) は，他手法と比較して鮮明な画像を生成する．その欠点として，入力で用いる潜在変数ベクトルの各要素が，生成した画像のどの部分に対応しているのか，人手で調べる必要がある。また，画像処理の分野で，異なる画像同士を合成させる手法は多数存在するが，出力画像がいびつな結果となり成果が出ていない．本研究では，異なる二つの入力画像を写像した 2 つの潜在変数ベクトルの各要素を自動的に明示化し，それらの各要素を組み合わせることで，二つの入力画像を指定通りに融合する手法の構築を目指す．提案手法から得られた生成画像の結果は，入力画像の特徴を含んだ画像が生成された．ただし，提案手法は，GAN と比較して細かな指定が可能になったが，画像内の人や物の形状に関する部分については，まだ細かな指定ができないため今後の課題である．

Generative Adversarial Network(GAN) is often used in the field of image generation. This method is able to generate more photo-realistic images than other methods. As GAN's disadvantage, it is necessary to manually check that each element of latent variable vector used in GAN's input correspond to which part of the generated image. Also, in the feald of image processing, there are many ways to combine different images. However, the output image becomes an artifact and no results have been obtained. In this study, each element of two latent variable vectors, which mapped two different input images, is automatically clarified, and we aim to build a method to fuse as specified by combining those elements. The generated image which was obtained from the proposed method contained the feature of the input image. However, the proposed method can be specified more detail than GAN, but it is a future task because it is not possible to specify in detail the parts related to the shapes of people and objects in the image.

1. はじめに

画像生成において，機械が自動的に新しい画像を生成することで，アニメーション作成や動画作成などへの分野の応用に期待されている．

画像生成の分野では，Generative Adversarial Network (GAN) [1] をベースとした手法が大きく成果を出している．GAN は，偽の画像を生成する Generator，真の画像と生成された偽の画像を識別する Discriminator，二つのニューラルネットで構成される．この手法の学習法が，互いのニューラルネットを競わせることから敵対的学習と呼ばれている．利点として，画像が分布に従うと直接仮定する必要がないため，画像の表現力を上げ，他手法よりも鮮明な画像を生成できる．しかし，この手法の欠点は，生成させる画像を選択できないところである．なぜなら，Generator の入力である潜在変数ベクトルの各要素が，生成される画像のどの部分に対応しているのかが実験的にしかわからないためである．そのため，手動に頼る必要があり，人手による膨大なコストがかかってしまう．

本研究では，自動で潜在変数ベクトルの各要素を明示化することで，生成したい画像を指定可能にすることを目指す．また，潜在変数ベクトルの各要素の明示化は，各入力画像に対応する各潜在変数ベクトルの要素を組み替えることで，2 種類の画像の融合を可能になると考えられる．明示化した潜在変数ベクトルの各要素を組み合わせることで，異なる画像同士を違和感なく融合する手法を提案する．本論文の貢献は，潜在変数ベクトルの各要素を組み合わせることで，異なる画像の指定した部分を融合可能にしたことである．以下各節で，関連研究，提案手法，予備実験について詳細に述べ，まとめを記す．

2. 関連研究

Style Separation and Synthesis via Generative Adversarial Network (S3-GAN) [2] は，潜在変数ベクトルの要素を Content 要素と Style 要素の 2 種類に分割して disentangle にする手法である．Content 要素とは，画像

† 関西学院大学大学院，Kwansei Gakuin University Graduate

内の形や輪郭などを表し，Style 要素は，画像全体の色あいである．S3-GAN は，二つの異なる入力画像の潜在変数ベクトルの Content 要素と Style 要素を，入れ替える．片方の画像の Content 要素を持ち，もう片方の画像の Style 要素を持った，新たな画像が生成できる．しかし，この手法は，潜在変数ベクトルを二つの要素でしか明示化していないため，画像生成のために，より細かな指定をするのは困難である．

Suzuki ら [3] は，学習済みの GAN の Generator を使用することで，ある画像の指定した部分を，別の画像の指定した部分に貼り付けることで，2 種類の異なる画像の部位を融合した新たな画像を生成させることを可能にした．融合の行い方は，ある画像を生成する学習済みの Generator の特徴 map の指定した部分に該当するところを 0 にし，その部分に別の画像の特徴 map の値を加算する．しかし，この手法は，特徴 map の値の部分は手動で指定する必要がある．

本研究では，潜在変数ベクトルの要素を詳細に明示化し，画像同士の融合も自動化する手法を提案する．

3. 提案手法

3.1 提案モデル

ここでは，異なる入力画像同士から融合した画像を生成させるモデルについて説明する．図 1 は，提案モデルを表現するグラフである．入力画像を A，B とおき，生成される画像は C とする．入力画像を潜在変数ベクトルに写像するニューラルネットは Encoder，潜在変数ベクトルから画像を生成するニューラルネットを Generator と呼ぶ．Generator，Encoder の構造については，後に 4.2 節で詳細に述べる．入力画像 A から Encoder により写像された潜在変数ベクトル $\mathbf{Z}^A = \{\mathbf{z}_1^A, \ldots, \mathbf{z}_N^A\}$，入力画像 B から Encoder により写像された潜在変数ベクトル $\mathbf{Z}^B = \{\mathbf{z}_1^B, \ldots, \mathbf{z}_N^B\}$ とする．$\mathbf{\Lambda} = \{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N\}$ は，潜在変数ベクトル $\mathbf{Z}^A$ と $\mathbf{Z}^B$ の各要素の足し合わせた方を調整する超パラメータとする．超パラメータ $\mathbf{\Lambda}$ の値は，ユーザ指定である．この超パラメータ $\mathbf{\Lambda}$ の値により，画像の融合度合いが調整される．入力画像 A，B の各潜在変数ベクトルを，超パラメータ $\mathbf{\Lambda}$ により，調整して足し合わせた $\mathbf{Z}^C$ を以下の式とする．

$$\mathbf{Z}^C = \mathbf{\Lambda} \circ \mathbf{Z}^A + (\mathbf{1} - \mathbf{\Lambda}) \circ \mathbf{Z}^B \tag{1}$$

生成画像 C は，$\mathbf{Z}^C$ を Generator に入力することで得られる．生成画像 C を生成する潜在変数ベクトル $\mathbf{Z}^C = \{\mathbf{z}_1^C, \ldots, \mathbf{z}_N^C\}$ とする．

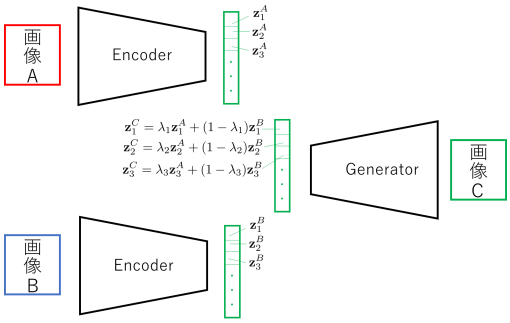

図 1: 入力画像 A，入力画像 B の，各潜在変数ベクトル $\mathbf{Z}^A, \mathbf{Z}^B$ の要素の割合を，超パラメータ $\mathbf{\Lambda}$ を調整して足し合わせた式 (1) の潜在変数ベクトルから，画像 C を生成する提案モデル

3.2 Loss 関数

提案モデルを学習するためには，図 2 に示されている Discriminator と Perceptual Network という二つのニューラルネットを導入する．Discriminator は，Generator から生成された画像が偽，真の画像が真であると，識別するためのニューラルネットである．Perceptual Network は，画像を入力して，画像を表すラベルを出力する分類のためのニューラルネットであるが，本研究では，このニューラルネットの各層の特徴 map を使用する．Discriminator と Perceptual Network の構造については，後に 4.2 節で詳細を述べる．また，Encoder，Generator，Discriminator，Perceptual Network を，最適化するために，目的関数には，5 つの誤差関数を用いる．各誤差関数は，Adversarial loss，Content Perceptual loss，Style Perceptual loss，Reconstruction loss，Total Variation Loss である．Adversarial loss は，より写実的で鮮明な画像を生成するために用いる．Content Perceptual loss は，入力画像の形状を，生成画像に反映させるために用いる．Style Perceptual loss は，入力画像の色合いなどを，生成画像に反映させるために用いる．Reconstruction loss は，生成画像が乱れないように，Generator に入力画像と同じ画像を，正確に再構成させるために用いる．Total Variational loss は，画像同士が不連続になりすぎて，突飛な歪さが生じるのを防ぐために用いる．各誤差関数の詳細については，以下に述べる．また，超パラメータ λ の値が，全て 1 なら，画像 C は画像 A に再構成される．全て 0 なら，画像 C は画像 B に再構成される．以下の節から，簡略化のため，Encoder，Generator，Discriminator，Perceptual Network は，E，G，D，P と呼ぶ．

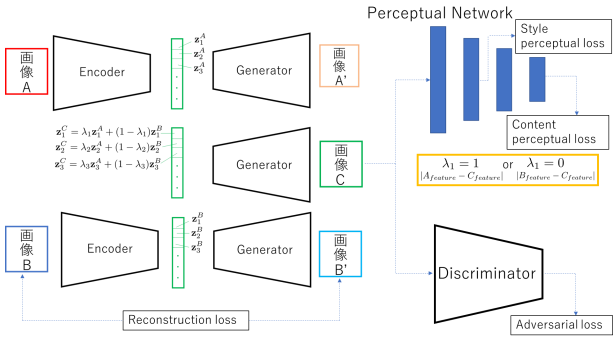

図 2: 提案モデルの学習のためには，Encoder，Generator のは他に，Discriminator，Perceptual Network を用いる．目的関数には，5 つの誤差関数 Adversarial loss，Content Perceptual loss，Style Perceptual loss，Reconstruction loss，Total Variation Loss を用いる．また，Content Perceptual loss，Style Perceptual loss は，超パラメータ λ の値により，画像 C を入力したときの Perceptual Network の特徴 map が，画像 A，画像 B の各 Perceptual Network の特徴 map のどちらと比較するか決まる．

3.2.1 Adversarial loss

E，G，D の 3 つのパラメータを最適化するために，Adversarial loss L_A を用いる．通常，GAN [1] で用いられる Adversarial loss は，Jensenn-Shannon divergence を用いる．しかし，それを用いると，勾配消失を引き起こし，Generator のパラメータの勾配が 0 になり，Generator の重みパラメータの学習が上手く進まず，学習が失敗することがある．そのため，Jensenn-Shannon divergence には学習の安定性がない．そこで，勾配消失を防ぎ，学習を安定化させるために，Wasserstein divergence [4] を用いる．

多様体 $\mathcal{M}$ 上の訓練データの分布を $p_{\mathcal{M}}$ とする．$p_{\mathcal{M}}$ からデータをサンプリングする過程を $I \sim p_{\mathcal{M}}$ とする．Adversarial loss は以下の式になる．

$$L_A(E,G,D) = \mathbb{E}_{I\sim p_{\mathcal{M}}}[D(I)] - \mathbb{E}_{A,B\sim p_{\mathcal{M}}}[D(G(\mathbf{Z}^C))], \quad (2)$$

$\mathbf{Z}^C$ は式 (1) から得られる．最大最小目的関数は，L_A を最適化するために用いられる．

$$arg\min_{E,G}\max_{D} L_A(E,G,D). \quad (3)$$

ここで，E と G は，生成画像 $G(\mathbf{Z}^C)$ が訓練データの画像と類似した画像を生成するために，L_A を最小化する．D は，生成画像 $G(\mathbf{Z}^C)$ は偽で，訓練データの画像は真であると，きちんと判別できるようにするために L_A を最大化する．L_A により，鮮明な生成画像 $G(\mathbf{Z}^C)$ が得られるようになる．

3.2.2 Content Perceptual loss

入力画像 A と入力画像 B を要素ごとに融合した画像 C が，画像 A や画像 B の形状を残したい場合，Content Perceptual LossL_c を用いる．通常，融合生成した画像 C の教師データは存在しない．そのため，P を使用し，融合生成した画像 C の特徴が，入力画像の形状の特徴に近くなるように誤差を最小化する [5, 6]．超パラメータ Λ の要素 λ_n が 1 の時は，入力画像 A の特徴の形状に近付ける．0 の時は，入力画像 B の特徴の形状に近付ける．ある入力画像 $I \sim p_{\mathcal{M}}$ を P に入力したときの，P の l 層の特徴 map は，$P_l(I)$ とする．L_c は，特徴間の平均 2 乗誤差で表される．

$$L_c(E,G) = \begin{cases} \sum_{l\in\mathcal{F}_c} \|P_l(C) - P_l(A)\|_2^2, & \lambda_n = 1. \\ \sum_{l\in\mathcal{F}_c} \|P_l(C) - P_l(B)\|_2^2, & \lambda_n = 0. \end{cases} \quad (4)$$

ここで，$\mathcal{F}_c$ とは，L_c で使われる P の特徴 map の集合の事である．

L_c で使われる P の特徴 map は，P の深い層の特徴 map である．P の深い層には，画像の形状に関する特徴 map が保存されているためである [6]．

3.2.3 Style Perceptual loss

入力画像 A と入力画像 B を要素ごとに融合した画像 C が，画像 A や画像 B の色合いを残したい場合，Style Perceptual LossL_s を用いる． L_c と同様の理由から，P の各層から出力した，融合生成画像 C の特徴が，入力画像の色合いの特徴に近くなるように，入力画像の特徴との誤差を最小化する．もし，画像 I を入力したときの，P の l 層の特徴 map$P_l(I)$ が，$C_l \times H_l \times W_l$ の型を持つなら，L_c は，グラム行列のフロベニウス距離の 2 乗誤差で以下のように表される．

$$L_s(E,G) = \begin{cases} \sum_{l\in\mathcal{F}_s} \|\psi_l(C) - \psi_l(A)\|_F^2, & \lambda_n = 1. \\ \sum_{l\in\mathcal{F}_s} \|\psi_l(C) - \psi_l(B)\|_F^2, & \lambda_n = 0. \end{cases} \quad (5)$$

ここで，$\mathcal{F}_s$ とは，L_s で使われる，P の特徴 map の集合の事である．$\psi_l(I)$ は $C_l \times C_l$ のグラム行列であり，C_l

は，特徴 map$P_l(I)$ のチャネル次元である．グラム行列 $\psi_l(I)$ の要素 $\psi_l^{c,c'}(I)$ は，以下の式で表される．

$$\psi_l^{c,c'}(I) = \frac{1}{C_l H_l W_l} \sum_{h,w} p_l^{h,w,c}(I) p_l^{h,w,c'}(I). \quad (6)$$

グラム行列は，式 (6) からわかるように，特徴 map の縦，横の空間情報を加算して消去し，チャネル同士で相関を取る行列にしている [5]．

L_s で使われる P の特徴 map は，P の浅い層の特徴 map である．P の浅い層には，画像の色や質感に関する特徴 map が保存されているためである [6]．

3.2.4 Reconstruction loss

Reconstruction lossL_R は，E と G が正確に，入力画像と同じ画像を，再構成させるために必要である．また，入力画像 A と入力画像 B の融合する要素を調整するために，式 (1) の超パラメータ Λ の要素が，全て 1 のとき入力画像 A を，全て 0 のとき入力画像 B を再構成させる必要がある．L_R は以下の式で表される．

$$\begin{aligned} L_R(E,G) &= \|A' - A\|_1 + \|B' - B\|_1 + R(I) \\ &= \|G(E(A)) - A\|_1 + \|G(E(B)) - B\|_1 + R(I), \end{aligned} \quad (7)$$

$$\begin{aligned} R(I) &= \|G(\mathbf{\Lambda} \circ E(A) + (\mathbf{1} - \mathbf{\Lambda}) \circ E(B)) - I\|_1 \\ &= \begin{cases} R(A), & \Lambda = \mathbf{1}. \\ R(B), & \Lambda = \mathbf{0}. \\ 0, & \Lambda \neq \mathbf{1} \cap \Lambda \neq \mathbf{0}. \end{cases} \end{aligned}$$

L_R には，L1 誤差関数を使用する．L2 誤差関数を使用した場合と比較して，画像がより鮮明になるためである [2]．

3.2.5 Total Variation loss

Total Variation loss L_{TV} は，生成した画像の，隣り合う画素値間の差分を少なくします．そのため，生成画像は，とても滑らかになり，いびつな不連続性を減らします．L_{TV} の式は，再構成画像と融合生成した画像のどちらにも適用する．

$$\phi(\boldsymbol{X}) = \sum_{i,j} ((x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2), \quad (8)$$

$$\begin{aligned} L_{TV}(E,G) &= \phi(C) + \phi(A') + \phi(B') \\ &= \phi(G(\mathbf{\Lambda} \circ E(A) + (\mathbf{1} - \mathbf{\Lambda}) \circ E(B))) \\ &\quad + \phi(G(E(A))) + \phi(G(E(B))). \end{aligned} \quad (9)$$

3.3 学習手順

この節では，提案モデルを学習するための手順について紹介する．提案モデルは，入力画像の潜在変数ベクトルの各要素を，超パラメータ Λ の要素を調節して足し合わせることで融合画像を生成する．そのためには，入力画像が，E により潜在変数ベクトルへ写像し，G によりきちんと画像を再構成させる必要がある．そのため，E と G に関しては，図 3 の Auto Encoder で事前学習を行う．また，第 3.2 節で定義した Loss 関数で，最初から学習するよりも，事前学習済みの方が学習を高速化できると考えられる．

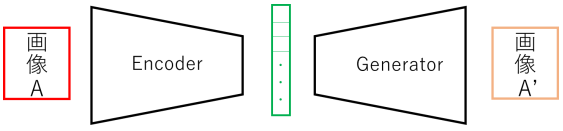

図 3: 入力画像がきちんと再構成されるように，E と G で事前学習を行う．入力が画像 A の場合である．

$$L_{AE}(E,G) = \|A' - A\|_1 = \|G(E(A)) - A\|_1. \quad (10)$$

Auto Encoder の学習には，式 (10) の絶対値誤差を使用する．式でも入力画像が A の場合である．

事前学習済みの E と G を使用して，第 3.2 節で定義した式 (2)，式 (4)，式 (5)，式 (7)，式 (9) を，重みづけ和して学習を行う．重みづけ和した目的関数を L_O とする．

$$\begin{aligned} L_O(E,G,D) &= w_1 L_A(E,G,D) + w_2 L_c(E,G) \\ &\quad + w_3 L_s(E,G) + w_4 L_R(E,G) + w_5 L_{TV}(E,G), \end{aligned} \quad (11)$$

ここで，式 (11) の，重み w_1, w_2, w_3, w_4, w_5 を調節することで，各 Loss 関数の重要度を調節する．以下の minmax 問題を解くことで、提案モデルの E と G を学習することができる．

$$E^\star, G^\star = arg \min_{E,G} \max_{D} L_O(E,G,D). \quad (12)$$

4. 予備実験

4.1 使用したデータセット

今実験で使用したのは，有名人の顔画像データセットである，CelebA dataset [7] を使用した．データセットは，10K の属性で分類された，202599 枚の有名人の顔画像データで構成されている．顔画像データの型は，(縦，

横，チャネル数) = (178，218，3) である．提案モデルの学習の時は，(128，128，3) へ型変換を行う．学習のための，訓練データを 200000 枚使用し，バッチサイズを 100 枚とし，訓練データからランダムに取り出す．テストデータを残りの 2599 枚とした．

4.2 使用したニューラルネットの構造

表 1: E，G，D の構造，BN: Batch Normalization

Encoder E

Layer name	Kernel size	stride	Output size
Input image			3×128×128
Conv, BN, Leaky ReLU	64×4×4	2	64×64×64
Conv, BN, Leaky ReLU	128×4×4	2	128×32×32
Conv, BN, Leaky ReLU	256×4×4	2	256×16×16
Conv, BN, Leaky ReLU	512×4×4	2	512×8×8
Conv, BN, Leaky ReLU	1024×4×4	2	1024×4×4

Generator G

Layer name	Kernel size	stride	Output size
Input image			1024×4×4
DeConv, BN, ReLU	512×4×4	2	512×8×8
DeConv, BN, ReLU	256×4×4	2	256×16×16
DeConv, BN, ReLU	256×4×4	2	128×32×32
DeConv, BN, ReLU	128×4×4	2	64×64×64
DeConv, BN, ReLU	64×4×4	2	3×128×128

Discriminator D

Layer name	Kernel size	stride	Output size
Input image			3×128×128
Conv, Leaky ReLU	64×4×4	2	64×64×64
Conv, BN, Leaky ReLU	128×4×4	2	128×32×32
Conv, BN, Leaky ReLU	256×4×4	2	256×16×16
Conv, BN, Leaky ReLU	512×4×4	2	512×8×8
Conv, BN, Leaky ReLU	1024×4×4	2	1024×4×4
Fully Connected	1×(1024×4×4)	2	1

E，G，D の構造の詳細は，表 1 で示されている．また，構造の作成は [2] を参考にした．E，D は，Convolution，Batch Normalization，Leaky ReLU を一つのブロックとして，5 ブロック使用する．しかし，D の出力は完全結合を使用する．G は，Deconvolution，Batch Normalization，ReLU を一つのブロックとして，4 ブロック使用する．G の出力ブロックは，Deconv，tanh とする．Batch Normalization を含まない層があるのは，サンプリングの振動とモデルの不安定性が生じるのを防ぐためである．G の出力に tanh を用いたのは，E と G の入力の前に入力画像の画素を [-1，1] に正規化し，G が生成する画像も [-1，1] に正規化された画像にするためである．表示の際には，[0，255] に戻す．[-1，1] に正規化して学習を行う方法は，学習時間を大幅に減らすことができるので使用した．

P には，ImageNet dataset [8] で事前学習された VGG-19network [9] を使用した．他にも，事前学習されたニューラルネットは存在するが，構造が複雑になり正確な特徴 map を抽出するのが困難になる．そのため，高い精度を持ち，構造が浅い層から深い層へ直線につながった簡単な構造を持つ，VGG-19 を用いた．

4.3 実装の詳細について

1 節で定義した，各潜在変数ベクトル $\mathbf{Z}^A$，$\mathbf{Z}^B$ の型は，[1024，4，4] となる．$\{\mathbf{z}_1^A, \ldots, \mathbf{z}_N^A\}$，$\{\mathbf{z}_1^B, \ldots, \mathbf{z}_N^B\}$ の要素数は，$N = 5$ とする．$\{\mathbf{z}_1^A, \ldots, \mathbf{z}_4^A\}$ の各要素の型は [204，4，4] とし，$\mathbf{z}_5^A$ の要素の型は [208，4，4] とする．$\mathbf{Z}^B$，$\mathbf{\Lambda}$ の各要素の型も同様とした．

3.2.2 節の content perceptual loss，3.2.3 節の style perceptual loss で使う特徴 map は，conv1_1，conv2_1，conv3_1，conv4_1，conv5_1 からの出力を用いた．3.2.2 節の content perceptual loss では conv4_1，conv5_1 の出力，3.2.3 節の style perceptual loss では conv1_1，conv2_1，conv3_1 の出力をそれぞれ用いた． 式 (11) の各重みの値は，$w_1 = 1, w_2 = 10^{-6}, w_3 = 5 \times 10^{-5}, w_4 = 30, w_5 = 1$ とした．3.2.2 節の content perceptual loss と 3.2.2 節の content perceptual loss は，他の loss 関数と比較して大きな値になるので，重みの値は他の loss 関数と比較して小さな値にし，桁数がそろうようにした [2].

E，G，D，P のネットワークの学習には，Adam 法を最適化手法として用いた．学習率は 0.001 とし，minibatch は 100 として実装した．各ネットワークの学習と推論は，1 台の NVIDIA Tesla V100 GPU 上で実装した．

4.4 実験結果

学習済みの提案モデルから，以下の図 4 で示されているように入力画像同士の融合が成功していることがわかった．$\mathbf{\Lambda}$ の各要素の値を調節して，入力画像の融合度合いを調節しながら，表示した．$\mathbf{\Lambda}$ の各要素が全て 1 のときは，入力画像 A が再構成され，全て 0 のときは，入力画像 B が再構成される．0〜1 の値では，1 に近い値は入力画像 A に寄った融合，0 に近い値は入力画像 B に寄った画像となる．課題としては，形状の異なる画像では，融合がうまくいかないところである．例えば，正面を向いた画像同士の融合は成功するが，横に向いた画像と正面に向いた画像の融合はうまくいかない．また，潜在変数ベクトルの各要素に関しても，$\mathbf{\Lambda}$ の各要素の値を動かしても変化がとれない要素が出てきた．

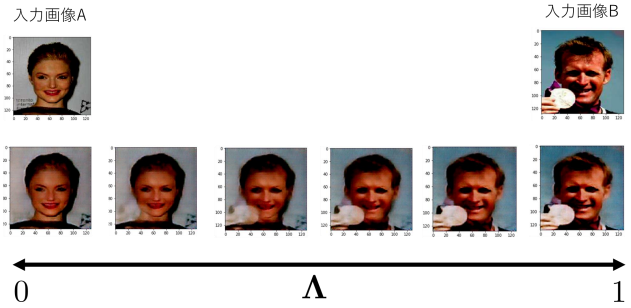

図 4: 一行目の，左が入力画像 A，右が入力画像 B である．二行目の画像は提案モデルから得られた生成画像である．生成画像は，$\mathbf{\Lambda}$ の全要素を，左から右まで 1〜0 の値を 0.2 ずつ動かして表示した．

5. まとめ

本稿では，指定した異なる入力画像同士を融合する手法を提案した．また，潜在変数ベクトルの演算に超パラメータを導入したことで，生成する画像の融合度合いの調節も可能にした．予備実験において，超パラメータの値が 1 のときは入力画像 A が得られ，0 のときは入力画像 B がきちんと生成された．超パラメータの値が 1〜0 のときは，融合生成された画像が生成可能であることもわかった．ただし，形状の異なる画像同士の融合がうまくいかないのは今後の課題である．目だけ融合させたいなど，より細かな融合の操作を可能にするためには，モデルの改良も必要だと考えられる．

参考文献

[1] Goodfellow, I., Pouget-Aradie, J.,Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. Generative Adversarial Nets. *Advances in Neural Information Processing Systems* , Vol. 2, pp. 2672–2680, 2014.

[2] Zhang, R., Tang, S., Li, Y., Guo, J., Zhang, Y., Li, J. and Yan, S. Style Separation and Synthesis via Generative Adversarial Networks. *The 26th ACM international conference on Multimedia* , pp.183–191, 2018.

[3] Suzuki, R., Koyoma, M., Miyato, T., Yonetsuji, T. and Zhu, H. Spatially Controllable Image Synthesis with Internal Representation Collaging. *arXiv:1811.10153*

[4] Arjovsky, M., Chintala, S. and Bottou, L. Wasserstein Generative Adversarial Networks. *Proceedings of the 34th International Conference on Machine Learning* , Vol. 70, pp. 214–223, 2017.

[5] Johnson, J., Alahi, A. and Fei-Fei, L. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *European Conference on Computer Vision* , Vol 9906, pp. 694–711, 2016.

[6] Johnson, J., Alahi, A. and Fei-Fei, L. Visualizing and Understanding Convolutional Networks. *European Conference on Computer Vision* , Vol 8689, pp. 818–833, 2014.

[7] Liu, Z., Luo, P., Wang, X. and Tang, X. Deep Learning Face Attributes in the Wild. *IEEE International Conference on Computer Vision*, pp. 3730–3738, 2015.

[8] Deng, J., Dong, W., Socher, R., Li, L., Li, K. and Li, F. ImageNet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition.*, 2019.

[9] Simonyan, K. and Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. . *arXiv preprint arXiv:1409.1556*, 2014.

6. 付録

形状の異なる画像同士を融合させた場合の生成画像の結果は，図 6 で示されているように，融合画像がいびつになる．

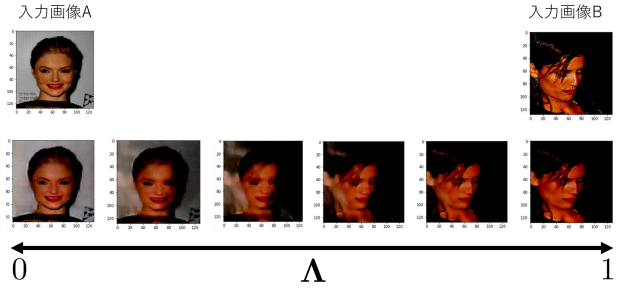

図 5: 一行目の，左が入力画像 A，右が入力画像 B である．二行目の画像は提案モデルから得られた生成画像である．生成画像は，$\mathbf{\Lambda}$ の全要素を，左から右まで 1〜0 の値を 0.2 ずつ動かして表示した．

P の各層から抽出される特徴 map は画像の様々な特徴を保持している．深い層の特徴 map では画像の輪郭に該当する特徴を保持している．浅い層の特徴 map では，画像の色合いに該当する特徴を保持している．

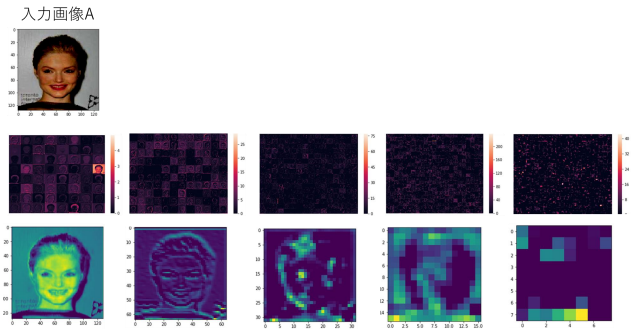

図 6: 一行目が入力画像 A．二行目の画像は P から抽出された特徴 map を全て表示している．三行目は二行目の各列の特徴 map の一部を表示している．二行目，三行目の各列に該当する特徴 map の抽出された層は，一列目:conv1_1，二列目:conv2_1，三列目:conv3_1，四列目:conv4_1，五列目:conv5_1 である．