社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

ラーメンスタイルエンコーダーを用いた スタイル特徴とマスク画像からの画像生成

趙　　宰亨† 下田　　和† 柳井　啓司†

† 電気通信大学 大学院情報理工学研究科 情報学専攻
E-mail: †{cho,shimoda-k,yanai}@mm.inf.uec.ac.jp

あらまし　近年、Web 上のブログや Twitter と Instagram のようなソーシャルネットワークサービス (SNS) などに大量の画像がアップロードされている。ユーザーは Web 上により魅力的な画像をアップロードしたいと考えて、画像をより魅力的に作るために画像を編集したり合成する。しかし、そのような画像の編集や合成は熟練した画像編集スキルと多くの時間を必要とする難しい作業である。一方、深層学習技術の発展により画像の生成や変換の技術が盛んに研究されている。特に GAN(Generative Adversarial Networks) [1] という高品質の画像を生成や変換する強力なフレームワークの登場により画像生成技術は大幅に改善された。本研究では、希望する形状とスタイルを基にした画像を生成することを目的にして、まず、GAN を基にした Image-to-image 変換ネットワークを使用して、スケッチ画像からリアルな画像の生成を行う。次に、新しい Style encoder を作成して、スタイルを考慮した画像生成をネットワークを作成した。実験では、ラーメン画像の実画像とマスク画像を学習して、1 つのスタイル画像のスタイル及び複数の特定のスタイル画像からスタイル特徴を抽出して、そのスタイルを考慮したラーメン画像の生成を行い、生成結果を評価した。

キーワード　Image-to-image 変換，スタイル変換，GAN，Style encoder

Image Synthesis Based on Style Features and Mask Images Using a Ramen Style Encoder

Jaehyeong CHO†, Wataru SHIMODA†, and Keiji YANAI†

† Department of Informatics, The University of Electro-Communications, Tokyo
E-mail: †{cho,shimoda-k,yanai}@mm.inf.uec.ac.jp

1. はじめに

近年、Web 上のブログや Twitter と Instagram のようなソーシャルネットワークサービス（SNS）などに大量の画像がアップロードされている。ユーザーは Web 上に画像をアップロードする時、より魅力的な画像をアップロードしたいから、または様々な目的で魅力的な画像を作りたいと考える。そのように画像を魅力的に作るために、元画像を編集したり、異なる画像を用いて画像を合成する場合がある。しかし、そのような画像を編集または合成する作業は熟練した画像編集のスキルと多くの時間を必要とする難しい作業である。

一方近年、深層学習の発展により様々な研究やタスクの精度が飛躍的に向上された。特に深層学習を用いた GAN(Generative Adversarial Networks) [1] という高品質の画像生成や変換する強力なフレームワークの登場により、様々な画像生成や変換に関する研究が盛んに行い、画像生成技術が大幅に改善され、より高品質の画像を生成することができるようになった。しかし、一般的な GAN のネットワークは生成画像の属性の制御ができなかったので、GAN の構造に条件を付与して生成画像を制御する Conditional GAN(cGAN) [2] を提案することで、生成画像を制御することができるようになった。Image-to-image 変換はソースドメインからターゲットドメインへのマッピングを学習して画像から画像変換を行う手法として、生成画像の属性が制御できる cGAN のネットワークに入力画像を潜在ベクトルにエンコーディングする Encoder を追加して、画像を変換を行う手法である。

GAN を基にした様々な研究には、人の顔画像、数字画像、風景画像または都市景観画像のデータセットが幅広く使用されている。一方、GAN を使用して食品の画像を生成または変換するタスクはかなり少ない。また、「ラーメン」は日本で最も人気

のある食べ物であり、アジア、アメリカ、ヨーロッパなど世界中で人気のある日本料理である。

本研究では、ピクセル単位のラベルが付けられたラーメン画像が含まれているラーメン画像データセットと Image-to-image ネットワークを用いて、ユーザーがスケッチしたマスク画像を基にして各要素の形状を制御したリアルなラーメン画像生成を行う。しかし、生成結果がユーザーが希望するスタイルと違う画像が生成され、生成画像のスタイルが制御できない問題がある。そのため、各要素のスタイル特徴を抽出する Style encoder を追加することで、スタイルを反映したラーメン画像生成を行う。

2. 関 連 研 究

GAN は Generator と Discriminator の 2 つのネットワークから構成され、Discriminator は実画像と合成画像を判別、Generator は Discriminator が実画像と合成された画像を判別できないようによりリアルな画像を生成することを目的にする画像生成のフレームワークである。GAN のネットワークは様々な応用が可能だったので、生成画像の品質を向上する研究及び多様なタスクに GAN を基にする応用研究が盛んに行っている。GAN は正規分布などの潜在変数をサンプリングして画像を生成するが、生成画像を制御ができない。Conditional GAN(cGAN) [2] は GAN の構造に条件を付与して属性を制御する画像生成ができるようになった。

Image-to-image 変換はソースドメインからターゲットドメインへのマッピングを学習することが目的として、cGAN の構造に Encoder を追加することで、画像変換ができるようになった。Pix2pix [3] は Encoder-Decoder 構造を U-Net [4] というネットワーク構造を使用し、互いに対応するペア画像を使用して画像変換を実現した。UNIT [5] と CycleGAN [6] では、ペア画像を学習し使用する Pix2pix とは異なり、互いに対応されない画像を用いて、画像変換を提案した。しかし、複数のドメイン間のマッピングを学習するためには、すべてのドメインに対してモデルが必要になるの問題がある。例えば、k 個のドメインを学習するためには、$k(k-1)$ 個のモデルの学習が必要がある。StarGAN [7] では、1 つのモデルで複数のドメイン間の Image-to-image 変換を提案した。MUNIT [8] と DRIT [9] は画像の各要素のコンテンツとスタイルなどの特徴を分離させ、その特徴を特徴別に制御して、画像変換を行う Disentangled representation による画像変換を提案した。Pix2pixHD [10] は、最大 256×256 の画像サイズまでが合成ができない制約がある Pix2pix を改善して、2048×1024 サイズの高解像度の画像の合成が可能になる。SPADE [11] では、正規化した後に損失される情報を防ぎ、より良い品質の画像生成のために、SPADE(spatially-adaptive normalization) という正規化レイヤーを提案する研究である。また、SEAN [12] は SPADE の手法で領域ごとのスタイルを変換する正規化レイヤーを提案することで、本研究と類似な研究である。

本研究では、実画像と実画像に対応するマスク画像を含めている画像データセットと高品質の Image-to-image 変換が可能である SPADE の手法を参照して研究を行う。

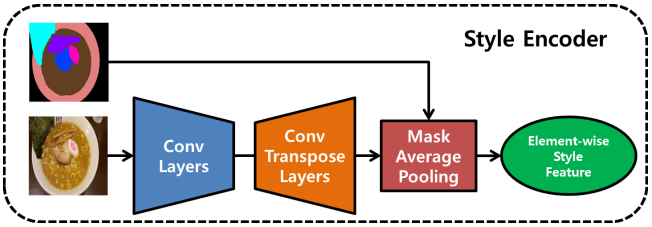

図 1　Style Encoder の構造図。

3. 手　　法

ここではマスク画像から画像生成する Image-to-image 変換のネットワークにスタイル特徴を抽出し、スタイルを考慮した画像生成について説明する。まず、スタイル画像から各要素のスタイルを抽出する Style encoder について説明し、次に Image-to-image 変換のネットワークである SPADE の手法と作成した Style encoder に組み合わせについて説明する。

3.1 Style Encoder

スケッチ画像を基にして生成した画像のスタイルを制御するために Style encoder を作成する。スタイルを抽出する方法としては、図 1 のような流れである。まず、スタイルに使用する画像を Convolution レイヤーと Transposed convolution レイヤーの encoder に入力して、入力画像の特徴マップを取得する。Convolution - Transposed convolution レイヤーの構造は SEAN [12] を参考にした。その後、入力画像に対応するマスク画像のサイズを調整し、特徴マップと組み合わせて Mask Average Pooling を行い、各要素ごとのスタイル特徴を抽出する。

Mask Average Pooling については、まず、スタイル画像から特徴マップ取得し、マスク画像を特徴マップのサイズと同様になるようにマスク画像を調整する。次に、図 2 の流れのように、各要素ごとのスタイル特徴マップとマスク画像の各要素ごとのセマンティックラベルマップを掛け算することで、マスキングされた特徴マップを取得する。その後、マスクピクセル数を持って特徴ごとに平均を計算する。最後に、特徴を足し算することでスタイル特徴を抽出することが可能である。

3.2 Generator と Discriminator

Generator はスタイルを考慮した画像生成をするために、Style encoder から抽出されたスタイル特徴と改良した SPADE [11] の Normalization を用いて Generator の学習を行う。具体的な方法として、Generator と残差ブロックの構造は既存の手法と類似だが、図 3 のように Normalization 構造をスタイル画像とマスクから抽出した要素ごとのスタイルとマスク画像を連結して正規化を行うように改良する。これにより、各要素の領域情報とスタイル情報を Generator に入力することができるので、既存の手法では難しかった各要素ごとのスタイルを制御することが可能になり、各要素のスタイルを考慮した画像生成ができるようになる。

Discriminator は既存手法である SPADE に使用した各レイヤーごとにを Instance normalization 適用し、Spectral normalization を使用する Multi-scale Discriminator を用いてネットワークの学習を行う。

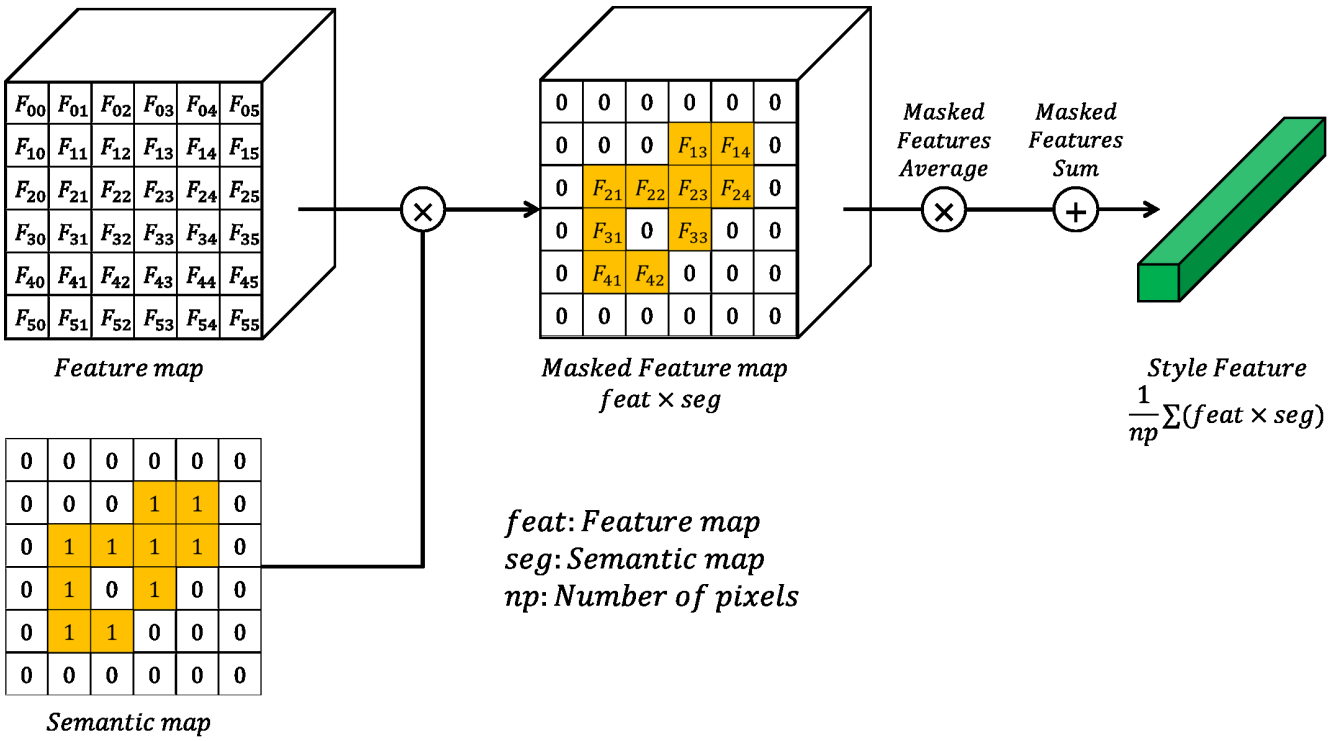

図 2　Mask Average Pooling の流れ図。

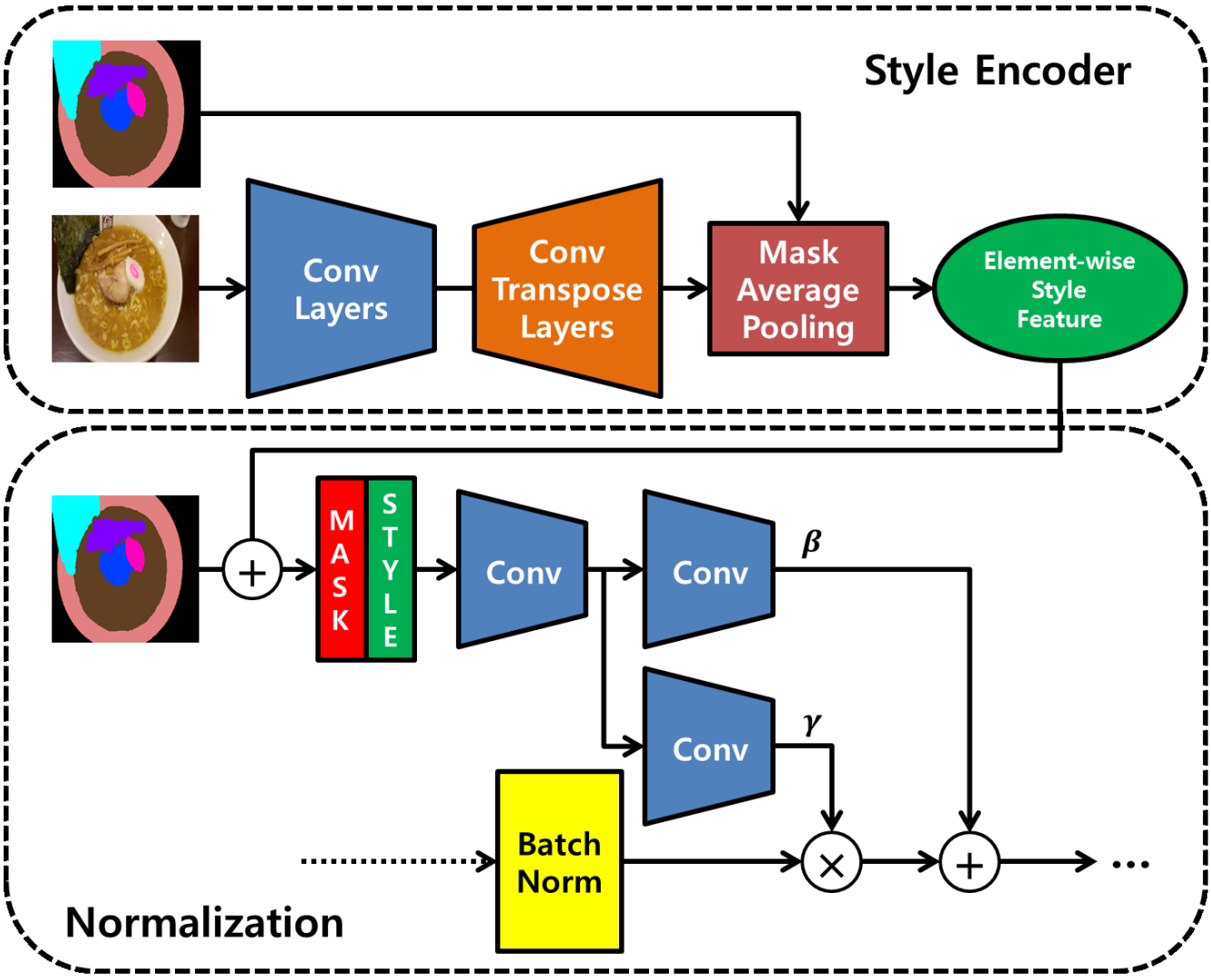

図 3　Style Encoder と Spade Normalization を組み合わせた構造図。

3.3 損失関数

本研究の損失関数は Pix2pixHD [10] と SPADE [11] の 3 つの損失関数に Style encoder を追加して抽出したスタイルを反映するように改良した損失関数を使用した。

(1) Adversarial 損失

Conditional Adversarial 損失は式 1 のミニマックスゲームを通して、実画像の条件付き分布をモデリングする。G は Generator、D_1、D_2 は Discriminator、E は Style encoder である。

$$\mathcal{L}_{adv}(G,D,E) = \min_{G,E}\max_{D_1,D_2}\sum_{k=1,2}\mathcal{L}_{GAN}(G,D_k,E) \tag{1}$$

$\mathcal{L}_{GAN}$ は次の式 2 のような Hinge 損失関数を用いている。その式で x は入力画像、s は x の領域分割したマスク画像である。

$$\begin{aligned}\mathcal{L}_{GAN}(G,D,E) = &-\mathbb{E}[min(0,-1+D_k(s,x))]\\ &-\mathbb{E}[min(0,-1-D_k(s,G(s,E(s,x))))]\end{aligned} \tag{2}$$

(2) Feature matching 損失

Discriminator D_k の i 番目レイヤーの特徴マップを $D_k^{(i)}$ と定義する。また、 T は D_k のすべてのレイヤーの数、N_i は各レイヤーでの要素の数である。

$$\begin{aligned}\mathcal{L}_{FM}(G,D_k,E) =&\mathbb{E}\sum_{i=1}^{T}\frac{1}{N_i}[\|D_k^{(i)}(s,x)\\ &-D_k^{(i)}(s,G(s,E(s,x)))\|_1]\end{aligned} \tag{3}$$

(3) Perceptual 損失

$F^{(i)}$ は VGG ネットワーク [13] の i 番目のレイヤーであり、M_i は VGG ネットワークの要素の数である。

$$\begin{aligned}\mathcal{L}_{perceptual}(G,F,E) =&\mathbb{E}\sum_{i=1}^{N}\frac{1}{M_i}[\|F^{(i)}(x)\\ &-F^{(i)}(G(s,E(s,x)))\|_1]\end{aligned} \tag{4}$$

最後に、その 3 つの損失関数を組み合わせて画像生成の学習を行う。本研究では、$\lambda_1 = \lambda_2 = 10$ として学習を行った。

$$\begin{aligned}\mathcal{L}_{total} = \min_{G,E}\Bigg(&\Big(\max_{D_1,D_2}\sum_{k=1,2}\mathcal{L}_{GAN}(G,D_k,E)\Big)\\ &+\lambda_1\sum_{k=1,2}\mathcal{L}_{FM}(G,D_k,E)+\lambda_2\mathcal{L}_{perceptual}(G,F,E)\Bigg)\end{aligned} \tag{5}$$

4. 実　　験

本実験では、次の 3 つの条件で画像生成の実験を行う。

- 従来手法と画像生成結果の比較
- 1 つのスタイル画像と任意のマスク画像
- 複数のスタイル画像と任意のマスク画像

また、各実験にはラーメン画像のデータセットを用いた実験を行う。

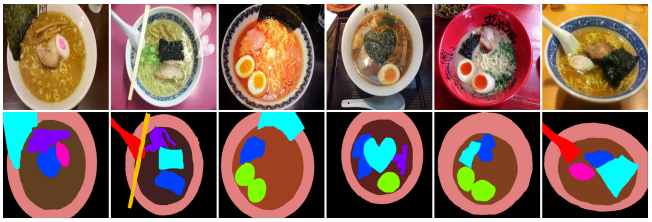

図 4　ラーメン画像のデータセットの例 （上：元画像、下：各要素を領域分割したマスク画像）。

表 1　スタイルを考慮した画像生成結果を FID とユーザー評価を実施した結果

	FID	ユーザー評価
SPADE	**73.19**	24.87%
Ours	78.53	**75.13%**

4.1 データセット

本研究では、555 枚のラーメン画像と各要素ごとにラベルが付けられたマスク画像で構成されているラーメン画像データセット [14] を使用した。また、このデータセットのマスク画像はマスク画像の各要素は 15 クラスにカテゴリで構成している。データセットの例は図 4 である。

本実験では、データセットの中で 500 枚ずつの実画像とマスク画像を用いて各タスクの学習を行い、55 枚ずつの実画像とマスク画像を使用してテストを行う。また、スープカテゴリの明確な確認のために、カテゴリの中で塩スープ、醤油スープ、味噌スープ、豚骨スープ、辛口スープの 5 つのスープカテゴリを 1 つのスープカテゴリに統合して、画像生成を行った時にスープのスタイルが良く反映されたかも確認した。

4.2 スタイルを考慮した画像生成

本実験では、3 つの条件で実験を行う。まず、スタイル画像から抽出したスタイルとスタイル画像に対応するマスク画像を用いた画像生成を行って、従来手法と比較して評価を行う。次に、1 つのスタイル画像から抽出したスタイルと任意のマスク画像を用いた画像生成を行う。最後に複数のスタイル画像から指定して抽出したスタイルと任意のマスク画像を用いて画像生成を行う。

4.2.1 従来手法と画像生成結果の比較

本実験では、1 つのスタイル画像とそのスタイル画像に対応するマスク画像を用いてスタイルを考慮した画像生成を行い、従来手法である SPADE と生成結果の比較を行った。生成結果は図 5 で表せる。結果の図としては、1 列目は入力マスク画像、2 列目は入力スタイル画像、3 列目は生成した結果画像である。評価には、Fréchet Inception Distance(FID) [15] とユーザー評価を用いて各手法を評価を行った。FID を用いた評価では, 生成画像とラーメンデータセットの画像 555 枚を用いて評価を行った。ユーザー評価では、従来手法と提案手法から生成された結果でスタイル画像からのスタイルが良く反映された手法の結果を選択することで評価を行った。評価の結果は表 1 で表せる。

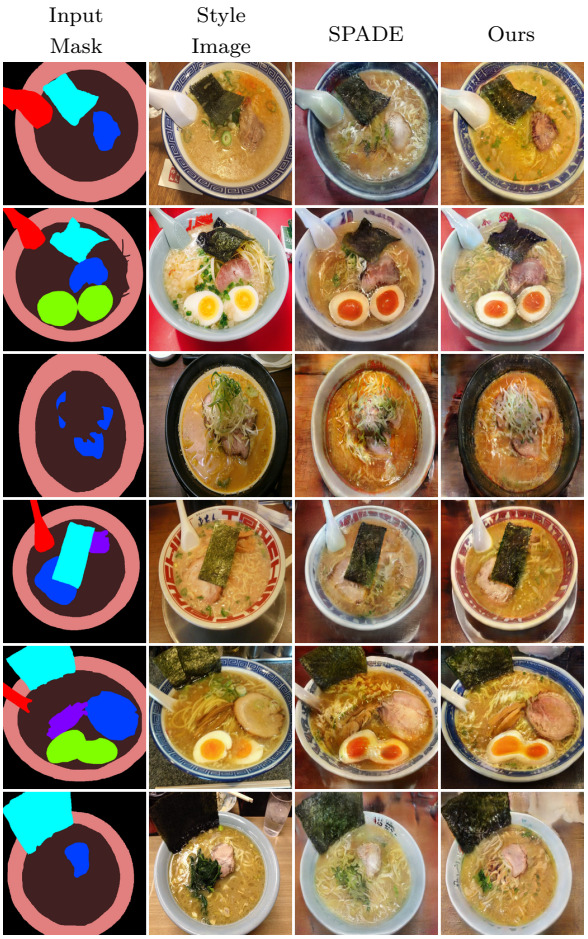

図 5　スタイルを考慮した画像生成結果の比較 （1 列目：入力マスク画像、2 列目：入力スタイル画像、3 列目：SPADE の結果、4 列目：本研究の手法の結果）。

4.2.2　1 つのスタイル画像から抽出したスタイルと任意のマスク画像から画像生成

次に, 特定のスタイル画像からスタイル特徴を抽出し、任意のマスク画像と抽出したスタイルを基にして、スタイルを考慮した画像生成を行う。図 6 は実験結果として、1 行目のスタイル画像からスタイル特徴を抽出したスタイルと 1 列目のマスク画像を用いてスタイルを考慮した画像生成を行った結果である。

4.2.3　複数のスタイル画像から抽出したスタイルと任意のマスク画像から画像生成

次に, 複数のスタイル画像から特定のスタイルを抽出し、マスク画像と抽出したスタイルを用いて画像生成を行った。また、本実験で抽出するスタイルは、結果の明確な比較のために、3 つのカテゴリ要素（背景、器、スープ）に限って、スタイル画像からスタイルを抽出し、スタイルを考慮した画像生成を行う。実験の結果は、図 7 で表せる。結果の図としては、1 行目にスタイル画像と各スタイル画像に抽出したスタイルのカテゴリが書いている。そして、そのスタイルと 1 列目のマスク画像を用いてスタイルを考慮した画像生成の結果である。

5.　考　　察

本実験では、ラーメン画像データセットを用いて、Style encoder を追加してスタイルを抽出し、そのスタイルを考慮した画像生成を試みた結果、従来の手法では難しかったスタイルを考慮した画像の生成ができた。

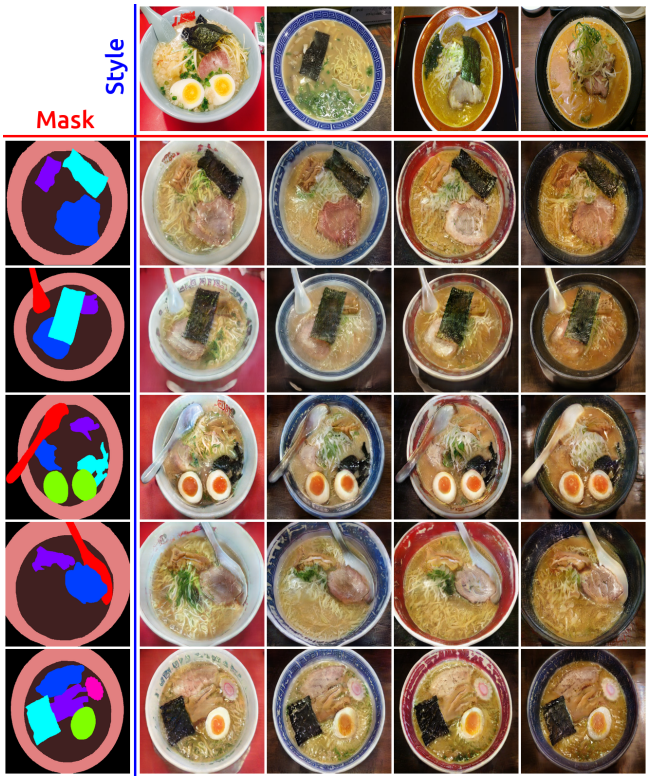

図 6　特定のスタイル画像から抽出したスタイルを考慮した画像生成の結果。

従来手法である SPADE と本研究の手法の結果を比較すると、FID を用いた評価は従来の手法が本研究の手法より良いスコアが出る結果を得られて、従来手法がより多様なラーメン画像の生成が可能であることをわかった。また、従来手法は特定のスタイルを指定して生成することではないので、生成しやすいスタイルで画像生成を行う傾向がある。そのため、FID を用いた評価は従来手法のほうが良い結果になったと考える。しかし、入力したスタイルが良く反映されたかを結果画像とユーザー評価の結果で比較すると、本研究の手法の方が既存手法よりカテゴリ要素のスタイルを良く反映させて画像生成を行ったことが見られる.

また、本研究では、入力マスクとは異なる特定のスタイル画像を用いた画像生成することや複数のスタイル画像から一部のスタイルを抽出し、そのスタイルを適用して画像生成する実験を試みた。この実験により、複数の特定画像から必要なスタイル特徴だけを抽出し、その様々なスタイル特徴の組み合わせを用いた画像生成を実現した.

しかし一方で、スタイルが良く反映されなかった結果も見られた。例えば、図 8 の 1 行目の箸、2 行目の器、3 行目のレンゲのように各要素のスタイルが良く反映されなかった結果の例である。この問題は、学習で使用したデータセットで一部の要素の形状が一貫されない問題または色などのスタイルの多様性が少ない問題があり、全体的なデータの数が少ないので、多様なスタイルの画像を学習できなかったと考えられる。

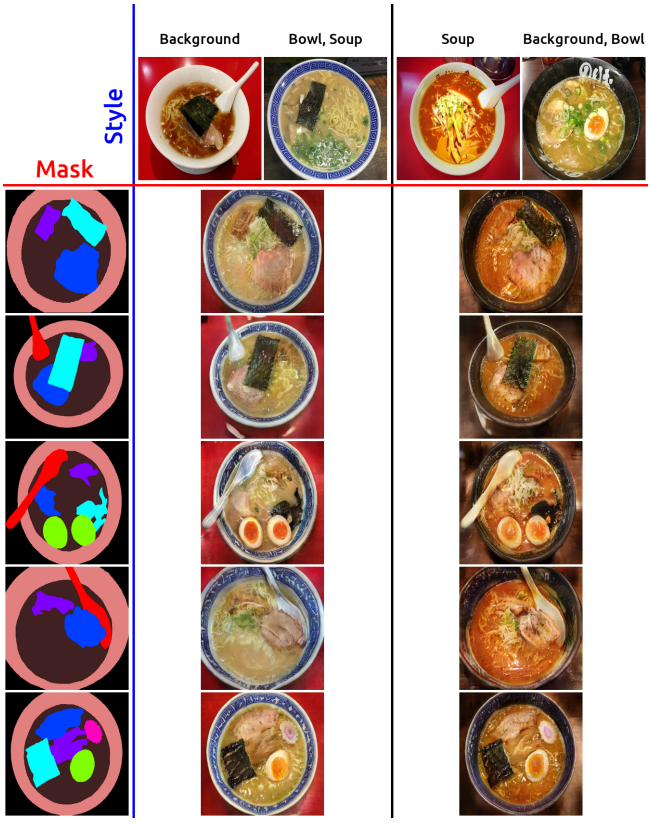

図 7　複数のスタイル画像から抽出したスタイルを考慮した画像生成の結果。

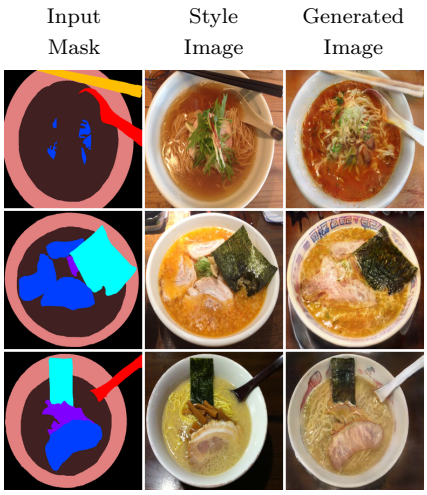

図 8　スタイルが反映されなかった結果の例（1 行目: 箸, 2 行目: 器, 3 行目: レンゲ）。

6. おわりに

本研究では、Image-to-image 変換のネットワークと Style encoder を用いてスタイルを考慮したラーメン画像生成を行った。

従来手法では各要素ごとのスタイルの制御ができないので、入力画像と生成画像の要素の色などのスタイルが異なる問題があった。そのため、入力したスタイル画像のスタイルを考慮する画像生成を行うために、Style encoder を作成し、SPADE を改良することで、スタイルを考慮した画像生成するネットワークを作成した。実験でラーメン画像データセットを用いて、ユーザーが簡単にスケッチした画像と好みに合わせたスタイルを持っているラーメン画像を用いて新しいラーメン画像を生成した。また、複数の画像からそれぞれ希望するスタイル特徴を抽出して、スタイルを反映した画像生成も可能だった。

今後、本研究で生成したラーメン画像をより良い品質で生成できるようにデータセットのデータを増やす予定がある。また、ラーメン画像以外のデータセット（顔、ファッションなど）を学習して画像生成を行う予定である。

謝辞 本研究は JSPS 科研費 17J10261、15H05915、17H01745、19H04929、17H06100 の助成を受けたものです。

文　献

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc.of Advances in Neural Information Processing Systems*, 2014.

[2] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv:1411.1784*, 2014.

[3] P. Isola, J. Y. 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.

[4] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc.of International Conference on Medical image computing and computer-assisted intervention*, 2015.

[5] M. Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Proc.of Advances in Neural Information Processing Systems*, 2017.

[6] 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.

[7] 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 *Proc.of IEEE Computer Vision and Pattern Recognition*, 2018.

[8] X. Huang, M. Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *Proc.of European Conference on Computer Vision*, 2018.

[9] H. Y. Lee, H. Y. Tseng, J. B. Huang, M. Singh, and M. H. Yang. Diverse image-to-image translation via disentangled representations. In *Proc.of European Conference on Computer Vision*, 2018.

[10] T. C. Wang, M. Y. Liu, J. Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proc.of IEEE Computer Vision and Pattern Recognition*, 2018.

[11] T. Park, M. Y. Liu, T. C. Wang, and J. Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proc.of IEEE Computer Vision and Pattern Recognition*, 2019.

[12] P Zhu, R Abdal, Y Qin, and P Wonka. Sean: Image synthesis with semantic region-adaptive normalization. *arXiv preprint arXiv:1911.12861*, 2019.

[13] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc.of International Conference on Learning Representations*, 2015.

[14] J. Cho, W. Shimoda, and K. Yanai. Ramen as you like: Sketch-based food image generation and editing. In *Proc.of ACM International Conference Multimedia*, 2019.

[15] 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. In *Proc.of Advances in Neural Information Processing Systems*, 2017.