

空間的なスタイル転送を利用したアイコン線画彩色

宮内 洸希^{†1} 清雄^{†1} 田原 康之^{†1} 大須賀 昭彦^{†1}

概要：近年, GAN (Generative Adversarial Network) を用いたアイコン自動彩色手法が考案されているが Reference 画像と線画の組み合わせ次第では彩色が失敗することがある。そこで, 本研究では空間的な対応を用いて色を転送する SCFT(Spatially Corresponding Feature Transfer) モジュールと色と構造に分けられた 2 つの識別機を用いた精度の高い彩色手法を提案している。手法の評価は 3 つの指標で行い, 生成された画像の品質を評価する FID スコア, 線画通りに塗られているかを評価する双方向線画ピクセル間距離, 色の類似度を評価する JS-Divergence を用いた。その結果, FID スコア, 双方向線画ピクセル間距離では既存手法よりも優れた評価を得ることができた。

1. はじめに

イラストは小説のキャラクターの挿絵, ノベルゲームの背景画等様々な形で商業的に利用されている。このようなイラストは線画を作る工程と色を塗る工程を経て作られる。これら 2 つの工程は熟練したスキルを持つイラストレーターでなければこなすのが難しいという問題がある。この問題を解決するために深層学習技術を利用したイラストを自動生成する手法が考案されている。PaintsChainer [1] や Style2Paints [2] と言った手法では与えられたアニメ調の人間の線画に対して自動で彩色を行うということが可能になった。

しかし, このような手法はアニメ調の人間の線画に対する結果でありそれ以外の種類の線画に対しては必ずしも良い彩色ができるわけではない。特に人間の線画に比べてマルチモーダルなデータが含まれるアイコンのような画像の線画に対しては学習が難しいので精度の高い彩色が難しい。アイコンの線画に対する彩色は Dual-cGAN [3] や Triple-cGAN [4] のような手法が考案されている。前者は与えられた線画と Reference 画像である他のアイコン画像を入力に線画どおりにアイコン画像の色で彩色するという手法である。しかしながらこの手法は線画とアイコン画像の組み合わせによってははみ出して塗ったり塗り残しが出来たりとあまり良い精度で塗れない時があるという欠点がある。また, Triple-cGAN ではドーナツのアイコンのようにくり抜きの構造を持つ線画に対してくり抜かれた部分は塗らないようにするということが可能になっているが入力として線画とアイコン画像さらにマスク画像の 3 つを要求する

のでユーザーに対しての負担が高くなっているという特徴がある。

現在, アイコンは Web サイト作成やアプリケーション作成の工程において需要が増している要素である。これら製作者はアイコンを作ることに慣れていない人間ばかりではなく線画はかろうじて描けるが色塗りはできないという人は少なくない。これらの人々に対しては深層学習によるアイコン自動彩色技術を用いれば制作の手伝いをする事ができる。しかし, 現在の手法ではアイコン画像の組み合わせによっては精度が低くなってしまいう問題がある。特に, アイコンは Web サイトやアプリケーションの UI の中でも重要な要素であり, 色合いが不揃いで塗り残しやはみ出しがある質が悪いアイコンではユーザーに与える印象も悪くなってしまふ。このため, 線画と Reference 画像の入力から高い精度の彩色ができる手法の開発が望まれている。

2. 関連研究

2.1 GAN

深層学習による画像生成で広く使われる手法に Generative Adversarial Network(GAN) [5] がある。GAN は生成器である Generator と識別器である Discriminator を敵対的に学習させることで学習データに即した生成データを得るようになる手法である。Generator においてはランダムノイズベクトル z を入力として受け取り, 偽物の画像を生成する。対して Discriminator に関してはデータセット中の画像と生成した偽物の画像を入力とし正誤判定, 入力された画像が本物が偽物かを判定する。これらのプロセスを交互に学習させることにより Generator はより本物らしい画像, Discriminator はより高い精度で判別できるようなるため生成データの向

^{†1} 現在, 電気通信大学
Presently with The University of Electro-Communications

上が見込まれる. これらを定式化すると GAN の目的関数 $V(D, G)$ は本物の画像 x とランダムノイズ z と生成器 $G(\cdot)$ と識別機 $D(\cdot)$ を用いて以下のように表すことができる.

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] \quad (1)$$

$$+ \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (2)$$

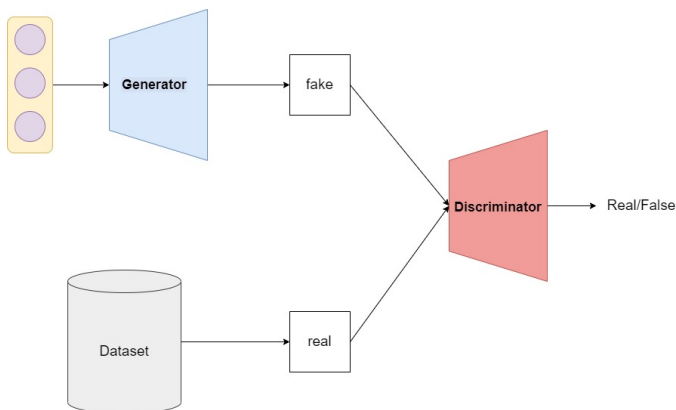


図1 GAN のアーキテクチャ

2.2 Dual-cGAN

Dual-cGAN [3] はアイコン画像と線画を入力として線画をアイコン画像の色の通りに彩色する手法である. アーキテクチャとしてとして線画とアイコン画像のペアを入力し線画通りに彩色されているかを判別する Discriminator, 色が似たアイコン同士を入力に色を判別する Discriminator をもつ.

2.3 Triple-cGAN

Triple-cGAN [4] は線画画像, 線画画像と同じ形のマスク画像, Reference 画像を入力として線画アイコンを生成する手法である. TriplecGAN では浮き輪やドーナツのアイコンのようにくりぬかれて色が無い部分をもつアイコンに対し空白部分に色を塗らずにアイコンを生成出来る. アーキテクチャとしては Dual-cGAN のものに加えてマスク画像の判定をする Discriminator が存在する. しかし, マスク画像が入力に必須になり現状線画からマスク画像を生成できる手法は存在しないためユーザーは線画以外にマスク画像を製作する手間をかける必要がある.

2.4 SIC

SIC [6] は Reference ベースのスケッチ彩色手法である. スケッチ画像の着色は漫画やイラスト作製などのコンテンツ制作において需要があるがスケッチという情報量の少ない画像に着色するのは難しいという問題がある. SIC では

靴やアニメ画像といったアライメントがそろったデータセットに対して SCFT モジュールによる特徴を利用し幾何学的な対応を取ることによって色を転送している. 本手法は様々な種類のスケッチ画像の色付けにおいて既存の手法と比較して定量的および定性的に最も高い評価値をもつ.

3. 提案手法

3.1 モデル構造について

提案するモデルは Encoder-Decoder モデルからなる Generator と構造, 色に関する Discriminator を持つモデルである. モデル図を図 2 に示す. 本モデルの Encoder は線画と Reference 画像から特徴を抽出し SCFT モジュールに送る. SCFT モジュールに入力されたあとは Reference と線画の特徴間の対応の学習を行う. その後, SCFT モジュールを出た特徴は ResBlock [7] に入力し SCFT モジュールからの特徴と ResBlock からの特徴を Decoder に入力して画像を生成する. 生成された画像が 2 つの Discriminator にかかけられて 1 枚の画像ペアに対する学習が終了する.

3.2 SCFT モジュールについて

Spatially Corresponding Feature Transfer モジュール (SCFT モジュール) は SIC [6] で提案されたアーキテクチャである. アーキテクチャの概要図を図 3 に示す. SCFT モジュールの目的は

- 参照画像のどの部分から色を持ってくるか
- スケッチ画像のどの部分に参照画像からの色を移すかの 2 つを学習させることである. まず Reference 画像, 線画の Encoder E_r, E_s から中間出力を含む L 個の活性化マップ (f_1, f_2, \dots, f_L) を生成する. 次に f_L の空間の大きさに合わせてダウンサンプリングを行いチャンネルの次元に合うように連結をして活性化マップ V をつくる. ここで φ は入力活性化マップ $f^l \in \mathbb{R}^{h_l \times w_l \times c_l}$ を $f^{l_p} \in \mathbb{R}^{h_p \times w_p \times c_p}$ の大きさにダウンサンプリングする関数である. また, \otimes はチャンネル単位の連結演算子を表す.

$$V = [\varphi(f^1); \varphi(f^2); \dots; \varphi(f^L)] \quad (3)$$

次に, V を $\tilde{V} = [v_1, v_2, \dots, v_{hw}] \in \mathbb{R}^{d_v \times hw}$ に変形する. ここで, $v_i \in \mathbb{R}^{d_v}$ は与えられた画像の i 番目の領域の特徴表現であり $d_v = \sum_{l=1}^L c_l$ となる. これらを用いて線画用 Encoder の E_s からの特徴と Reference 用 Encoder の E_r の出力から特徴 \tilde{V}_s の v_i^s と \tilde{V}_r の v_j^r をそれぞれ取得する. v_i^s と v_j^r が与えられたとき Attention map $A \in \mathbb{R}^{h_w \times h_w}$ を計算する. その要素 α_{ij} は [8] で提案されたドット積によって計算される. この Attention Map はクエリとキーの類似度を内積の計算することで求めることができる. このとき v_i^s と v_j^r の内積そのまま計算するのではなく $W_q, W_k \in \mathbb{R}^{d_v \times d_v}$ をそれぞれ v_i^s と v_j^r クエリベクトルとキーベクトルに変換してから内積を計算する.

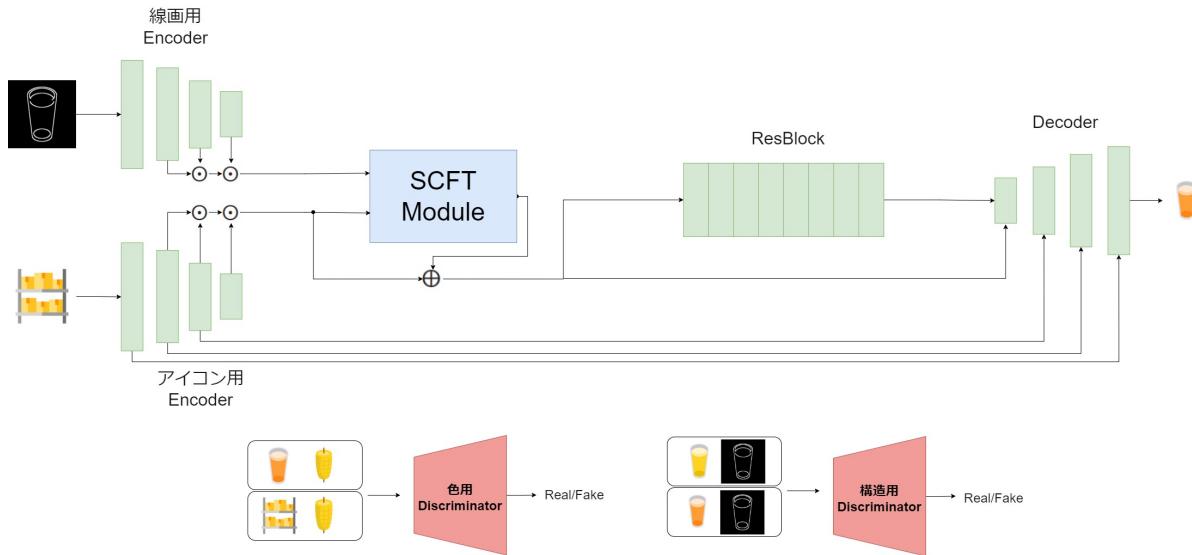


図2 提案モデル

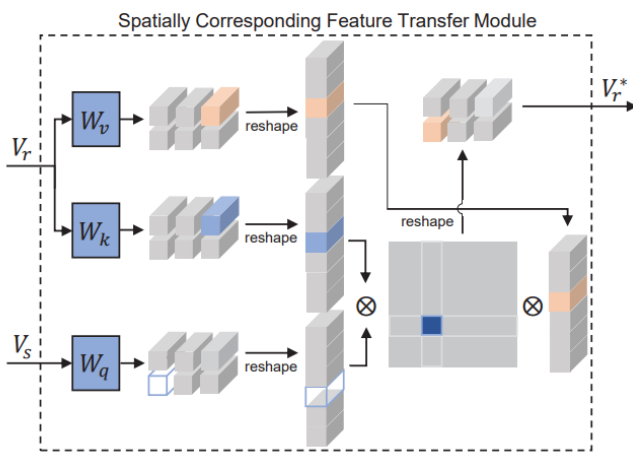


図3 SCFT モジュール ([6] より引用)

$$\alpha_{ij} = \text{softmax}_j \left(\frac{(W_q v_i^s) \cdot (W_k v_j^r)}{\sqrt{d_v}} \right), \quad (4)$$

なお、 $\sqrt{d_v}$ は勾配が小さくなりすぎて学習が進まなくなるのを防ぐためのスケール要素を表す。 α_{ij} は v_i^s が v_j^r からどれだけの情報をもたらすべきかを表す係数である。つまり、この α が大きいほど色 Reference から線画へ色が転送されやすくなる。ここで、位置 i のコンテキストベクトル v_i^s を次のように定義する。

$$v_i^s = \sum_j \alpha_{ij} W_v v_j^r \quad (5)$$

この $W_v \in \mathbb{R}^{d_v \times d_v}$ は Reference 画像の意味的に対応する領域の色特徴のベクトルへの線形変換行列である。最後にスケッチ画像の特徴量 v_i^s にコンテキストベクトル v_i^s を加えて Reference 画像の対応する部分の情報で補強された特徴量ベクトル c_i を形成する。

$$c_i = v_i^s + v_i^s \quad (6)$$

この c_i が Decoder に入力され彩色された画像が生成される。

3.3 Discriminator

Discriminator の構造を図4に示す。構造、色の Discriminator ともに Convolution 層, SpectralNorm 層 [9], Leaky ReLU 層からなる ConvBlock と Convolution 層, Tanh 層からなるブロックで構成される。構造の Discriminator には本物の線画, 本物のアイコン画像を正解の入力にし本物の線画, 生成画像を不正解の入力にしている。また、色の Discriminator では学習データすべてを k-means アルゴリズムを使い色ごとに分け似た色同士のアイコンを正解の入力, 生成画像と Reference 画像のペアを不正解の入力にした。

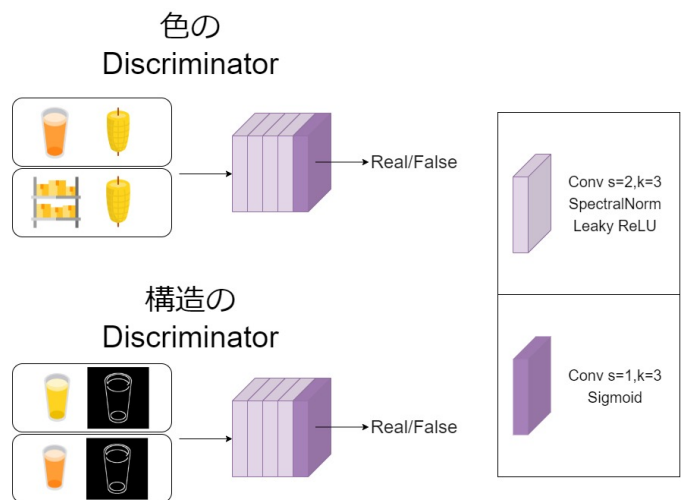


図4 Discriminator のアーキテクチャ

3.4 損失関数について

3.4.1 Triplet Loss

式 5 の Attention Map を計算するために使用されるピクセル単位のクエリとキーベクトルの間の学習のために、Triplet loss を変形した類似性ベースの Triplet loss を用いる。 $S(-, -)$ は scaled dot-product を表しクエリベクトル v_q 、つまり線画の特徴に対し Reference 画像の正負の領域からの特徴である v_k^p と v_k^n との scaled dot-product [8] をマージン γ を保ちながら計算していく。

$$L_{tr} = \max(0, [-S(v_q, v_k^p) + S(v_q, v_k^n) + \gamma]) \quad (7)$$

3.4.2 Adversarial loss

本手法では Discriminator は 2 つあり片方は生成画像が線画通りに似ることを促すための構造の Discriminator である。 I_t, I_s, I_r をそれぞれ本物のアイコンの画像, 本物のアイコンの画像の線画, Reference 画像とする。すると, 生成画像は Generator である $G(\cdot)$ を用いれば $G(I_s, I_r)$ となりこれらを構造の Discriminator である D_s に入力すると損失関数 L_s は次のように書ける。

$$L_s = \mathbb{E}_{I_t, I_s} [\log D_s(I_t, I_s)] \quad (8)$$

$$+ \mathbb{E}_{I_s, I_r} [\log(1 - D_s(G(I_s, I_r), I_s))] \quad (9)$$

2 つある Discriminator のもう一方は生成画像の色の分布が Reference 画像に似ることを促すためのものである。ここで k-means を利用し色を分類する。 $k(\cdot)$ は入力されたアイコンの属する色のクラスを表す。同じ色に分類された色のペアの画像である I_1 と I_2 を正解の入力にすることで生成される偽物画像の色を似せることを行う。

$$L_c = \mathbb{E}_{I_1, I_2} [\log(1 - D_c(G(I_1, I_2), I_r))] \quad (10)$$

$$+ \mathbb{E}_{I_1, I_2} (I_1 = k(I_2)) [\log D_c(I_1, I_2)], \quad (11)$$

3.4.3 Recognition Loss

この Recognition Loss は生成画像の線画が元の線画に近づくことで塗り残しやはみ出しを減少させる目的がある。ここで $C(\cdot)$ は Canny フィルタ [10] とする。生成画像 $G(I_s, I_r)$ に Canny フィルタをかけることにより線画を作成し元の線画 I_s との L1 ノルムを計算することによってより元の線画通りに $G(I_s, I_r)$ が生成されるようにしている。

$$(12)$$

3.4.4 全体の損失関数

全体の損失関数は次の式 13 で表すことができる。

$$\min_G \max_D \mathcal{L}_{total} = \lambda_{tr} \mathcal{L}_{tr} + \lambda_{rec} \mathcal{L}_{rec} \quad (13)$$

$$+ \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s \quad (14)$$

4. 提案手法

4.1 実験

比較対象として, CycleGAN [11], Pix2pix [12], Dual-cGAN [3], SIC [6] の 4 つの手法を比較対象にした。

学習 epoch 数は CycleGAN は 60epoch, Pix2Pix は 200epoch, Dual-cGAN は 2000epoch, SIR は 100epoch, 提案手法を 750 エポックとした。

4.2 Data Augumentation

過学習を防ぐために Data Augumentation を行った。本手法では Reference 画像を上下左右反転する処理を行った。Data Augumentation の概要図を図 5 に示す。

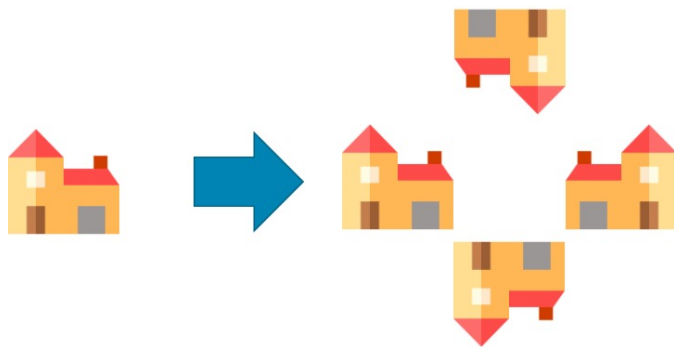


図 5 Data-Augumentation の例

4.3 実験環境

提案手法のネットワークのパラメータ最適化手法として Adam [13] を用いた。また, 提案手法及び比較手法において学習には NVIDIA Tesla V100 SXM2 を 1 台用いた。Loss 関数のパラメータは $\lambda_{tr} = 1, \lambda_{rec} = 3, \lambda_c = 1, \lambda_s = 1$ である。

4.4 ハイパーパラメータ

提案手法と比較手法の Generator, Discriminator の学習率 lr_G, lr_D や Adam の最適化に用いる値 β_1, β_2 は以下の通りである。なお, アーキテクチャとして Discriminator を 2 つもつ提案手法と Dual-cGAN においては Discriminator ごとに学習率の差異はない。

表 1 ハイパーパラメータ

手法	lr_G	lr_D	β_1	β_2
提案手法	0.001	0.002	0.5	0.999
CycleGAN	0.0002	0.0002	0.5	0.999
Pix2pix	0.0002	0.0002	0.5	0.999
Dual-cGAN	0.00005	0.0002	0	0.999
SIC	0.001	0.002	0.5	0.999

4.5 データセット

実験では Training と Test に用いるデータセットが異なる。モデルの Training にはで公開されているフラットスタイルアイコンデータセット [3] を用いた。Test に用いたデータ



図 6 Training に用いたデータセットの例

セットは Flaticon [14] から集めたフラットスタイルアイコンデータセットである。

Training 用のデータセットは 256×256 のアイコン画像が 12675 枚, Test 用のデータセットは 256×256 のアイコン画像が 1239 枚含まれている。

4.6 実験結果

図 7 に提案手法で生成したアイコン画像を示す。図 7 を



図 7 生成結果

見ると生成結果は線画通りに彩色されていることがわかる。また、色に関しては大部分の色は Reference 通りだが一部

Reference がない色が発現しているのがわかる。

4.7 他の手法との比較

図 8 に既存手法と提案手法の比較画像を示す。図 8 を

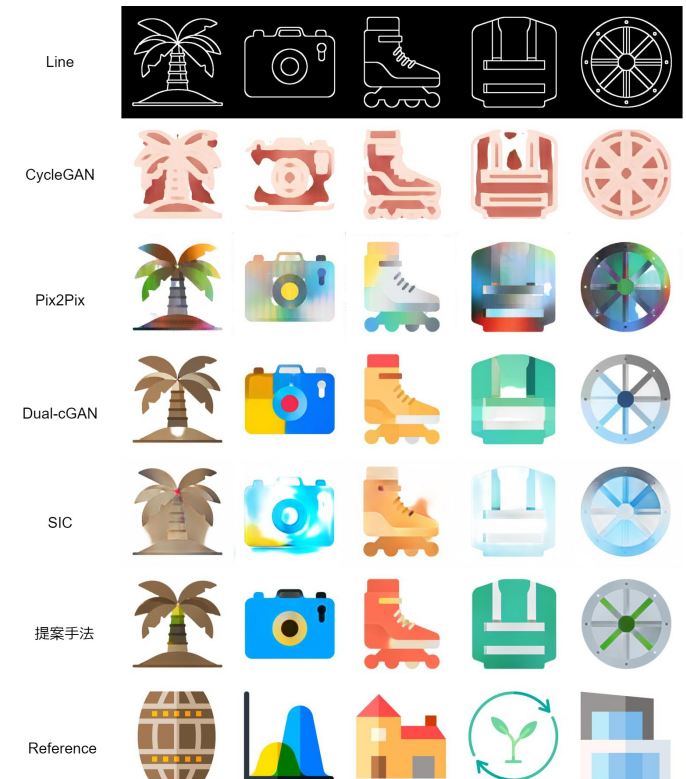


図 8 他の手法との比較

見ると CycleGAN においては線画通りにも、Reference の色通りにも塗れていないことがわかる。Pix2pix においては形は線画通りではあるが詳細な部分のはみ出しや塗り残しがあり、更に Reference の色をほとんど反映していない。Dual-cGAN において色は Reference の色を反映できているが本来線がない部分で色が別れて線が出来上がっており、部分的に塗り残しができる部分があることがわかる。SIC は色は Reference の色が反映されているがのはみ出しや塗り残しが目立つ。提案手法においては一部 Reference がない色が発現しているが線画通りに塗れていると言える。

5. 評価

5.1 FID スコア

5.1.1 FID の説明

FID [15] は Inception V2 のトレーニング済みモデルを用いて、Reference 画像と生成画像の特徴ベクトルを抽出し、その特徴間の距離を測ったものである。これにより生成モデルの評価が可能になる。

5.1.2 FID の評価結果

客観評価では画像分布間の近さを表す指標を算出する FID のスコアの結果を以下に記す。表から提案手法が既存

表 2 FID スコア

手法	評価値
提案手法	37.91
CycleGAN	102.59
Pix2pix	86.16
Dual-cGAN	40.27
SIC	84.33

手法のスコアを上回っていることがわかる。よって、提案手法は既存手法よりも生成画像の品質が高いと言える。

5.2 JS-Divergence

5.2.1 JS-Divergence の説明

JS-Divergence は確率分布の距離を測るための指標である。同じ確率分布の距離を測るものとして KL-Divergence があるが KL-Divergence は非対称であるので JS-Divergence を採用する。本評価実験では生成画像と Reference 画像から 3D ヒストグラムを抽出したものを JS-Divergence で分布の距離を測ることで色の類似度を算出する。これを評価用のすべての画像に行い平均を求めたものを評価値とした。

5.2.2 JS-Divergence の評価結果

外れ値の除去のために下位 5 % と上位 5 % のデータは取り除いた。評価値は表 3 の通りである。表から提案手法は既

表 3 JS-Divergence の評価値

手法	評価値
提案手法	0.508
CycleGAN	0.679
Pix2pix	0.608
Dual-cGAN	0.433
SIC	0.365

存手法よりも Reference 画像通りに彩色できていないことが言える。

5.3 双方向線画ピクセル間距離

5.3.1 双方向線画ピクセル間距離の説明

生成画像が線画通りに塗られているかを測定する必要がある。ここでは双方向線画ピクセル間距離はピクセル同士の距離を測るものである。生成画像の線画と彩色される前の線画から線画部分のピクセルの座標を取得する。生成画像の線画ピクセルの座標全てに関して塗られる前の線画画像のピクセル座標の中で最も近い点とのユークリッド距離 D_A を算出してその和を求める。その後、塗られる前の線画画像のピクセル座標全てに関して生成画像の線画ピクセルの座標の中で最も近い点との距離 D_B を算出してその和を求める。 $(D_A + D_B)/2$ を双方向線画ピクセル間距離として求める。これを評価用のすべての画像に行い平均を求めたものを評価値とした。この評価手法の概要図を図 9 に示す。

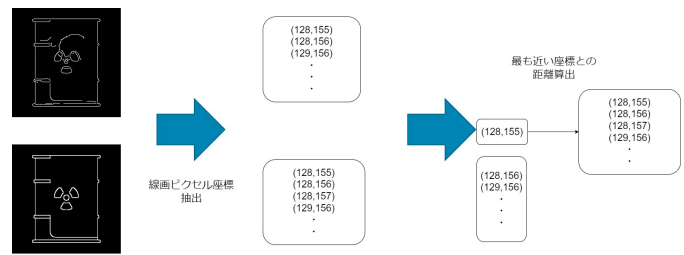


図 9 双方向線画ピクセル間距離の算出

5.3.2 双方向線画ピクセル間距離の評価結果

外れ値の除去のために下位 5 % と上位 5 % のデータは取り除いた。評価値は表 4 の通りである。表から評価値におい

表 4 双方向線画ピクセル間距離の評価値

手法	評価値
提案手法	0.257
CycleGAN	5.720
Pix2pix	0.680
Dual-cGAN	0.344
SIC	1.561

て提案手法が既存手法を上回っているのがわかる。ここから提案手法は既存手法よりも線画通りに彩色できていることが言える。

6. 考察

双方向線画ピクセル間距離の評価においては提案手法が既存手法よりも低い数値となっている。これは、はみ出しや塗り残しが平均的に起きていないということを表している。提案手法は既存手法より線画通りに塗れていることがわかる。提案手法の Generator の部分の SCFT モジュール、Triplet Loss、線画の L1 ロスが影響を与えたと考えられる。特に SCFT モジュールと Triplet Loss ではアイコンの色をどの線画の部分に移すかを学習するため精度の向上に貢献したのではないかと考える。

対して、JS-Divergence による色の類似度の評価では提案手法は既存手法より高い数値になっている。これは、提案手法が既存手法より Reference 画像の色で線画を塗れていないということを表している。例として図 10 に示すとおり Reference 画像にない色が発現している生成画像が見られている。この原因としては SCFT モジュールや Triplet Loss によって Reference 画像の詳細な部分の色が転送されないようになったのではないかと考える。これにより、Reference 画像の大部分を占める色のみが転送され残りの部分は Reference 画像にない色で彩色されてしまったと考える。このため色の類似度を測る JS-Divergence の値が悪くなったと考える。JS-Divergence の値を改善するためにはより強力に色を似せる Loss 関数を導入することや、他の Loss 関数のパラメータを調整する必要がある。

FID スコアは提案手法が最もよい結果であったがこれは



図 10 Reference がない色が出ている部分の例

画像間の特徴の分布を比較しているためであると考えられる。提案手法は JS-Divergence が既存手法と比べて良い結果ではないが双方向線画ピクセル間距離の値が良いのでより本物らしい画像が生成できていたため FID スコアが高かったと考えられる。

7. おわりに

7.1 まとめ

本論文では、アイコンの線画画像に対して別のアイコンを Reference 画像として彩色を行う手法が提案されている。本論文で提案された手法では幾何学的な対応を行う SCFT モジュールを持つ Generator と構造、色それぞれを判別する 2 つの Discriminator によるアーキテクチャを用いている。このアーキテクチャを用いて学習を行いモデルを作製し、画像生成実験を行ったところ FID スコアによる評価と双方向線画ピクセル間距離による評価で既存手法と比較して最も高い評価が得られた。

7.2 今後の展望

本手法では Reference 画像の色を自動で線画画像に転送することを行っているが、実用的には Reference 画像の欲しい色の部分を任意の線画の部分に転送できればよりユーザーが使いやすくなることが考えられる。今後の課題としては、Reference 画像と線画画像にヒントを与えて彩色を行う手法を開発する必要があると考える。そのためには、ヒントが付与されたデータセットの作製、また効率的に色を転移することができるネットワークを考案が望まれる。今後は前述の方法についての検証を行い既存手法と比較し評価したいと考える。

8. 謝辞

本研究は JSPS 科研費 JP18H03229, JP18H03340,

JP18K19835, JP19H04113, JP19K12107, JP21H03496 の助成を受けたものです。

参考文献

- [1] Taizan Yonetsuji. Paintschainer. https://petalica-paint.pixiv.dev/index_ja.html, 2017.
- [2] Lvmin Zhang, Chengze Li, Edgar Simo-Serra, Yi Ji, Tien-Tsin Wong, and Chunping Liu. User-guided line art flat filling with split filling mechanism. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [3] Tsai-Ho Sun, Chien-Hsun Lai, Sai-Keung Wong, and Yu-Shuen Wang. Adversarial colorization of icons based on contour and color conditions. *MM '19*, pp. 683–691, 2019.
- [4] Qin-Ru Han, Wen-Zhe Zhu, and Qing Zhu. Icon colorization based on triple conditional generative adversarial networks. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pp. 391–394, 2020.
- [5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, DavidWarde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems (NIPS)*, 2014.
- [6] Junsoo Lee, Eungyeup Kim, Yunsung Lee, Dongjun Kim, Jaehyuk Chang, and Jaegul Choo. Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS'17*, p. 6000–6010. Curran Associates Inc., 2017.
- [9] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *In International Conference on Learning Representations (ICLR)*, 2018.
- [10] John Canny. A computational approach to edge detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1986.
- [11] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.
- [12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *In International Conference on Learning Representations (ICLR)*, 2015.
- [14] Flaticon. <https://www.flaticon.com/>. (Accessed on 09/10/2021).
- [15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Neural Information Processing Systems (NIPS)*, 2017.