Conditional GAN による食事写真の属性操作

成冨　志優† 堀田　大地†† 丹野　良介††† 下田　和††† 柳井　啓司†††

† 電気通信大学 情報理工学域 1 類 メディア情報学プログラム

†† 電気通信大学 情報理工学域 3 類 機械システムプログラム

††† 電気通信大学 大学院情報理工学研究科 情報学専攻

E-mail: †{n1610480,h1610581}@edu.cc.uec.ac.jp, ††{tanno-r,shimoda-k,yanai}@mm.inf.uec.ac.jp

あらまし　本論文では，深層学習技術を用いて，自動的に食事画像を生成するという新しい問題に焦点を当てる．食事画像と変換先のカテゴリ情報を入力すると，リアルタイムに特定のカテゴリに変換された食事画像を生成する．本研究は近年盛んに研究が行われている Neural Style Transfer [2] や Generative Adversarial Network [3] を用いた画像生成及び画像変換に関連しているが，食事画像生成というタスクは独自の課題を含んでいる．変換された画像は入力画像の形状を維持し，期待する変換先のカテゴリ情報を反映する必要がある．また，深層学習においては，学習に用いるべき画像枚数が多ければ多いほど良いとされるが，各カテゴリについて必ずしも十分な画像枚数を用意できない場合が多くある．その為，学習枚数が少ないカテゴリへの変換が適切に行われない可能性がある．本論文では，Image-to-Image 変換手法の 1 つである CycleGAN [16] の手法を拡張し，1 つの Generator で複数のカテゴリへと変換可能とする conditional CycleGAN を用いた食事画像変換手法を提案する．CycleGAN [16] ベースの手法を取ることで，入力画像の形状を維持したまま，特定のカテゴリへの変換を実現した．また，1 つの生成器が複数のカテゴリへの変換を担うことで，画像枚数が少ない特定のカテゴリが存在した場合でも，どのカテゴリへも一定の質を保って変換することが可能となる．評価実験では，食事画像変換タスクにおいて，[15] で用いられている Neural Multi Style Transfer による変換手法と比較して，CycleGAN ベースの本手法を用いた場合の方が本タスクにおいて有効な結果が得られた．

キーワード　Generative Adversarial Network，深層学習，食事画像生成，画像変換，アプリケーション

1. はじめに

近年，生成モデルと深層学習を組合せた深層生成モデル Generative Adversarial Networks(GAN) が従来手法と比べてより本物らしい画像を生成できるとして注目を集めている．訓練データの分布に近似するよう最適化することで本物らしい画像の生成に成功している．GAN の研究において用いられるデータセットは CelebA データセットの顔画像や MNIST の数字文字画像，LSUN の居住画像など，ある程度パターンが限られる画像群が通常用いられる．また，最近では，[7] のように衣服画像へのデザイン転送タスクといった新しい課題を提案し，GAN や Neural Style Transfer のような深層学習技術を応用する研究がでてきている．一方で，本研究のような食事に限定した食事画像生成・変換に関する研究は未だ存在しないのが現状である．

2. 目　　的

本研究では，深層学習技術を用いて，自動的に食事画像を生成・変換するという新しい問題に焦点を当てる．食事画像と変換先のカテゴリ情報を入力すると，リアルタイムに特定のカテゴリに変換された食事画像を生成することを目指す．Image-to-Image 変換手法の 1 つである CycleGAN の手法を拡張し，図 1 のように 1 つの Generator で複数のカテゴリへと変換可能とする conditional CycleGAN を用いた食事画像変換手法を提案し，[15] で用いられている Neural Multi Style Transfer による変換手法と比較することで，食事画像生成・変換タスクにおいて本手法の有効性を示す．

3. 関 連 研 究

GAN は一様分布や正規分布などからノイズベクトル z をサンプリングするため，生成される画像のコントロールをすることができない．そこで，GAN の構造に条件付き信号 conditional vector を付与することで，条件付き確立分布を学習するモデルに拡張したものが cGAN である．一方で，cGAN には入力画像を潜在表現に落とし込む機構 (Encoder) が欠けているため，画像の変換は行うことができない．pix2pix は Adversarial Loss と ConvDeconvNet を組合せることで，画像のペア集合間の変換方法を学習することが可能となり，線画彩色や白黒画像のカラー化などの変換を学習させることができる．

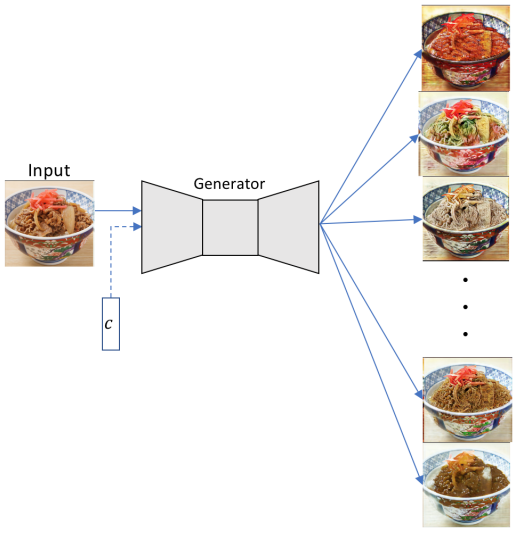

図 1　conditional CyleGAN の概念図

[16] では学習データ間 X, Y の写像を学習する方法が提案された．通常の GAN で用いられる損失関数に再構築誤差である

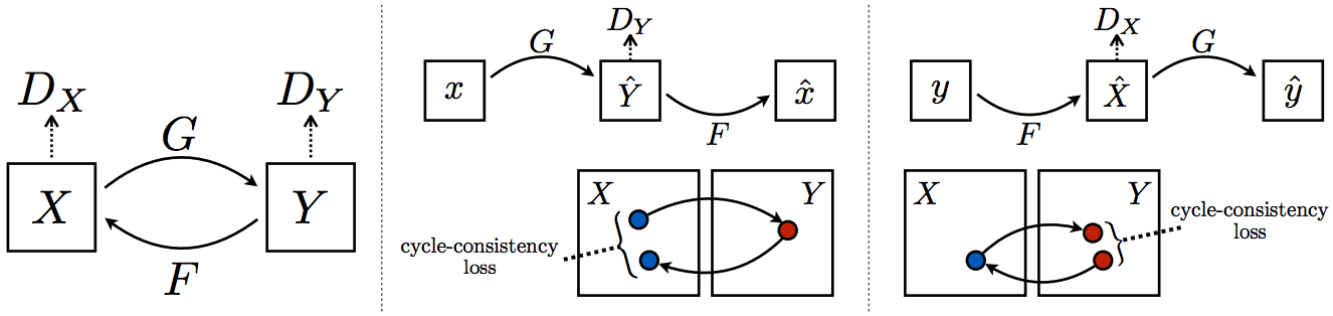

図 2　CycleGAN の構造 ([16] から引用)

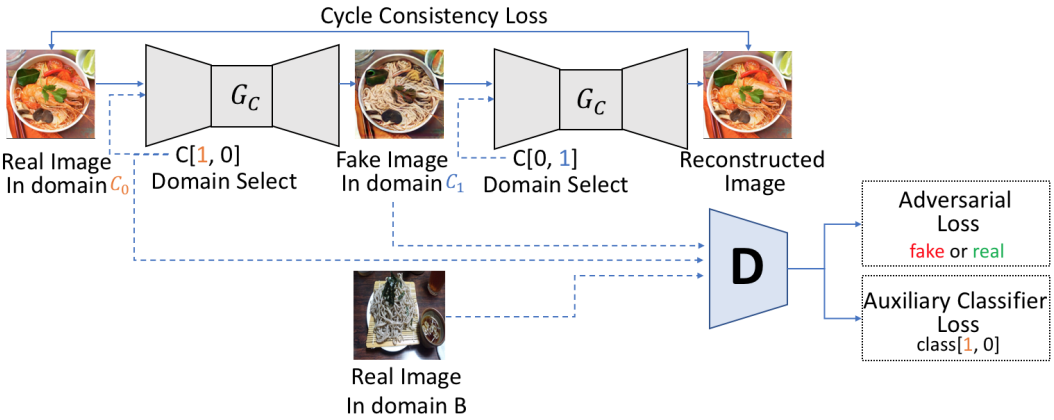

図 3　conditional CycleGAN のネットワーク全体

Cycle Consistency Loss を追加することで,「集合 X, Y に共通する構造を保って」変換する写像関数の学習に成功している．よって，本研究においても Cycle Consistency Loss による制約を設けることで,「集合 X, Y に共通する構造を保って」，つまりは，食事画像であるならば，食事の部分のみ別のカテゴリの食事に変換し，それ以外は，元の形状を保ったまま変換されることが可能になると考えた．

4.　生成・変換手法

4.1　conditional CycleGAN による方法

4.1.1　pix2pix

Cycle Consistency Loss とは CycleGAN [16] により提案された損失の 1 つである．先行研究である pix2pix [6] は conditional GAN の一種であるが，通常の GAN では，一様分布や正規分布からサンプリングしたノイズベクトル z を Generator への入力とするが，pix2pix や後述する CycleGAN では，画像 x を Generator の入力とする点が大きく異なる点である．入力に用いていた乱数 z は直接サンプリングする代わりに Generator の複数の層に Dropout [14] という形でノイズを加えるように代替されている．

pix2pix では，式 1 で表される cGAN の損失関数に加えて，より本物らしい画像を生成するために，式 2 の L1 正則化項の追加と Discriminator のベース構造に [12] で提案された PatchGAN を組合せた式 3 が最終的な pix2pix の損失関数となる．入力には変換前と変換後の画像のペアを必要とし，(変換元画像，変換先画像) or (変換元画像，Generator が生成した画像) のいずれのペアであるかを Discriminator に判断させるように学習する．

$$L_{cGAN}(G, D) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}[\log D(x, y)] + \mathbb{E}_{\boldsymbol{x},\boldsymbol{z}}[\log(1 - D(x, G(x, z)))] \tag{1}$$

$$L_{L_1}(G) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}[||y - G(x, z)||_1] \tag{2}$$

$$G^* = arg \min_G \max_D L_{GAN}(G, D) + \lambda L_{L_1}(G) \tag{3}$$

4.1.2　CycleGAN

pix2pix [6] では，変換前と変換後の画像の 1 対 1 ペアを必要とする制限があったが，CycleGAN [16] ではドメイン間の写像を学習できるように拡張することで，1 対 1 に対応せずとも学習が行えるのが特徴である．ここで，図 2 のようにあるドメイン X とドメイン Y があるとして，$X \to Y$ への写像を G，その逆写像 $Y \to X$ を F と定義する．また，入力が G によって生成された偽物の X か元の X のデータかを判別する D_Y，入力が Y によって生成された偽物の Y か元の Y のデータかを判別する D_X をそれぞれ定義する．この G, F, D_X, D_Y を式 4 と式 5 の 3 つの損失の和で表される式 6 を用いて学習する．式 4 は Vanilla GAN で用いられる Adversarial Loss そのままであるが，式 5 は Cycle Consistency Loss と呼ばれるもので，ドメイン X に属する x から生成された $\hat{Y}$ を再度，ドメイン X に属する $\hat{x}$ に戻しても元のドメイン X に一致するように制約をかけるものである．この Cycle Consistency Loss を小さくすることは，$G(F(x))$ により変換した結果がそれぞれ元のデータを再構築できるだけの情報を保持することを意味する．よって，学習に成功した場合は，$G(F(x))$ とした場合,「ドメイン X とドメイン Y に共通する構造を保ったまま，一方のドメインに属するデータをもう一方のドメインのデータに変換する」写像関数が得られることになる．

$$L_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{\boldsymbol{y}\sim p_{data}(\boldsymbol{y})}[\log D_Y(y)] + \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log(1 - D_Y(G(x)))] \tag{4}$$

$$L_{cyc}(G, F) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(x)}[||F(G(x)) - x||_1] + \mathbb{E}_{\boldsymbol{y}\sim p_{data}(y)}[||G(F(y)) - y||_1] \tag{5}$$

$$L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F) \tag{6}$$

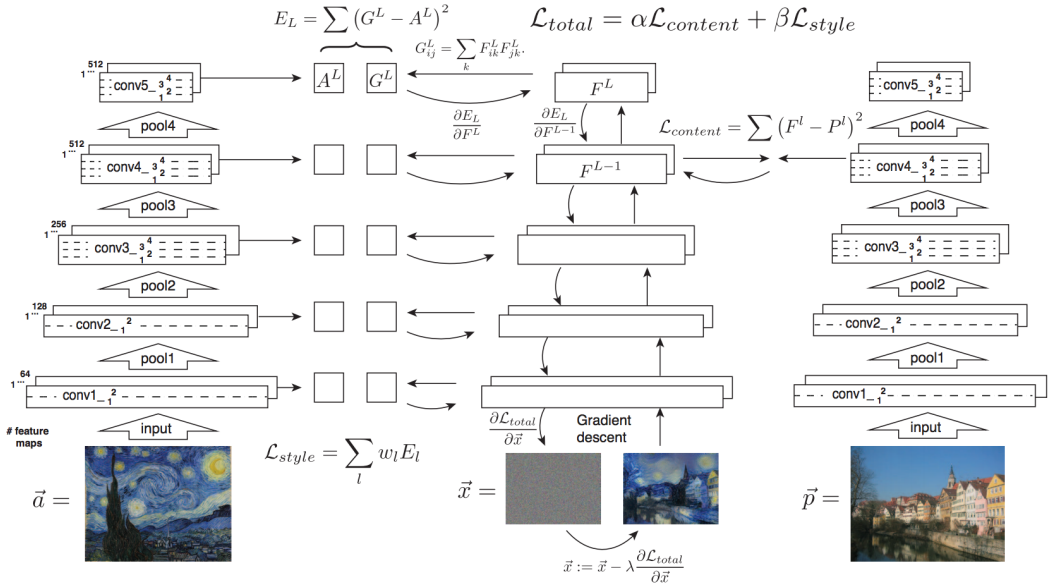

図 4　順伝搬及び誤差逆伝播を複数回繰り返すため計算コストが高い ([2] より引用)

4.1.3　conditional CycleGAN

図 3 として conditional CycleGAN(cCycleGAN) の模式図を示す．CycleGAN [16] を conditional 化することで，1 つの Generator で複数のカテゴリへと変換可能とする conditional CycleGAN に拡張してある．CycleGAN の conditional 化には，[1] と同様に [11] で提案されている分類誤差項 Auxiliary Classifier Loss を Discriminator に追加することで実現する．本物か偽物かの判断をさせるだけでなく，Discriminator にどのカテゴリに属する画像かの識別も同時に学習させることで，複数のカテゴリに変換可能な Generator の学習を行った．こうすることで，Generator は単に Discriminator を欺くように画像を生成するだけでなく，Discriminator の識別エラーを最小限に抑えるように偽物のサンプルを生成できるようになる．つまり，各カテゴリのサンプルを生成できるように最適化されることを意味する．

よって，最終的な損失関数は，Adversarial Loss L_{adv} に式 7 で表される Cycle Consistency Loss と式 8，式 9 で表される Auxiliary Classifier Loss にそれぞれの重みバイアス項 λ_{ccl} 及び λ_{acl} を追加した式 10，式 11 を conditional CycleGAN の損失関数として用いた．

$$L_{ccl} = \mathbb{E}_{x,c,c'}[||x - G(G(x,c),c')||_1] \tag{7}$$

$$L_{acl}^{real} = \mathbb{E}[-\log D_{acl}(c'|x)] \tag{8}$$

$$L_{acl}^{fake} = \mathbb{E}_{x,c}[-\log D_{acl}(c|G(x,c))] \tag{9}$$

$$L_D = L_{adv} + \lambda_{acl} L_{acl}^{real} \tag{10}$$

$$L_G = L_{adv} + \lambda_{acl} L_{acl}^{fake} + \lambda_{ccl} L_{ccl} \tag{11}$$

4.2　Neural Multi Style Transfer による方法

Gatys らが提唱した “Neural Style Transfer”(スタイル変換) を発端として，この分野に関する研究は急速に進んでいる．Gatys らの手法では，図 4 のように，コンテンツ画像，スタイル画像，目的画像をそれぞれ，$\vec{p}, \vec{a}, \vec{x}$ として，一様乱数で初期化した目的画像 $\vec{x}$ を入力した時の n_m 番目の中間層の出力の i 番目のチャンネルの座標 j の値を $\mathrm{Conv}_{n_m}(\vec{x})_{i,j}$ とした時，学習済み CNN(VGG-16 or VGG-19) の特徴マップ間の相関行列は式 12 のグラム行列 G を用いて

$$G(\vec{x})^l_{i_1,i_2} = \sum_k (conv_l(\vec{x})_{i_1,k} conv_l(\vec{x})_{i_2,k}) \tag{12}$$

と表される．このグラム行列 G を用いてコンテンツ損失 $L_{content}$ とスタイル損失 L_{style} は式 13, 式 14 で表される．式 14 中の A_l は各層の画素数などの差異を吸収する係数を表す．

$$L_{content}(\vec{x},\vec{p}) = \sum_{i,j} (conv_{4_2}(\vec{x})_{i,j} conv_{4_2}(\vec{p})_{i,j})^2 \tag{13}$$

$$L_{style}(\vec{x},\vec{a}) = \frac{1}{A_l} \sum_{n=1}^{5} \sum_{i_1,i_2} (G(\vec{x})^{n_1}_{i_1,i_2} - G(\vec{a})^{n_1}_{i_1,i_2})^2 \tag{14}$$

目的画像の生成は，まず，$\vec{x}$ を一様乱数で初期化を行い，$L_{content}, L_{style}$ の線形和 (式 15) を最小化するよう勾配降下法 (式 16) により最適化を行う．

$$L_{total}(\vec{x},\vec{p},\vec{a}) = \alpha L_{content}(\vec{x},\vec{p}) + \beta L_{style}(\vec{x},\vec{a}) \tag{15}$$

$$\vec{x} \leftarrow \vec{x} - \lambda \frac{\partial L_{total}}{\partial \vec{x}} \tag{16}$$

Gatys らの手法の発想の着眼点は，コンテンツ画像の信号が CNN の各層を順伝播している間に劣化する情報を，スタイル画像から抽出されたスタイル情報により置き換えることにある．

しかしながら，Gatys らの手法では，図 4 にあるように feed-forward 及び back propagation を複数回繰り返すため，GPU を使う場合でも，画像の生成に数十秒程度要するなど，処理に時間がかかる．

この問題を解決するために，feed-forward のみを使ってスタイル変換を高速に行う研究が世界中で多くなされている．

Johnson ら [8] は図 7 に挙げるような feed-forward style transfer network として，Downsampling 層，複数の Residual block，Upsampling 層から構成される ConvDeconvNetwork f_w を学習する “perceptual loss” を提案した．まず，図 7 のように変数が設定されているとして，層の Feature Loss は式 17 で表される．この時，式 17 中の C, H, W はそれぞれ Channel, Height, Width を表す．

$$l^{\phi,j}_{feat}(\hat{y},y_c) = \frac{1}{C_j H_j W_j} \|\phi_j(\hat{y}) - \phi_j(y_c)\|_2^2 \tag{17}$$

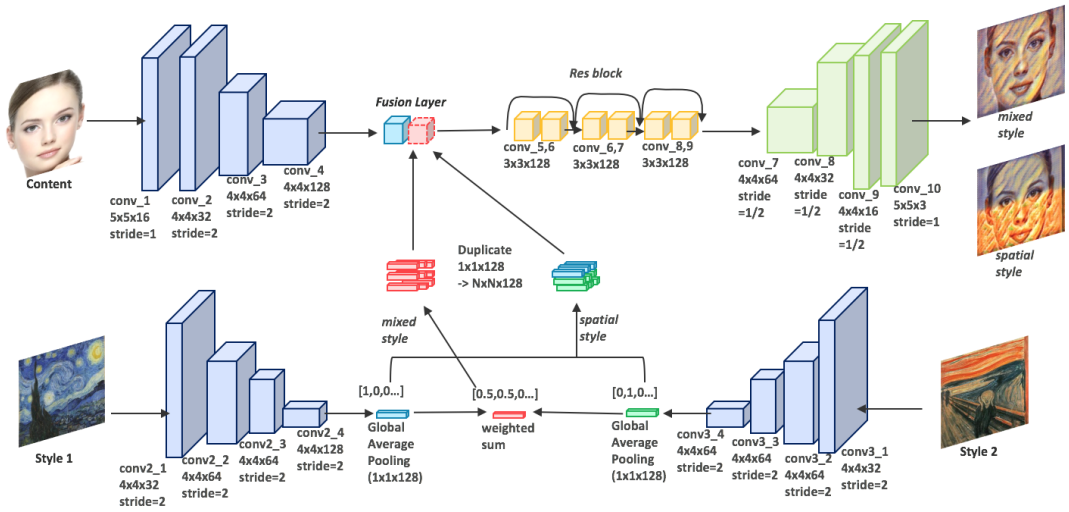

図 5　multi style 変換ネットワーク ([15] より引用)

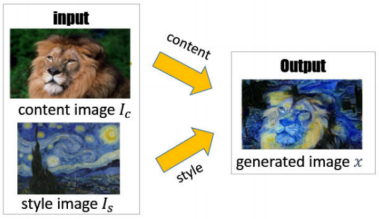

図 6　Neural Style Transfer アルゴリズム．スタイル画像の画風をコンテンツ画像に転送可能

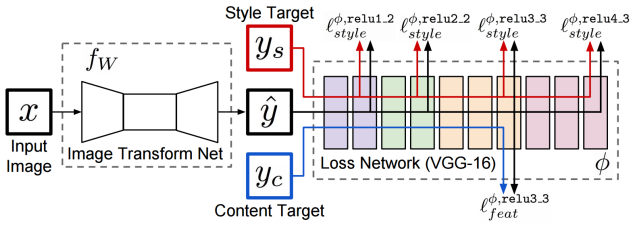

図 7　特定のスタイルの変換を feed-forward で行う CNN を学習することで高速スタイル変換を実現 ([8] より引用)

また，各層におけるグラム行列 (式 18) を用いて Style Loss は式 19 で表される．

$$G^{\phi}_{j}(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'} \quad (18)$$

$$l^{\phi,j}_{style}(\hat{y}, y_s) = \|G^{\phi}_{j}(\hat{y}) - G^{\phi}_{j}(y_s)\|^2_F \quad (19)$$

式 17 と式 19 の線形和に $\hat{y}$ の分散を減少させる正則化項 $\lambda_{TV} l_{TV}(y)$ を追加した式 20 を最小化するように f_w の学習を行う．

$$\hat{y} = \arg\min_{y} \lambda_c l^{\phi,j}_{feat}(y, y_c) + \lambda_s l^{\phi,J}_{style}(y, y_s) + \lambda_{TV} l_{TV}(y) \quad (20)$$

これにより，特定のスタイルの変換を feed-forward で行う CNN を学習しておくことで，Gatys らの手法と比較して約 1000 倍 (500 iterations) ほど高速に画像を変換可能にする．しかし，この手法では，スタイル変換ネットワーク f_w はスタイル毎に学習する必要があり，1 つのモデルで単一のスタイルしか表現することができない．そのため，消費メモリの増大，学習に時間がかかる，変換の質が画像の質に依存する，などの問題点がある．よって，本論文では 1 つのモデルで複数のスタイルに変換可能なように拡張した Neural Multi Style Transfer [15] という手法を用いることにした．

まず，図 5 上段のメインネットワークである ConvDeconvNet に図 5 下段にある通りスタイル転送 Net を追加する．各スタイルをスタイル転送 Net に入力し，128 次元実数値スタイルベクトルを事前に算出し，線形重ね合せにより複数のスタイル重みを表現するスタイルベクトルを求める．その後，ConvDeconvNet の中間層における特徴量マップに，事前に求めたスタイルベクトルをメインネットワークの特徴量マップと同じサイズ分コピーを作成し，Fusion Layer で結合させることで，複数スタイルの同時学習を可能にした．この Fusion Layer は飯塚ら [4] の研究により発想を得ている．スタイル転送 Net の最後の層に Global Average Pooling(GAP) を用いることで，各チャネルの平均 (≃ スタイル表現) を出力し，変換後の画像の質が向上することから (Li ら [9])，本ネットワークでも GAP を用いている．ネットワークの学習方法は基本的には Johnson ら [8] と同じであり，損失関数として pre-trained VGG-16 [13] を用いて，各スタイル画像と約 8 万枚のコンテンツ画像 (MS-COCO) により学習を行う．

5. 実　　験

5.1 学習データ

5.1.1 conditional CycleGAN による方法

Cycle Consistency Loss を追加することで，「集合 X, Y に共通する構造を保って」変換することが可能である．そのため，学習データに「共通する構造」がある方が変換が上手くいくと推測される．よって今回は，「丼」という制約を設けて UECFOOD-100 [10] の 100 カテゴリの食事の中から「丼」の構造をもつ 10 個のカテゴリを選出した．その 10 カテゴリについて高品質な食事画像の選別のために，Twitter からクロールした画像データベースの中から UECFOOD-100 [10] で学習した食事認識エンジンを用いて，各カテゴリ毎に認識精度が高い順にリランキングした結果から表 1 にある枚数分を学習データとした．この中で「ラーメン」のカテゴリに至ってはその種類の多様性が他のカテゴリと比べて高かったため（例えば，「二郎系のラーメン」は基本的に「丼」からはみ出るほどの具材が乗っているため，他のラーメンと比べて差が大きい．つまり，「共通する構造」が同カテゴリであるが，差が大きくなってしまい，学習が難しくなる恐れがある．)，図 8 の処理を行った．8 万枚の「ラーメン」画像に対して，ImageNet で学習済みの VGG16 を特徴抽出器として用い，224x224x3(150,528 次元) を fc6 層

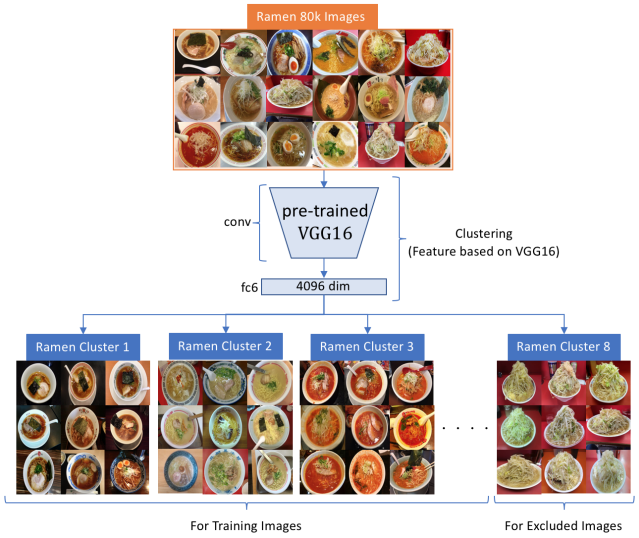

図 8　多様性があるカテゴリに対するクラスタの構築

(4,096 次元) まで特徴量を圧縮して k-means により k 個のクラスタ (今回は，k=8 に設定した) に分割を行った．想定していた通り,「二郎系ラーメン」が大部分を占めるクラスタを得られたため，そのクラスタを除外した画像を「ラーメン」カテゴリの画像とした．全ての学習において，訓練 9 割，テスト 1 割となるように配分した．

表 1　学習データ

カテゴリ	学習枚数
冷やし中華	13,499
ミートスパゲティ	7,138
蕎麦	3,530
ラーメン	74,007
焼きそば	24,760
白飯	21,324
カレーライス	34,216
牛丼	18,396
うな重	5,329
炒飯	27,854
合計	230,053

5.1.2　Neural Multi Style Transfer による方法

約 8 万枚の content 画像 (MS-COCO) と表 1 のカテゴリの中から style 画像を選出して学習に用いた．学習に用いる style 画像は各カテゴリで 1 枚のみしか利用しないため，用いた style 画像が変換結果の質にダイレクトに反映してしまう．その為，最良の style 画像表現を求める手段が必要であるが，現状，style 変換に用いる最適な style 画像を選択する手法は確立されていない．例えば，複数の style 画像を統合して 1 つの style 画像とすることを考える．k 枚の異なるスタイル画像を用いる場合のスタイル表現の統合方法として，単純な各画像の総和，平均値や [17] の用いたコンテンツ画像に基づく最適化ではスタイル変換の質が低下したとの報告があった．また，[5] のように画像集合から集合中の画像に共通する style を学習し，学習された style を用いる方法も提案されている．

今回は，以下の 2 つのパターンの style 画像を学習に用いることにした．学習に用いた 2 パターンの style 画像の表現を図 9 に示す．

（1）　表 1 のカテゴリの中から代表 style 画像を主観で 1 枚ずつ選出．

（2）　複数の style 画像の統合．

5.2　学習モデル構造

5.2.1　conditional CycleGAN による方法

conditional CycleGAN のネットワークの詳細を表 2 表 3 に示す．Generator 部分を [8] で提案された ConvDeconvNet の中間層に Residual Block を何層も積層する FastStyleNet の構造を用いて 256x256 の画像を学習に用いた．また，Discriminator には [12] で提案された PatchGAN を採用してある．重みの更新頻度は Discriminator を 5 回更新した後に Generator を 1 回更新するようにした．学習は NVIDIA Quadro P6000 を利用しバッチサイズ 32，最適化手法には Adam を用いて 20epoch 繰り返した．テスト時は 512x512 の画像を生成するようにした．

表 2　conditional CycleGAN の Generator Architecture

Layer	Kernel	Stride	Filters	Batch Norm	Activation
Concatenation				No	
Convolution	7 x 7	1	64	Yes	Leaky ReLU
Convolution	4 x 4	2	128	Yes	Leaky ReLU
Convolution	4 x 4	2	256	Yes	Leaky ReLU
Residual Block	3 x 3	1	256	Yes	Leaky ReLU
Residual Block	3 x 3	1	256	Yes	Leaky ReLU
Residual Block	3 x 3	1	256	Yes	Leaky ReLU
Residual Block	3 x 3	1	256	Yes	Leaky ReLU
Residual Block	3 x 3	1	256	Yes	Leaky ReLU
Residual Block	3 x 3	1	256	Yes	Leaky ReLU
fractionally-strided					
Convolution	4 x 4	2	128	Yes	Leaky ReLU
fractionally-strided					
Convolution	4 x 4	2	64	Yes	Leaky ReLU
Convolution	7 x 7	1	3	Yes	Tanh

表 3　conditional CycleGAN の Discriminator Architecture

Layer	Kernel	Stride	Filters	Batch Norm	Activation
Convolution	4 x 4	2	64	No	Leaky ReLU
Convolution	4 x 4	2	128	No	Leaky ReLU
Convolution	4 x 4	2	256	No	Leaky ReLU
Convolution	4 x 4	2	512	No	Leaky ReLU
Convolution	4 x 4	2	1024	No	Leaky ReLU
Convolution	4 x 4	2	2048	No	Leaky ReLU
Convolution(For Adv)	3 x 3	1	1	No	
Convolution(For Aux)	2 x 2	1	10	No	

表 4　Neural Multi Style Transfer Architecture

Layer	Kernel	Stride	Filters	Batch Norm	Activation
Convolution	9 x 9	1	32	Yes	Leaky ReLU
Convolution	4 x 4	2	64	Yes	Leaky ReLU
Convolution	4 x 4	2	64	Yes	Leaky ReLU
Convolution	4 x 4	2	128	Yes	Leaky ReLU
Concatenation				No	
Residual Block	3 x 3	1	128	Yes	Leaky ReLU
Residual Block	3 x 3	1	128	Yes	Leaky ReLU
Residual Block	3 x 3	1	128	Yes	Leaky ReLU
Residual Block	3 x 3	1	128	Yes	Leaky ReLU
Residual Block	3 x 3	1	128	Yes	Leaky ReLU
Convolution	4 x 4	1/2	64	Yes	Leaky ReLU
Convolution	4 x 4	1/2	64	Yes	Leaky ReLU
Convolution	4 x 4	1/2	32	Yes	Leaky ReLU
Convolution	9 x 9	1	3	Yes	Tanh

5.2.2　Neural Multi Style Transfer による方法

Neural Multi Style Transfer に用いた ConvDeconvNet の詳細を表 4 に示す．

5.3　結　　果

5.3.1　conditional CycleGAN による食事画像変換

本手法により変換した結果を図 10 に示す．最左列を入力画像として，最上部の 10 カテゴリのドメインへ同時に変換した例を示してある．食事が複数品目ある場合に対しても正確に食事領域のみ対象のドメインへと変換できていることがわかる．再構築誤差 Cycle Consistency Loss により「ドメイン X ドメイン Y に共通する構造を保ったまま，あるドメインに属するデータをもう一方のドメインのデータに変換する」写像関数の学習に成功し,「共通構造=丼，器，食器」の概念を Generator が獲得していることを意味する．また，分類誤差 Auxiliary Classifier Loss を導入することで Generator は単に Discriminator を欺くように画像を生成するだけでなく，Discriminator の分類エラーを最小限に抑えるように偽物のサンプルを生成できるようになり，各ドメインのサンプルを生成できるように最適化されることで，歪みや GAN に特有のブラーが掛かっていない高いクオリティで変換できていることがみてとれる．

5.3.2　Neural Multi Style Transfer による食事画像変換

本手法により変換した結果を図 11，図 12 に示す．図 11 の変換結果 (4 行 10 列分) は content 画像の多くが赤色に反応してしまっており,「ミートスパ」の style 表現に引っ張られているような結果が得られた．一方で，複数の style 表現を統合した style 画像を用いた図 12 の場合は，図 11 よりも多様な style 表現を変換結果に反映していることがみてとれる．例えば,「カレー」の style への変換結果は，mix style 画像の中に青色の一風変わったカレー画像が含まれているが，変換結果の中にも青色成分が含まれており，mix することで多様な表現を content 画像に反映できていると考える．

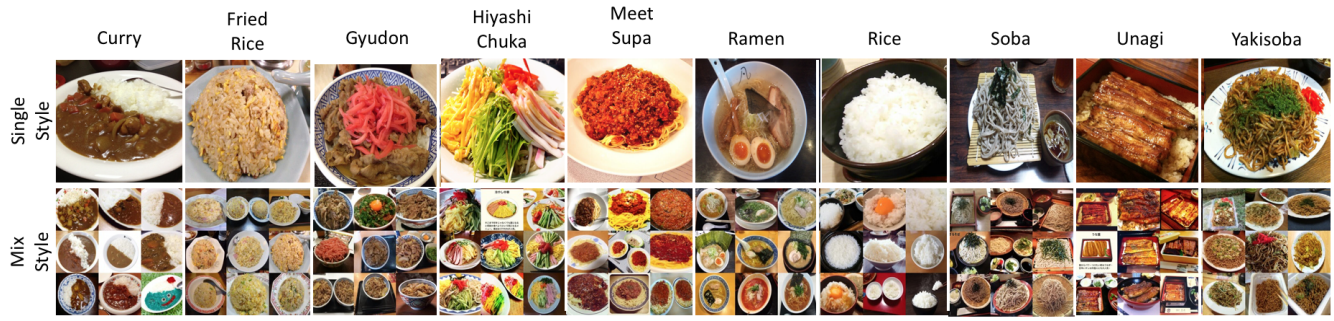

図 9　学習に用いる Single Style と Mix Style

図 10　cCycleGAN による食事画像変換結果

変換のクオリティについては，cCycleGAN のように再構築誤差 Cycle Consistency Loss による制約がないため，本来は変換に不必要な背景領域も style 表現に変換されてしまっているなど,「共通の構造を保って」変換するという目的には現状の style 変換のままでは，食事画像変換には適さないと思われる．

6. 考　　察

6.1 学習に用いるデータ数の変化による変換結果のクオリティへの影響

学習に用いるデータ数が表 1 と比べて小規模な場合，変換結果のクオリティがどのように変化するかの考察を行った．学習に用いるデータセットは 3 種類あり，まとめると以下のようになる.「白飯」「冷やし中華」の 2 カテゴリについて，1 カテゴリあたりの画像枚数，総画像枚数がどのようなクオリティを齎すかの考察を行った．

（1） 1 カテゴリ 1 千枚．合計 1 万枚のデータセット．
（2） 1 カテゴリ 1 万枚．合計 10 万枚のデータセット．
（3） 表 1 の合計約 23 万枚のデータセット．

各条件により学習したモデルで変換した画像を総学習枚数が少ない順に左から並べたものを図 13 に示す．各カテゴリ千枚の比較的小規模なデータセットでも変換先ドメインの大域的特徴を捉えることには成功しているが，局所的にみると細かいディテールまでは再現できていないようにみえる．つまり，画像枚数が多ければ多いほど，大域的特徴に加えて局所的な特徴をもった細部の細かい部分まで正確に変換先のドメインに変換可能な写像関数の学習ができている結果となった．また，図 13 のカテゴリ「冷やし中華」の変換結果に着目すると，1 列目は 1 千枚，2 列目は 1 万枚，3 列目は表 1 にある通り，1.3 万枚と 2 列目と 3 列目で画像枚数は 3 千枚ほどしか変わらない．しかし，2 列目より 3 列目の方が細かい部分まで変換できていることがわかる．一方で,「冷やし中華」以外の画像枚数も考慮すると，2 列目は総学習枚数 10 万枚に対し，3 列目は 23 万枚の大規模データセットを用いている．他カテゴリの画像から得られた特徴も上手く変換結果に反映されていることがこの結果から伺えるが，これは，1 つの Generator で複数のカテゴリに変換可能にすることで,「食事変換」という共通特徴を Generator が獲得していることを意味する．つまり，1 つの生成器が複数のカテゴリへの変換を担うことで，画像枚数が少ない特定のカテゴリが存在した場合でも，どのカテゴリへも一定の質を保って変換することが可能としていることになる．

7. まとめと今後の課題

本研究では，深層学習技術を用いて，自動的に食事画像を生成・変換するという新しい問題を提唱し，Generative Adver-

図 11　single style 画像を用いたスタイル変換による食事画像変換結果

図 12　mix style 画像を用いたスタイル変換による食事画像変換結果

sarial Networks 及び Neural Style Transfer の両側面から課題の解決に挑戦した研究である．Image-to-Image 変換手法の 1 つである CycleGAN の手法を拡張した conditional CycleGAN を用いることで，

（1） 変換前と変換後で共通構造を保ったままの変換

（2） 複数のカテゴリへの変換を行うことで，変換カテゴリの共通特徴の獲得による変換クオリティの向上

を実現することで，高品質に食事画像を変換可能な手法を提案した．また，GAN とは別の手法である Neural Multi Style Transfer [15] による変換手法と比較することで，食事画像生成・変換タスクにおいて本手法の有効性を示した．

今後の課題としては，現状，学習したモデルの有効性を示すために，主観的な定性評価しか行っていないため，他者による客観評価実験や変換した画像が期待するターゲットドメインへと変換できているかについて，食事画像分類問題を解くことで定量評価としたい．また，本研究で学習したモデルを用いてモバイルアプリとして実装する予定である．

文　献

[1] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. arXiv:1711.09020, 2017.

[2] L. A. Gatys, A. S. Ecker, and M. Bethge. Image Style Transfer Using Convolutional Neural Networks. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. In *Advances in Neural*

図 13　学習に用いるデータ枚数のクオリティへの影響結果．総画像枚数が少ない順に表示してある．(最左列は Input 画像，Input を除いた，左 3 列：「白飯」カテゴリ，右 3 列：「冷やし中華」カテゴリ)

Information Processing Systems, 2014.

[4] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. In *Proc. of SIGGRAPH*, 2016.

[5] H. Ikuta, K. Ogaki, and Y. Odagiri. Blending Texture Features from Multiple Reference Images for Style Transfer. In *Proc. of SIGGRAPH Asia Technical Briefs*, 2016.

[6] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2017.

[7] S. Jiang and Y. Fu. Fashion Style Generator. In *Proc. of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017.

[8] J. Johnson, A. Alahi, and L. Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Proc. of European Conference on Computer Vision*, 2016.

[9] Y. Li, N. Wang, J. Liu, and X. Hou. Demystifying Neural Style Transfer. In *Proc. of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017.

[10] Y. Matsuda, H. Hoashi, and K. Yanai. Recognition of Multiple-Food Images by Detecting Candidate Regions. In *Proc. of IEEE International Conference on Multimedia and Expo*, 2012.

[11] A. Odena, C. Olah, and J. Shlens. Conditional Image Synthesis With Auxiliary Classifier GANs. In *Proc. of the 34th International Conference on Machine Learning*, 2017.

[12] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. Context Encoders: Feature Learning by Inpainting. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[13] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *Proc. of International Conference on Learning Representations*, 2014.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[15] R. Tanno and K. Yanai. DeepStyleCam: A Real-time Style Transfer App on iOS. In *Proc. of International MultiMedia Modeing Conference*, 2017.

[16] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *Proc. of IEEE International Conference on Computer Vision*, 2017.

[17] 幾田光, 大垣慶介, and 小田桐優理. 畳み込みニューラルネットワークを用いたテクスチャ特徴量の混合に基づく自然なテクスチャ転写. In 第 *19* 回 画像の認識・ 理解シンポジウム, 2016.