Vol.2017-MPS-112 No.16
2017/2/27

DCGAN を用いたイラスト画像生成の一手法

山下澄奈† 石川由羽† 髙田雅美† 城和貴†

概要：本稿では，DCGAN を用いてイラスト画像の自動生成を行う．日本におけるキャラクタービジネスは以前から盛んであり市場規模は増加し続けている．最近では，企業や地方自治体の独自のキャラクターが宣伝活動を行う例も多い．また，SNS のプロフィール写真を独自のキャラクターのイラスト画像にする者も多く，独自のキャラクターを生成し使用する需要が増加している．しかし，知識や技術を持たない者がイラスト画像を 1 から作成することは困難である．そこで，本研究では DCGAN を用いてイラスト画像の生成を行う．また，生成されたイラスト画像を別のニューラルネットワークに入力し変換を試みる．最初に生成された画像と処理を加えた画像の比較をし，考察を行う．

キーワード：DCGAN，画像生成，Deep Learning

A method to create illustrate images using DCGAN

SUMINA YAMASHITA† YU ISHIKAWA†
MASAMI TAKATA† KAZUKI JOE†

1. はじめに

日本のキャラクタービジネスの起源は，1963 年に由来する[1]．当時，明治製菓が鉄腕アトムのキャラクターのシールをマーブルチョコのおまけとして販売したところ爆発的な人気となった．1997 年には，たまごっちやポケモンのキャラクター商品が予想をはるかに上回る売り上げをみせ，社会現象として取り上げられた．このようにキャラクタービジネスは，以前から盛んであり，様々な人気のあるキャラクターが存在する．例えば，ウォルトディズニー社のミッキー[2], 株式会社サンリオのキティ[3]などが代表的な例である．また，企業や地方自治体が独自のキャラクターを生み出しそのキャラクターを用い商品や企業，地方自治体の宣伝活動を行っている例が多く存在する．主に，熊本県の宣伝活動を行っているくまもん[4]は国内のメディアで多数取り上げられるなど絶大な人気を誇っている．さらに，Twitter や Google+などといった SNS（Social Networking Service）サイトやブログの普及により個人のプロフィール写真をキャラクターに設定する者も多く存在する．それらのプロフィール写真は，既存のキャラクターを用いるだけでなく個人で考え出したキャラクターを使用する者もいる．以上より，独自のキャラクターを生成し使用する需要が増加していると考えられる．

企業や地方自治体が，新たに独自のキャラクターを生成したり，SNS のプロフィール写真を独自のキャラクターに設定したりするには，本人が生成するか他社に依頼する必要がある．しかし，他者に作成を依頼する場合，価値観や金銭面の問題からトラブルに発展している例も存在する．こうした問題は，独自にキャラクターを生成することで解決できる．しかし，専門の知識や技術を持たない者が 1 からキャラクターを考え生み出すのは困難である．

キャラクター画像を独自に生成するサービスは，ちゃんりおメーカー[5]やキャラメイクファクトリー-きせかえ Flash-[6]など，複数存在する．これらは，目や髪などキャラクターを生成するためのパーツが複数用意されておりそれらを組み合わせることで生成する．この作業は，ユーザ自らが手動で行う必要がある．この場合，選択されるパーツが偏り似たような特徴を持つキャラクターが多く生成される可能性がある．このような問題を解決するために，多様性を持った画像の生成を行う必要がある．

本稿では Deep Convolutional Generative Adversarial Networks（DCGAN）[7]を用いてキャラクターのイラスト画像生成を行う．DCGAN では，画像を生成するニューラルネットワークに乱数を入力することで画像の生成を行う．そのため，個人が画像生成のためのパーツ選択をする必要はない．また，乱数の値によって生成される画像が異なるため多様な画像が生成可能であると考えられる．

以下，2 章ではイラスト画像生成の既存研究について述べる．3 章では，DCGAN を用いたイラスト画像の自動生成の手法について述べる．4 章では，3 章で述べた手法を用いてイラスト画像の自動生成を行う．

2. 関連研究

イラスト画像の生成を行っている研究を 2 つ紹介する．第 1 章で述べたキャラクター画像生成サービスでは，パーツの選択はユーザ自らが手動で行っているため，似た特徴を持つ画像が多く生成される可能性がある．これらの研究では，イラスト画像のパーツの組み合わせ方法や，パーツの変形処理を加えることにより多様性を持たせている．

1 つ目は，Hotogi らの感性語と対話型遺伝的アルゴリズム（interactive Genetic Algorithm: iGA）を用いたキャラクタ

† 奈良女子大学
Nara Women's University

ーの自動生成である[8]．iGA では，遺伝的アルゴリズム（Genetic Algorithm: GA）の評価部分を機械ではなく人間の主観により行う．これにより，評価を行う人間の感性，嗜好が反映されるという特徴がある．この研究では，2258 種類のキャラクターから人気のあるキャラクターの上位 100 位と人気のないキャラクターの下位 100 位を選択する．次に選択されたキャラクターから，人気の出るキャラクターの特徴を抽出する.人気の出るキャラクターの特徴として，足が短い，黒く丸い目など 30 種類の特徴が抽出された．イラスト画像生成の手法は，まず，ユーザがイラスト画像のイメージを決定づけるための感性語を選択する．例えば，cute や amiable などである．次に，イラスト画像を生成するための目や輪郭などのパーツを自動で選択し，合計 8 つのイラスト画像を生成する．生成されたイラスト画像に対し，前述の抽出された特徴に基づきランク付けを行い，4 つのイラスト画像を選ぶ．そして，選択された 4 つのイラスト画像に対しユーザが評価を行う．ユーザの評価が行われた後に，ルーレット選択とランダム交叉，突然変異を行う．パーツの構成から，選択・交叉・突然変異までの処理を一定世代繰り返し最終的なイラスト画像の決定を行う．

2 つ目は，小松らによるモーフィングによる似顔絵生成である[9]．この研究は，実際の人間の実写画像から似顔絵画像の生成を行っている．ここでは，実写画像をデフォルメし,手書き風画像に変換した画像を似顔絵と称している．従来の似顔絵生成ではあらかじめ用意された目や口などのパーツから，ユーザ自らが自身の特徴と似ていると思うものを手動で選択し，それらを合成している．しかし，パーツの数は有限であるため，ユーザの特徴に似ているパーツが存在しない場合，ユーザの特徴を捉えた似顔絵が生成できない問題点がある.この問題を解決する手法として，実写画像の特徴を適切に表現するパーツイラストを合成し，それらを組み合わせて似顔絵を生成する手法を提案している．まず，実写画像から眉や口などの複数の各パーツの代表点を取得し，取得した代表点に基づきパーツごとの画像に分割する．代表点の取得は顔検出用の API を使用している．次に，曲線入力とスライダ操作により特徴該当度を算出する．特徴該当度とは，目が丸い眉が細いなど設定された基準に,それぞれのパーツが該当する割合を示す数値である．曲線入力は，パーツの実写画像に対しユーザが直接曲線を手描き入力し,曲線の形状から各特徴点の該当度を求める．そして，これらをモーフィング時の重み付に使用し，実画像の特徴に近いパーツのイラストを合成する．最後に，パーツのイラストを輪郭のイラストの上に配置することによって似顔絵の生成が完了する．

iGA による生成では,交叉や突然変異を行っているため，多様性が生まれ，人間では考え付きにくい優良な画像が生成される可能性がある．また，モーフィングによる似顔絵自動生成では，パーツの選択後に，実画像に近づけるためモーフィングを行っており多様性が生まれる．しかし，以上の研究は，どちらも画像を生成するためのパーツが用意されており，それらを自動的に組み合わせることにより画像の生成を行っている．そのため，生成可能な画像のバリエーションが少なくなるという問題点が生じる．この問題は，パーツを増加させることによって解決できると考えられるが，大量に用意するのは困難である．

3. イラスト生成の提案手法

3.1 全体の流れ

従来のイラスト生成手法では，イラストを生成するためのパーツの数が有限であるため，生成されるイラストのバリエーションが少なくなる可能性がある．そこで，本稿では画像生成技術である DCGAN と呼ばれるアルゴリズムを使用する．DCGAN では入力する乱数の値により生成される画像が異なる．そのため，多様性が生まれ，この問題を解決できると考えられる．

DCGAN を用いた画像の生成の例を複数示す．DCGAN の提案者である Alec らは,乱数からベッドルームの画像を生成している[7]．また，女性のイラスト画像生成を試みた例もある[10]．これらは，数十万枚の画像を基に生成される．また，ベッドルームの画像なら直方体の物体，女性のイラスト画像なら目や口など，ある程度共通した特徴がある．しかし，個人で，そのような共通したデータを大量に用意するのは困難である．そのため，本稿では共通した特徴を持たない少量の画像データからのイラスト画像生成を目標とする．しかし，共通の特徴を持たない少量の画像データから画像を生成することは困難であると考えられる．そこで本稿では,DCGAN を用いてイラスト画像を生成後，生成されたイラスト画像に対して変換処理を行うニューラルネットワークを適用することで，人間が見て自然だと感じる画像への変換を目指す．

以下，提案手法について説明する．図 1 に全体の流れを示す.

手順 1.　DCGAN を用いたイラスト画像の生成
手順 2.　画像のベクトルの復元
手順 3.　画像の変換処理を行うニューラルネットワーク構築

手順 1 では，DCGAN を用いた画像の生成を行う．Generator と呼ばれるニューラルネットワークで画像の生成を行う．Discriminator と呼ばれるニューラルネットワークで，入力された画像が，生成画像であるかデータセットの画像であるかの識別を行う．手順 2 では，データセットの画像を生成するベクトルを再現するため，画像をベクトルに復元するニューラルネットワークを生成する．このニューラルネットワークを，Restorer とする．Restorer は，乱

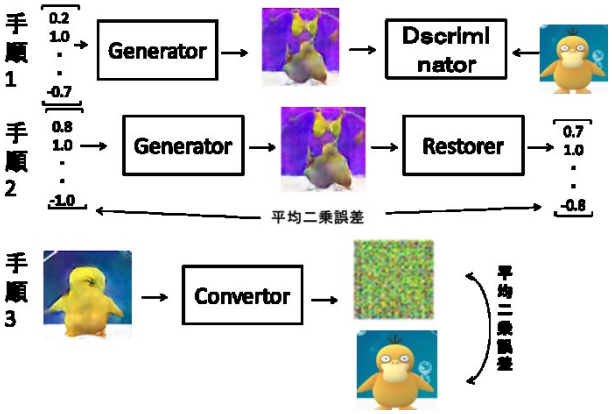

図 1:全体の流れ

数によって生成されたベクトルと画像から復元したベクトルの誤差を逆伝播することで学習が行われる. 手順 3 では, Generator によって生成された画像を, 人間が見て自然だと感じる画像への変換を行うため, 画像の変換処理を行うニューラルネットワークの生成を行う. このニューラルネットワークを Convertor とする. Convertor は, 元画像と教師画像の誤差を伝播することで学習が行われる.

本稿で用いるイラスト画像データは, スマートフォン向けゲームアプリケーションのスクリーンショット画像を用いる[11][12]. これらを用いる理由として, ゲームに出現するキャラクターの種類が多いことから, 様々な特徴をもったキャラクターの画像を使用できるためである[13]. 使用する画像はキャラクターの正面画像および側面画像, 背面画像を使用する. 本研究で使用するデータは 3 チャネル縦 96 ピクセル横 96 ピクセルの 2385 枚の PNG 形式の画像である. 図 2 は, 本稿で用いる画像データの一例である. 図 2 の左は, 正面画像, 中央は側面画像, 右は背面画像である.

3.2 DCGAN を用いたイラスト画像生成

最初に, DCGAN を用いた画像の生成を行う. 画像の生成を行う生成器と画像の識別を行う識別器を学習させることによって, 画像の自動生成を行う Generative Adversarial Network（GAN）と呼ばれるアルゴリズムがある[14]. GAN には画像の生成を行う Generator と画像の識別を行う Discriminator の 2 つのニューラルネットワークが存在する. まず, Generator はR次元の乱数ベクトルから画像の生成を行う. 次に Discriminator は, 画像が Generator で生成された画像であるか, データセット由来の画像であるかを識別する. Generator は Discriminator にデータセット由来の画像だと識別させるような画像を生成するように学習を行い, Discriminator は識別する画像が生成された画像かデータセット由来の画像か正しく識別できるように学習を行う.

GAN にバッチ正規化を適用し, Generator に畳み込み層を使用し, Discriminator で逆畳み込み層を用いたアルゴリズムを DCGAN という. DCGAN における Generator の入力はR次元の乱数ベクトル, 出力は画像である. また,

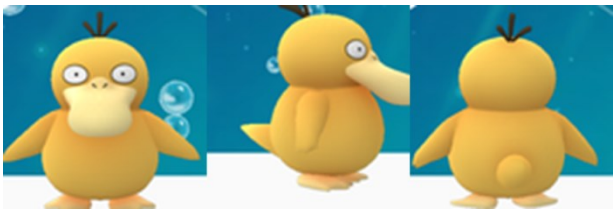

図 2:データセットの画像
正面画像（左）側面画像（中央）背面画像（右）

Discriminator の入力は画像で, 出力は入力画像がデータセットの画像である確率, 生成された画像である確率である. DCGAN では以下の式の最適化を行う. 式(1)の価値関数Vでは, D についての最大であり, かつ G についての最小となるよう D, G を求める. D は, 入力画像が本物である確率を出力する. G は, 画像を生成する. 式(2)(3)のは, それぞれソフトマックスエントロピー関数Eである. $p_{data(x)}$は真のデータの分布でありXは学習データの集合である. $p_{z(z)}$は入力ノイズであり, Z は個々のノイズである. Aは, Discriminator が入力画像を正しく判別できた時に大きくなる. Bの$\log\left(1-D(G(Z))\right)$は, Generator が Discriminator をだますような画像が生成されると大きくなる.

$$min_G max_D V(D,G) = A + B \quad (1)$$

$$A = \mathrm{E}_{X\sim p_{data}(X)}(\log[D(X)]) \quad (2)$$

$$B = \mathrm{E}_{Z\sim p_{Z(Z)}}\left[\log\left(1-D(G(Z))\right)\right] \quad (3)$$

図 3, 図 4 はそれぞれ Generator と Discriminator の概要図である. ch, h, wはそれぞれ出力データのチャンネル数, 高さ, 横幅である. kx, kyはそれぞれカーネルサイズの幅と高さである. 本稿では, 計算速度の向上のために活性化関数として Exponential Liner Units[15]を設定している.

3.3 画像のベクトルの復元

次に, 画像のベクトル化の復元を述べる. 前述の通り, Generator では乱数ベクトルから画像の生成を行っている. そのため, これらのベクトルの組み合わせによっては, データセットに存在するような画像が生成できると考えられる. そこで, 本節では, データセットの画像を生成するベクトルを再現する. Generator では, 画像を乱数で生成するため生成画像のベクトルは判別できるが, データセットの画像のベクトルは分からない. そのため, 入力画像を再現するベクトルを推定するためのニューラルネットワーク Restorer を生成する. Restorer の生成の手順を以下に示す. 図 5 は Restorer の概要図である.

手順2-1. 入力画像の生成
手順2-2. 入力画像を 100 次元のベクトルに復元
手順2-3. 誤差の計算

まず, 手順 2-1 では 3.2 節で学習した Generator を使用し

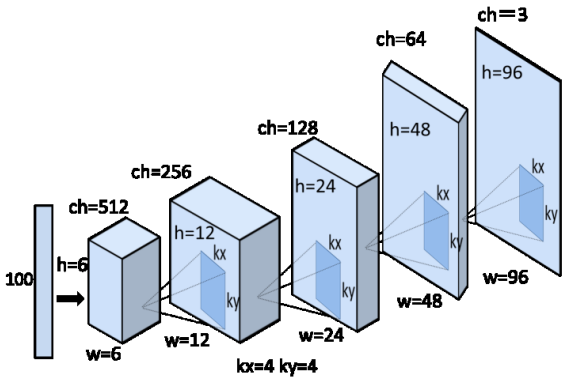

図 3:Generator の概要図

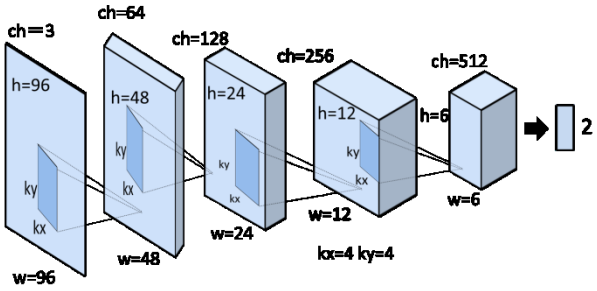

図 4:Dscriminator の概要図

100 次元のベクトルから画像を生成する. 100 次元のベクトルは一様乱数によって生成される．手順 2-2 では，生成された画像を入力画像とし，入力画像の畳み込み演算を繰り返すことによって, 100 次元のベクトルを生成する. 次に，手順 3 では手順 2-1 のベクトルと手順 2-2 で生成されたベクトルの平均二乗誤差を求める．この平均二乗誤差を最小化するように学習を進める．具体的には，以下の式を最適化する．y_iは画像を生成するときに使用するベクトル，t_iは画像から復元されたベクトルである．また，nはバッチサイズである．本稿では，バッチサイズは 100 である．

$$\mathrm{argmin}\frac{1}{n}\sum_{i=1}^{n}\|y_i - t_i\|^2 \qquad (4)$$

これにより，Generator によって生成された画像であれば，復元されたベクトルからでも入力画像に似ている画像を生成することができる． Restorer は，畳み込み層から成る．

3.4 画像の変換処理を行うニューラルネットワーク構築

図 6 の左図は，データセットの画像である．図 6 の右図は Restorer を用い，データセットにある画像を 100 次元のベクトルに復元し，そのベクトルを Generator に入力することで画像の生成を行った例である．形状は似ているが細部まで描けていないことが分かる．これは，Generator の学習が不完全であること，Restorer の学習では Generator によって生成された画像のみ使用して学習しているため，生成画像以外には対応しきれていないことが原因である．そこで，データセットの画像を利用して生成された画像を，該当する画像に変換する．この変換ができれば，データセットの画像を利用して生成した画像でない場合も，画像の変換が可能であると考えられる．以下は，画像の変換を行う Convertor の生成手順である．Convertor は畳み込み層と逆畳み込み層から成る．図 7 に Convertor の概要図を示す.

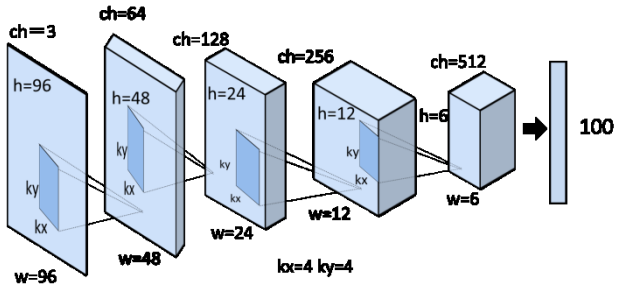

図 5:Restorer の概要図

図 6:データセットの画像（左）Restorer に入力後 Generator に入力した画像（右）

手順3-1. データセットの画像のベクトルの復元
手順3-2. 復元されたベクトルから画像の生成
手順3-3. 生成画像の変換
手順3-4. 誤差の計算

手順 3-1 では 3.3 節の Restorer にデータセットの画像を入力し，100 次元のベクトルを得る．次に，手順 3-2 では手順 3-1 で得たベクトルを入力とし 3.2 節で生成された Generator を用いて画像の生成を行う．手順 3-3 では，手順 3-2 で得られた画像を入力とし, 画像の変換を行う. まず, 入力された画像に畳み込み演算を行い，入力画像の特徴量を圧縮する．次に圧縮された特徴量に，逆畳み込み演算を繰り返すことで変換画像の生成を行う．手順 3-4 では教師画像と手順 3-3 で生成された変換画像の誤差の計算を行う. 教師画像はデータセットの画像である．ここで誤差は，変換画像と教師画像の個々のピクセルに対する平均二乗誤差とする．具体的には，以下の式を最適化する．aは，画像の縦ピクセル数，bは，横のピクセル数である．$s(\mathrm{i},\mathrm{j})$は入力画像の各ピクセル, $k(i,j)$は教師画像の各ピクセルである.

$$\mathrm{argmin}\frac{1}{n}\left(\frac{1}{h\times w}\sum_{i=1}^{a}\sum_{j=1}^{b}\{s(i,j)-k(i,j)\}^2\right) \qquad (5)$$

式(5)で，得られた平均二乗誤差を逆伝播させ Convertor の学習を行う.

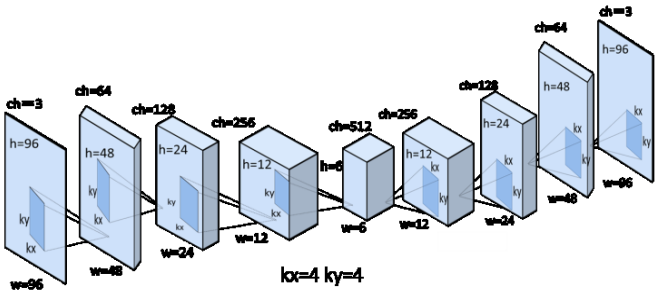

図 7:Convertor の概要図

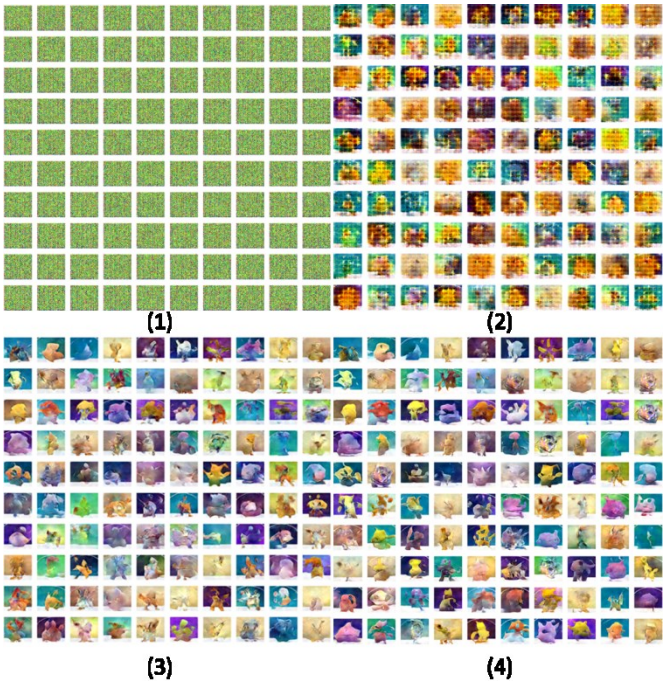

図 8:DCGAN を用いて生成された画像
学習開始直後（1）と 2 エポック後（2）
7 エポック後（3）と 11 エポック後（4）

4. 結果と考察

本稿では，第 3 章で述べた方法を使用しイラスト画像の生成を行う．また，DCGAN を用いてイラスト画像を生成した結果と生成した画像を Convertor に入力した結果を比較する．

4.1 生成された画像

図 8 は，DCGAN を用いて生成した画像 100 枚である．(1)は，学習開始直後，(2)は学習開始から 2 エポック後の画像，(3)は 7 エポック後の画像，(4)は学習開始から 11 エポック後の画像である．

図 9 の上の画像は Generator を用いて生成された画像，図 10 の下の画像は学習開始から 12 エポック後の Convertor に生成画像を入力した結果である．カーネルのサイズは（4,4）である．また，図 11 は Convertor の最終層以外のカーネルのサイズを（6,6）に変化させた結果である．最終層では，画像サイズを(96,96)に合わせるためにカーネルサイズを(8,8)に設定している．上の画像は Generator を用いて生

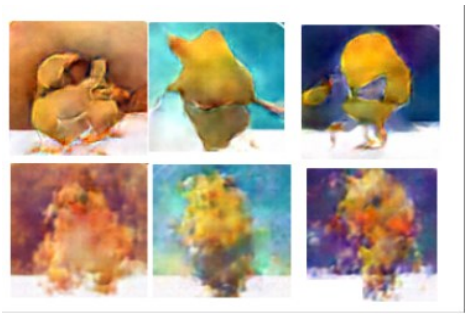

図 9::Generator で生成された画像（上）
カーネルサイズ(4,4)の Convertor で変換した画

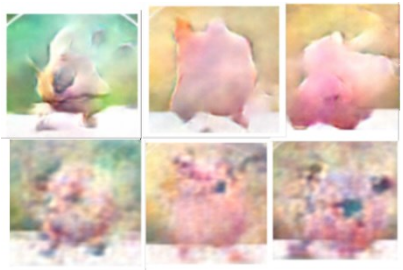

図 10: Generator で生成された画像（上）
カーネルサイズ(4,4)の Convertor で変換した画像

図 11:図 10 と近い印象を持つデータセットの画像
成された画像，下の画像は学習開始から 16 エポック後の Convertor に生成画像を入力した結果である．

4.2 考察

図 8 から，学習開始直後は，ノイズだらけである画像が，2 エポック後は薄くではあるが形状が現れている．しかし，画像は荒く格子状の物が現れている．まだ，学ぶ習は十分行われていないことが分かる．7 エポック目は，はっきりと形状が現れていることが分かる．また，多様な形状の画像が現れている．図 8 の，画像の各上 5 行は，それぞれ同じ位置の画像を同様の乱数から生成している．(2)と(3)を比較すると形状は明確に現れ，使用される色のバリエーション増加していることが分かる．しかし，形状のみ描かれており細部まで描かれていないこと，あまり自然とは言えない形状の画像が生成されていることが分かる．(3)の画像と(4)の画像では，形状に大きな変化はみられない．そのため，7 エポック目での学習が限界であると考えられる．

図 10 から，カーネルサイズ（4,4）の Convertor を適用した結果，わずかではあるが，イラスト画像の形状が変化しているのが変わる．しかし，全体的に画像が荒くなっている．

左下の画像は最も顕著に形状や色について画像の変換が行われている．中央上の画像は，同系色で描かれている．しかし，中央下の画像は下半分の色が変化しているのが分かる．また，右下の画像は，右上の画像にはない赤い丸が

見られる．これは，右上の画像にある隙間を補完するかのように描かれている．図 10 から，カーネルサイズ（6,6）の Convertor を適用した結果，画像の変換が行われていることが分かる．この場合も画像は荒くなっている．左上の画像は，三角形のような形状であるのに対し，左下の画像は形状が丸く変換していることが分かる．中央下の画像と右下の画像はデータセットの画像に近い画像に変換されおり，どちらも形状の変化がみられる．また，形状だけではなく体の内部に青い円の目のような物や黒い物体が現れているのが分かる．変換された画像に近いデータセットの画像を図 12 に示す．図 11 と図 12 を比較した結果，データセットの画像に近い画像に変換されていることが分かる．

このような，データセットに似た印象を持つ画像への変換がみられる理由として，Generator で生成された画像とデータセットの画像を入力とし，Restorer と Generator を使用し生成された画像が，類似しているためだと考えられる．しかし，生成された画像と，上述した方法を使用し生成される画像が類似していない場合，顕著な画像の変換がされていない．これは，Convertor が過学習をおこしている可能性が考えられる．

Convertor の目的は，データセットに存在する画像を復元することではないが，画像の変換が行われていることは分かる．これより，Convertor によって画像の変換が可能なことが分かる．しかし，現状では荒い画像が生成される，細部まで描ききれていない問題点が生じている．そのため，今後，より人間が自然だと感じる画像への変換を行う必要がある．解決方法として，背景の削除，更なるカーネルサイズの変更や畳み込み層，逆畳み込み層の層数の変更，更なる関数の適応などが考えられる．過学習を防ぐ方法としては，ドロップアウト関数の適用が考えられる．また，多段階の GAN[16]を使用するなどして精度の向上を目指す．

5. まとめ

本稿では，DCGAN を用いてイラスト画像の生成を行い，生成した画像の変換を行うための一手法を提案した．事前に自らがイラスト画像の基となる目や口などのパーツ画像を用意しそれらを組み合わせることによりイラスト画像の生成を行っている研究が存在する．この場合，パーツの数は有限であるため，バリエーションが少なくなる問題点が挙げられる．DCGAN を用いて画像を生成する場合，入力をR次元のベクトルとするため 1 度学習を行えば多様な画像が生成可能である． DCGAN を用いた画像生成を行っている研究は複数存在する．この場合，共通の特徴をもった画像を数十万枚以上用意している．しかし，共通の特徴をもつデータを大量に用意するのは困難である．本稿で用いたデータも，鳥のような見た目の特徴をもつキャラクターの画像や魚のような特徴をもつキャラクターの画像など様々な特徴のキャラクターの画像を使用している．このような画像を少量使用しイラスト画像の生成を行う場合，人間の目から見て自然だと感じるイラスト画像の生成は困難である．そのため，画像を変換させるニューラルネットワークを適用した．そのニューラルネットワークを適用した結果，変換された画像が荒くなる問題が生じたが，わずかではあるが画像の変換が行われていることが分かった．また，データセットの画像に近づくように変換している画像もあるため，本稿で提案した手法による画像の変換は有効であると考えられる．前述した問題に対応するには，背景の削除，更なるカーネルサイズの変更や畳み込み層，逆畳み込み層の層数の変更，更なる関数の適応などが考えられる．過学習を防ぐ方法としては，ドロップアウト関数の適用が考えられる．また，多段階の GAN を使用するなどして精度の向上を目指すことによって人間の目から見て自然だと感じるイラスト画像の生成を目標とする．

参考文献

[1] 日本におけるキャラクター・ライセンス・ビジネスに関する考察，Japan Marketing Academy Conference Proceedings vol.5（2016）
[2] A WEB PAGE，ウォルト・ディズニー・ジャパン，http://www.disney.co.jp/corporate.html(accessed 2017-01-29)
[3] A WEB PAGE，株式会社サンリオ公式サイト，http://www.sanrio.co.jp/ (accessed 2017-01-29).
[4] A WEB PAFE，くまもんオフィシャルサイト，http://kumamon-official.jp/(accessed 2017-1-29)
[5] A WEB PAGE，ちゃんりおメーカー，https://chanrio.com/(accessed 2017-01-24)
[6] A WEB PAGE キャラメイクファクトリー-きせかえ Flash-，http://www15.atpages.jp/kisekae99/kisekae.html(accessed 2017-1-24)
[7] Alec Radford, Luke Metz, Soumith Chintala: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks arXiv preprint arXiv:1511.063434
[8] Maho HOTOGI，Masafumi HAGIWARA:Analyses of Local Mascot Characters and Proposal Automatic Character Creation System Using Affective WordsInternational Journal of Affective Engineering Vol. 14 (2015) No. 4 p. 299-307
[9] 小松璃子，伊藤貴之，パーツ単位のモーフィングによる似顔絵生成，芸術科学会論文誌 Vol. 14, No. 5, pp. 180-187
[10] A WEB PAGE, http://qiita.com/mattya/ (accessed 2016-12-01).
[11] A WEB PAGE，Pokémon GO 公式ページ，http://www.pokemongo.jp/ (accessed 2015-12-08).
[12] A WEB PAGE，http://bohemia.hatenablog.com (accessed 2016-12-01).
[13] A WEB PAGE，ポケットモンスターオフィシャルサイト，http://www.pokemon.co.jp/(accessed 2017-01-29)
[14] Goodfellow, I., Pouget-Abadie, J.,Mirza, M., Xu, B., Warde-Farley, D., Ozair, S.,Courville, A., and Bengio, Y.: Generative adversarialnets, in Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
[15] Djork-Arné Clevert, Thomas Unterthiner, Sepp Hochreiter Fast and Accurate Deep Network Learning by Exponential Linear Units arXiv preprint arXiv1511.07289
[16] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, Dimitris Metaxas，StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks，arXiv preprint arXiv:1612.03242