Context-aware GAN による画像自動生成

中村　玄貴† 　馬　　強†

† 京都大学情報学研究科　〒 606–8501 京都府京都市左京区吉田本町 36-1
E-mail: †nakamura-kenki@db.soc.i.kyoto-u.ac.jp, ††qiang@i.kyoto-u.ac.jp

あらまし　敵対的生成ネットワーク (GAN) を用いてテキストから画像を自動生成する研究が盛んに行なわれている．既存研究の多くは，一つの入力テキストからそれに対応する画像生成を行っているため，生成される画像が不自然であったり，関連性の強いテキストを入力しても生成された画像間の関連がまったくなかったりする場合がある．地域や商品などのプロモーションに画像を用いたい場面には別のテキストや写真があり，入力テキストが表現する内容以外に，これらの関連テキストや写真（本研究ではコンテキストと呼ぶ）を考慮すべきである．そこで，本研究では，テキスト情報に加えてコンテキスト画像を入力とすることで，テキストに表現されないコンテキストを反映した画像生成を行う Context-aware GAN を提案する．

キーワード　GAN，自動生成，コンテキスト

1. はじめに

ニューラルネットワークを用いた，画像とテキストの相互生成の研究は盛んに行なわれている [1] [2] [3] [4] [5]．Zhou らは画像に適切なタグを付与するため，画像とタグのペアを学習データとして用意してニューラルネットワークに学習させて，入力画像のタグを生成する手法を提案している [1]．またこの手法を応用し，画像を複数領域に分割して，それらの領域に含まれる物体を検出し，それらを整理して人間の理解可能な文章に仕上げことで画像の説明文を自動生成する研究も行われている [2]．

Ian らは画像生成のフレームワークとして GAN(敵対的生成ネットワーク) を提案している [3]．これは現在の画像生成の主要な手法であり，これを基にした多くの発展的な手法が提案されている．例えば Radford らは GAN に畳み込みニューラルネットワークを組み合わせ，多層化することで鮮明な画像生成を行うモデルを提案している [4]．

従来の GAN は入力された乱数に対し，学習を行ったデータセットに似た画像をランダムに生成する．例えば，花の画像を学習した GAN はランダムな花の画像を自動生成する．しかし花には色や形といった特徴がある．黄色い花の画像を生成したい時，あらゆる花の画像を学習した GAN から求める黄色い花が生成されるとは限らない．

そこで，Reed らは GAN を拡張して入力テキストの対応する画像を自動生成するモデルを提案している [5]．画像とそれに対応したテキストのペアを学習に用いている．上の例でいえば，花の特徴を文章で入力することでその文章に従った画像を取得することができる．

しかし一つの画像からのみ生成されたような画像は未だ実用的とは言えない．Reed らの提案している従来の手法 [5] を用いて一つのテキストから画像を生成することは可能であるが，前後のテキスト・画像を考慮していないため，前後の画像やテキストと全く無関係の画像を生成してしまう可能性が高い．

SNS へ旅行の思い出を掲載しようとした際に適切な写真を撮っていなかった場合，生成して補いたい場面があったとする．その写真は一つのテキストの内容からのみ生成されるべきではなく，前後の写真やテキストの内容を踏まえたものが望ましい．例えば，入力テキストが「夜景を見た」のみであっても，誰と見たか，どこで見たかといった情報が存在しそれらの情報を反映できるればより実用的な画像を生成することが可能になる．

我々の先行研究 [6] [7] では VisualStorytelling [8] で提供されているアルバムを用いて，テキストからだけでなく，生成対象画像の前後の画像を考慮した自動画像生成手法を提案している．しかし，この手法では，画像が鮮明に生成できない問題とデータセットのアルバムに含まれる画像数が少なく，また画像にテキストが不適切であるため，アルバム数を増やすと学習が収束しなくなるといった問題がある．

そこで，本研究では，使用するデータセットを VisualStorytelling [8] から VQA [12] に変更することでデータセットに関する問題の解決を図る．また．Reed らの提案する text-conditional GAN [5] を拡張し，テキスト情報に加えてコンテキスト画像から抽出される情報を入力とすることで，テキストに表現されないコンテキストを反映した画像生成を行う Context-aware GAN を提案する．

（1） 生成画像が鮮明でないことへの対策．

StackGAN [11] で提案されている，サイズの小さな画像を生成し，それを入力とすることでより大きく鮮明な画像を生成する手法を用いる．

（2） 学習データセットの関係で，コンテキスト情報を考慮した画像生成がうまくできないことへの対策．

VQA [12] のデータセットを用いる．

Context-aware GAN の最終的な目標は，入力されたテキストとコンテキスト画像から共通部分と異なった部分を抽出し，テキストに沿って，コンテキストを踏まえた画像生成を行うことである．

本研究では，生成する画像の内容を前景に当たる人物などのエンティティと，背景にあたる場所（室内，室外，公園，海辺

など）に分けて考える．生成する画像が，コンテキストのエンティティ（前景）または背景（場所）との共通を持たせてコンテキストに沿った画像生成を行う．

本論文の構成は以下である．2. 節で関連研究について紹介する．3. 節で提案手法の詳細を述べる．5. 節でまとめと今後の課題について述べる．

2. 関 連 研 究

2.1 Generative Adversarial Nets(GAN)

Ian らの提案している GAN は画像生成を行うニューラルネットワークのフレームワークである [3]．Generator と呼ばれる画像生成を行うニューラルネットワークと Discriminator と呼ばれる画像の判別を行うニューラルネットワークの二種類から構成されている．

Discriminator は入力された画像が本物画像であるか生成画像であるかを判別できるように学習させていく．出力は入力画像が本物である確率を [0,1] で出力する．Generator は Discriminetor を欺くことのできるような画像を生成できるように生成する．

Discriminator を学習させたのち，Generator によって生成された画像が Discriminator を騙せるように Generator のパラメータを学習していく．生成画像 G に対して Discriminator が本物の画像と判別する確率が高くなるよう Generator にフィードバックさせる．また Generator を学習させたのち，それの出力画像と本物の画像を判別できるように Discriminator を学習させていく．このように交互に Generator と Discriminator を学習させていくことで本物のような画像を Generator が生成することを可能にしている．

Radford らは GAN に畳み込みニューラルネットーワーク (CNN) を用いて多層化することにより，それよりもさらにリアルな画像生成を可能にする DCGAN を提案している [4].

Martin らは discriminator の損失関数に Earth-Mover(EM) 距離を用いた Wasserstein GAN を提案している [9]．これは出力結果に偏りが生じるモードコラプスという問題を解決し，安定的な学習と学習の収束の観測を可能にしている．

2.2 Text-conditional GAN

Reed らは従来の GAN を拡張し，テキストを入力としてそれに対応する画像を生成するモデルを提案している [5]．この手法の中で入力となるテキストを condition としている．Generator の入力ベクトルを全て乱数にするのではなく，そこにテキスト情報をベクトル化したものを用いることでその内容に沿った画像の自動生成を行う．また，Discriminator は GAN では入力される画像が生成画像か本物の画像かを見分けるものであったが，ここでは入力画像が本物の画像かどうかに加え，テキスト情報と一致しているかどうかを判断する．入力画像が本物であり，かつテキストがその内容を表すときのみ真と判断するように学習させていく．そのため，学習データにはテキスト (キャプション) と画像がセットになったデータセットを用いる必要がある．

さらに Generator は乱数 z とテキストのベクトルを入力して画像生成を行うものであるが，テキストベクトルと画像を入力することで z を逆算することもできる．これにより求められた z は入力に用いた画像のスタイルを保持したまま，テキスト情報を踏まえた画像を出力できるものになる．

単一のテキストから画像を生成することが可能である一方で，その前後にあるコンテキストの情報を反映することができない．本研究では Reed らのモデルを拡張し，コンテキストに存在するオブジェクト情報も反映した画像を生成する．

2.3 VGAN(GAN for Video)

Vondrick らはコマ送りになった画像の列となっているデータセットから動画を生成する GAN の拡張モデルを提案している [10]．GAN では入力された z に対して直接演算を行って，画像を生成するが，この手法では Generator 内で z を分岐させ，Foreground 部と Background 部のニューラルネットワークにそれぞれ代入して前景と背景の画像をそれぞれ生成し，合成することで動画生成を行う．Background 部は全ての動画のコマで共通部分となる背景を生成する．つまり，背景は Background 部の出力は一枚の画像である．Foreground 部はオブジェクトがコマ送りに動いている複数の画像を前景として生成する．Mask は Foreground のオブジェクトを反転させた画像となっていて，Background を合成するときに背景からオブジェクト部分を抜き取り合成できるようにするために用いられる．

画像列を生成するという点では，画像列からコンテキストを考慮した画像生成を行う Context-aware GAN と同じである．しかし，VGAN は Foreground 部はコマ送りで少しずつ動き，Background 部は完全に固定されているが Context-aware GAN が考慮する画像列は映っているオブジェクトが同じであってもその場面は著しく変化しているため，この手法で生成を行うことはできない．

またこの手法では Multi-Scale Architecture が用いられている．Multi-Scale Architecture は小さな画像をまず生成し，それから再帰的に大きな画像生成を行っていくことで局所的特徴も捉えた画像を生成しようとするものである．

$$Y_k = G_k(X, Y_{k-1})(k > 1) \tag{1}$$

上式のように様々なサイズの画像を出力する Generator を複数構築し，その出力を再帰的に次の Generator の入力としている．

2.4 StackGAN

Zhang らは従来の画像生成手法に比べ，より鮮明で大きな画像を自動生成する StackGAN を提案している [11]．Text-conditional GAN はテキストに沿った 64 × 64 のサイズの画像を生成可能である．これよりも大きな画像生成を行うと，鮮明な画像が生成できなくなってしまう．そのため，この手法では画像生成を Stage1, Stage2 という 2 種類の GAN を用いる段階に分けている．

Stage1 では Text-conditional GAN と同様に画像と，それに対応するテキストからなるデータセットを学習し，入力テキストから 64 × 64 のサイズの画像生成を行うモデルを作成する．

Stage2 では入力テキストと，そのテキストを Stage1 に入力することで生成した 64 × 64 のサイズの画像を入力として GAN を学習させる．入力画像を入力テキストに沿ったままアップサンプリングを行うことで，Stage1 よりもよりテキスト情報を鮮明に反映した 256 × 256 のサイズの画像を自動生成する．

2.5 CycleGAN

Zhu らは GAN を拡張し，2 グループの画像の特徴を学習して一方のグループからもう一方のグループへ画像変換を行う手法を提案している [14]．例えば，馬とシマウマの画像変換を行いたい時，馬とシマウマだけが入れ替わっており，姿勢や位置，背景が全く同じような画像セットを大量に用意することができれば，このような画像変換を行うことは難しくない．しかしそのようなデータセットを用意することは不可能であるため，GAN を用いて Unpair な 2 画像グループから特徴を学び，画像変換を行うようにしている．

2.6 StarGAN

CycleGAN [14] を用いることで任意の 2 グループ間で画像変換を行うことができるが，グループの数が n 個に増えた際，${}_nC_2$ の数だけ CycleGAN を用意する必要がある．そこで Choi らは CycleGAN に conditional な情報を入力できるよう改良することで一つの GAN で任意のグループ間の画像変換が可能になる StarGAN を提案している [15].

2.7 SeqGAN

Yu らは GAN を自然言語生成に応用する手法を提案している [16]．画像生成のモデルを学習させる際，出力値は連続値となるため学習を行えるが，言語では例えば単語に振られた ID を生成しなければいけないため，離散値となりそのままでは学習を行うことができない．そこで，強化学習・MCMC の手法を導入し，マルコフ木探索によって生成された文章が自然になるよう，Generator はある地点における次の単語の生成確率を出力し，報酬との尤度を最大化するよう学習を行う．

3. 提 案 手 法

本研究では，コンテキストを考慮した画像生成を行えるようにする手法を提案する．Context-aware GAN へは入力として画像に対応したテキストとコンテキスト画像を入力として与え，それらを考慮した画像を生成する．

前述したように，本研究では，コンテキストを前景（エンティティ）と背景（場面）に分けて考える．生成画像とコンテキスト画像が共通の前景または背景の要素を持たせるように画像を生成する．

図 1 に示されているように，Conext-aware GAN の処理が大きく以下の二つの部分に分けられる．

（1） コンテキストネットワーク：入力テキストとコンテキスト画像の差分を求め，コンテキスト画像から生成画像に取り込めるべき関連性を保つ部分（場面またはエンティティ）を明らかにして画像生成部に渡す．

（2） 画像生成ネットワーク：入力テキストとコンテキスト処理で抽出されたコンテキストを入力とし，画像を生成する．

画像生成ネットワークでは,「インドア」か「アウトドア」という場面（背景）を反映した上で，映っている人物などのエンティティ（前景）を認識できるように画像生成を行う必要がある．つまり，これは場面という画像全体で表現する大域的な情報と，画像の一部で表現される局所的な情報の両方を鮮明に出力する．そこで，本研究では，Vondrick らが提案している手法 [10] の中で用いられている Multi-Scale Architecture を用いて画像生成を数段階に分割し，小さな画像生成から，それを用いてより大きな画像生成を行っていくことで，局所情報と大域情報の両方を踏まえた画像生成が行えるようにしている．

3.1 データセット

学習データセットとして VQA を用いる [12]．画像例は図 2 に示す．このデータセットは画像認識のコンペティション用にクラウドワーカーによって作成されたものである．VQA では画像作成時にどのように作成されたかの情報が残されているため，オブジェクトの種類と位置を取得することができる．

オブジェクトは複数あるが，場面としては「インドア」か「アウトドア」かの 2 つしかない．クラウドワーカーはそれぞれの場面について用意された UI を用いて画像を作っていく．それぞれの場面でしか用いることのできないオブジェクトも存在している．場面が「インドア」か「アウトドア」かの情報も提供されているため，関係のあるコンテキスト画像の収集・グルーピングも簡単に行うことができる．

3.2 コンテキストネットワーク

データセットに付与されている場面の情報から同じ場面の画像をすべてグルーピングする．今回のデータセットは場面は「インドア」と「アウトドア」の 2 つであり，また映っている人物の種類が 20 種類あるため，その数だけ分類を行うことができる．画像数は 20,000 枚あり，1 グループあたりの画像も十分なため，我々の先行研究 [6] [7] で問題となっていた，コンテキスト画像が少ないために，コンテキストを学習できない問題は解決している．これらを適切に識別できる識別器を学習させる．識別器のモデルは図 3 に示す．出力は softmax 関数により，one-hot vector となるようにする．

入力画像には前景・背景が必ず存在しているが，入力テキストには前景または背景の詳細の描写が必ずあるとは限らない．

- 背景について

入力テキストに背景の情報が描写されていればそれが背景であり，描写されていなければコンテキスト画像から抽出された背景画像を用いる．

- 前景について

入力テキストにエンティティの情報が描写されていたとしても,「女性」という単語ではどのような女性かはわからない．女性に当てはまるエンティティを入力画像から探して，コンテキスト画像に含まれる特定の女性を出力できるようにする．

以上から差分を考慮して，背景が「インドア」か「アウトドア」か，映っている人物がどのような女性か男性かな生成画像に反映させたい情報を one-hot vector の形式にして入力とする．

3.2.1 画像の背景・前景識別器

識別器の学習式は以下の通り．

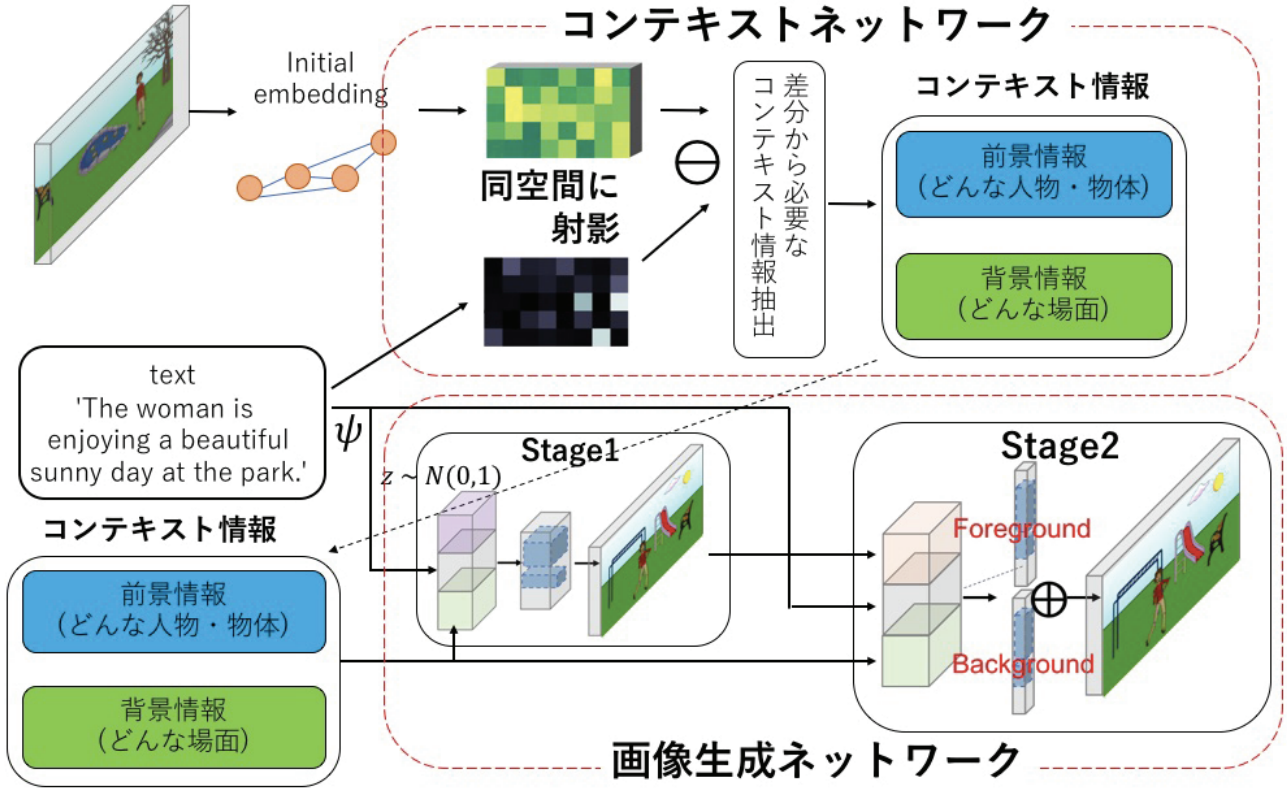

図 1　Context-aware GAN の一連の流れ.

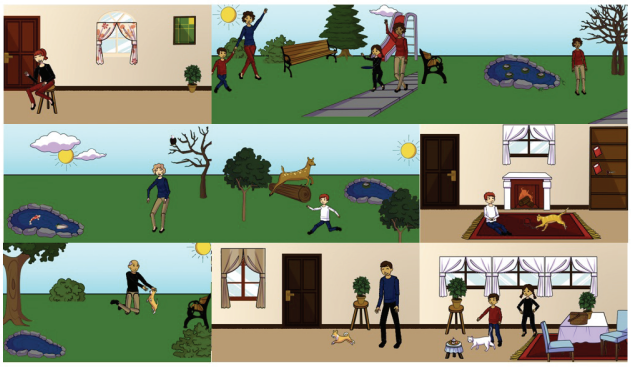

図 2　データセットの例

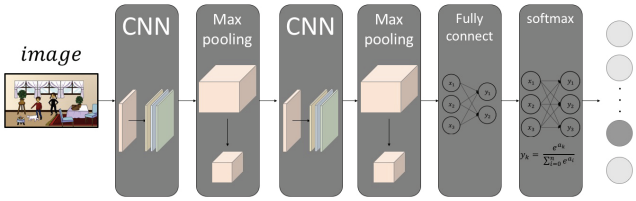

図 3　識　別　器

$$L = -\sum_{batchsize} label * log[R(I)] \tag{2}$$

ここで R(I) は画像を入力した Recognizer の出力値であり，「インドア」の確率と「アウトドア」の確率，または，人物の種類がどれであるかそれぞれの確率を出力する．softmax 関数により合計が 1 になっている．label は正解ラベルであり R(I) と同じ形式で，正解の方の要素が 1, それ以外が 0 になっているベクトルである．これの乗算結果を最大化 (マイナスをつけた損失関数を最小化) することで，R(I) の出力が label に近づく．つまり，入力画像の分類ができるように学習が進む．

その後，この出力結果をそのまま Context-aware GAN へ入力するのではなく，出力 softmax 関数のうち，高くなっている確率の方を 1, そうでない方を 0 としてベクトル化し，Context-aware GAN へ入力する．

Algorithm1 にアルゴリズムを示す．画像とラベルを学習データから取り出し，ニューラルネットワークを Algorithm1 に記載された式で学習させる．ここで，$l^{(i)} * log[R(x^{(i)})]$ の乗算は 2 つの数とも 2 要素を持つベクトルであり，出力はその内積となる．

Algorithm 1 recognizer algorithm.

1: **for** number of GAN training iteration (Step 1) **do**
2: *Sample minibatch of m images* $\{x^{(1)}, ..., x^{(m)}\}$ *from data*
3: *distribution* $p_{data}(x)$
4: *Sample minibatch of m labels* $\{l^{(1)}, ..., l^{(m)}\}$ *from data*
5: *distribution* $p_{data}(x)$
6: *Update the neural network by ascending its stochastic*
7: *gradient* :

$$\nabla\theta_n \frac{1}{m}\sum_{i=1}^{m}[l^{(i)} * log[R(x^{(i)})]]$$

8: **end for**

3.2.2 テキストの前景・背景処理

StandfordParser を利用して，構文木を構築して，登場エンティティと場所を抽出する．登場エンティティを前景，場所を背景とする．

3.3 画像生成ネットワーク

入力テキストのみを考慮して生成された画像は適応場面を考えると実用的でない場面がある．そこでテキスト情報だけでなく，オブジェクトや背景などコンテキスト情報も反映させる．学習データとして前の段階でも用いていたテキストと画像のペアに加え，コンテキストネットワークによってコンテキスト画像から抽出された情報を用いる．

ここでは小さな画像から徐々に大きな画像へ再帰的に生成を行っていき，最終的に鮮明な画像を生成する．Vondrick らのモデル [10] に用いられている Multi-Scale Architecture を用いることで，大域的な特徴と局所的な特徴の両方を捉えた画像を生成することができる．

一番最初に画像を生成する Generator には一様分布から得た乱数とテキスト，コンテキスト画像から得られた識別器の出力結果を入力する．二つ目以降の Generator については乱数の入力は行わず，代わりに前段階の Generator により生成された小さい画像を入力とし，より大きな画像を生成する．

ここで，コンテキスト画像から得られた識別器の出力結果とは，前景識別器と背景識別器について出力された結果と入力テキストとの差分から求められる．具体的には，入力テキストに背景の情報が描写されていればそれが背景であり，描写されていなければコンテキスト画像から背景識別器を用いて抽出された背景情報を反映させる．また，入力テキストに前景エンティティの情報が描写されていたとしても，その詳細はコンテキスト画像にある場合があるので，テキストにある情報のより詳細なものをコンテキスト画像から抽出し反映させる．コンテキストの入力値は以上の情報から前景・背景についてそれぞれの one-hot vector となる．

Stage2 では画像を入力としているため，その画像から特徴を取り出す必要がある．配置されているオブジェクトをより鮮明にしながら自然な写真になるよう，局所的な特徴を捉えるためダウンサンプリングを行う．ダウンサンプリングにより抽出された特徴と，ベクトル化されたテキスト情報を組み合わせてアップサンプリングを行うことでより大きく鮮明でテキスト，コンテキストを考慮した画像生成を行う．

テキストをベクトル化するには Reed らの手法 [5] で用いられている，画像のより詳細な情報を反映してテキストをベクトル化する手法 [18] を用いる．

3.4 損 失 関 数

Generator の損失関数について，最終的に生成される画像を学習データに近づけたいが，そのためにはそれぞれの Generator で生成される画像がすべて，その出力画像サイズにリサイズされた学習データに近いものである必要がある．よってそれぞれの Generator について，判別器となる Discriminator を準備し，各々の Generator の損失関数を合計したものが Context-aware GAN の損失関数と定義できる．よって Context-aware GAN の Generator は以下の L_G を最大化するよう学習を行う．

$$I(G_k) = \begin{cases} G_k(z, C, T) & (k = 1) \\ G_k(I(G_{k-1}), C, T)) & (k > 1) \end{cases}$$

$$L_G = \sum_k \mathbb{E}_{(C,T)\sim p_{data}}[log(D(I(G_k), C, T))]$$

ここで，$I(G_k)$ は k 番目の Generator から出力された画像である．初めの Generator は入力が z であり，それ以降は 1 つ前の出力画像を Generator に入力することで得られている．これを用いて損失関数を求める．損失関数はすべての Generator の損失関数の話であるため，上式のように再帰的に定義することができる．なお，C はコンテキスト画像を識別器に入力した際の出力結果，T はテキスト情報である．$D(I(G_k), C, T)$ は Discriminator に $I(G_k), C, T$ を入力した時の判別結果 [0, 1] である．

また Discriminator は L_D をそれぞれ最小化するように学習を行う．Discriminator についても Generator と同様にそれぞれの Discriminator の損失関数の合計となっている．

$$\begin{aligned} L_D = & \sum_k \mathbb{E}_{(X,C,T)\sim p_{data}}[-log(D(X, C, T))] \\ & + \mathbb{E}_{(C,T)\sim p_{data}}[-log(1 - D(I(G_k), C, T))] \end{aligned} \tag{3}$$

ここで X は学習データセットにある生成対象の元画像である.Discriminator を学習させる際は，本物の画像が入力されれば 1 を，生成画像であれば 0 を確率として出力するようにする．

3.5 Generator と Discriminator の構造

次にそれぞれの Generator, Discriminator の構造について述べる．

一番最初に用いられる GAN(Stage1) と 2 回目以降に用いられる GAN(Stage2) の構成は異なっている．3 回目以降の GAN については 2 回目の GAN と全く同じものをパラメータの数を変えて用いているだけなので，同じ構成になっている．

- Stage1 について : Stage1 の役割は，入力されたテキスト，コンテキストに沿っている画像の外形を出力することである．出力される画像は初めはかなり小さいため，鮮明には成り得ないが，鮮明にしていく過程は Stage2 が行う．

入力されたテキスト，コンテキストに沿って画像生成を行うモデルは Reed らのモデル [5] を用いた．このモデルとの違いはテキスト情報に加えてコンテキスト情報も入力としているため，テキストとコンテキストをエンコードしたものを合成したベクトルを Generator の入力としている点である．

一様分布から得られた乱数，テキスト，コンテキストをすべてベクトル化しそれを Deconvolution によりアップサンプリングを行い，画像を生成する．生成された画像が Discriminator を欺けるように Generator を学習させる．

- Stage2 について : Stage2 が Stage1 と違う点は入力として画像が与えられている点である．サイズは小さいが，元となる画像が与えられるということは，コンテキストやテキストに沿いながら，与えられた画像をより鮮明に，かつ大きくしてい

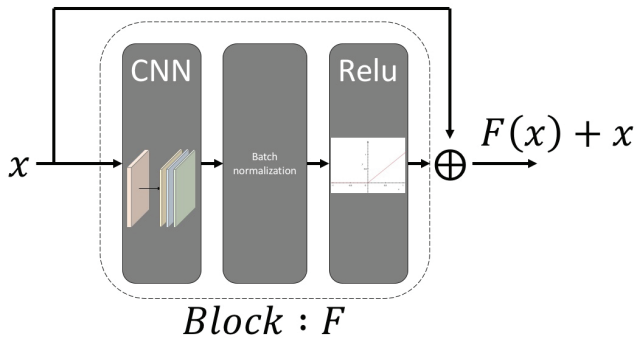

図 4　Residual Blocks

く必要がある．

Reed らのモデル [5] では画像を入力しないので，Stage2 のモデルには用いることができない．ここには Zhang らの提案しているモデル [11] の構造を用いる．このモデルは Stage1 で Reed らのモデル [5] と同様にテキストから 64 × 64 の画像生成を行う．そのあと，Stage2 でテキストに加え，Stage1 の出力結果である画像を入力し 256 × 256 の画像を生成する．Stage2 はテキストに沿った画像を作るだけでなく，入力された画像をより鮮明にする働きがある．よってこのモデルを用いることで，前段階で生成された画像をより鮮明にしてテキスト，コンテキストに沿った画像生成を行うことができる．

ここで Generator は入力された画像をまずダウンサンプリングし特徴を抽出する．この特徴ベクトルとテキストをエンコードしたベクトルを合成する．この合成ベクトルをアップサンプリングしていくことで入力された画像よりも，より鮮明な画像を生成することができる．

ダウンサンプリングし合成したベクトルをアップサンプリングする前に He らの提案する Residual Blocks [17] を用いる．より大きく鮮明な画像を作るにはニューラルネットワークの層を深くする必要がある．しかし，単純に層を連結していくと学習が進みにくくなり，時間も多くかかってしまう．そこで図 4 のように，ある層と batch normalization, Relu をひとまとめに Block とし．その block の出力結果と block への入力値の合計をこの複数層の出力とする．これにより，層は必要な差分を学習することになり，より細かな情報を早く学習することができる．

Stage2 を再帰的に定義して Context-aware GAN を構築する．ただし，パラメータの数は入出力される画像サイズが異なるため毎回変える必要がある．

Algorithm2 にアルゴリズムを示す．この式のように再帰的に Generator の出力を用いて損失関数を定義していく．

3.6　学習について

再帰的に定義するうえで，ニューラルネットワークのサイズはより大きな画像を出力する Stage2 の方が大きくなる．そのため，この学習を行うと，小さな GAN の学習よりも大きな GAN の学習が先に進んでしまう．起きな GAN の学習が進まなくなってくると，その一つ前の GAN の学習が進み，それによって入力となる画像の鮮明化が進み，また GAN が学習していく．

Algorithm 2 Context-aware GAN training algorithm.

for number of training iteration **do**

Define Generators $\{G^{(1)}, ..., G^{(k)}\}$ *and*

Discriminators $\{D^{(1)}, ..., D^{(k)}\}$ $(now\ k\ =\ 3)$

Sample minibatch of m noise samples $\{z^{(1)}, ..., z^{(m)}\}$

Sample minibatch of m images $\{x^{(1)}, ..., x^{(m)}\}$ *from data distribution* $p_{data}(x)$

Sample minibatch of m texts $\{t^{(1)}, ..., t^{(m)}\}$ *from data distribution* $p_{data}(x)$

Sample minibatch of m context images $\{c^{(1)}, ..., c^{(m)}\}$ *from data distribution* $p_{data}(x)$

Define real_rogit and fake_logit

$$real\_rogit = \sum_{i=1}^{k}\sum_{j=1}^{m}[D^{(i)}(x^{(i)}, t^{(i)}, c^{(i)})]$$

$$I(G_k) = \begin{cases} G_k(z, C, T) & (k = 1) \\ G_k(I(G_{k-1}), C, T)) & (k > 1) \end{cases}$$

$$fake\_rogit = \sum_{i=1}^{k}\sum_{j=1}^{m}[D^{(i)}(I(G_i), t^{(i)}, c^{(i)})]$$

Update the discriminator by ascending its stochastic gradient :

$$\nabla\theta_d[log(real\_logit) + log(1 - (fake\_logit)$$

Update the generator by descending its stochastic gradient :

$$\nabla\theta_g[log(1 - (fake\_logit)$$

end for

本手法のモデルでは，より小さいサイズの画像を入力として画像生成を行うため，その入力画像がしっかり生成されているほど，GAN の学習は進みやすいと考えられる. よって本来は小さい GAN から学習が進んだ方が収束は早くなるはずであるので，本手法では収束までの時間が通常の GAN の学習に比べても長くかかる．

4.　実験結果とその考察

Generator の数は 3 として実装を行い，実験を行った (k=3).

今回の実験ではコンテキストとしてコンテキスト画像の背景のみを考慮している．今後は，場面の情報だけでなく，映っているオブジェクトの情報 (例えば，映っている人物が男性なのか女性なのか，何人いるのかといった情報) も判別器で判断し入力できるようにしたい．

したがって，今回は画像の背景識別器の出力結果を単純利用して Context-aware GAN を実装しているが，今後の課題として，出力結果と入力テキストの関係性を考慮して，コンテキスト情報として必要な情報を判断した上で入力とできるように学習を行えるよう改良していきたい．

4.1　出 力 画 像

図 5 に示す．データセットの画像は全て 400 × 700 のサイズであるが，計算機資源を考慮して，本研究では学習データを 60 × 105 に縮小して行った．1 つ目の Generator の出力サイズを

図 5　Context-aware GAN の出力画像

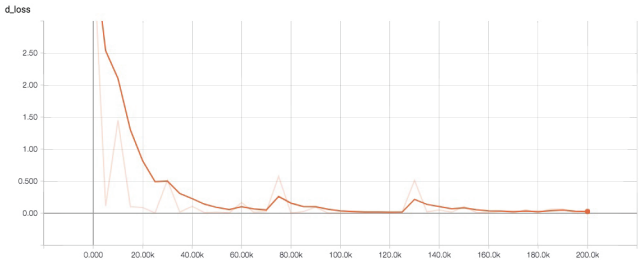

図 6　discriminator の損失関数

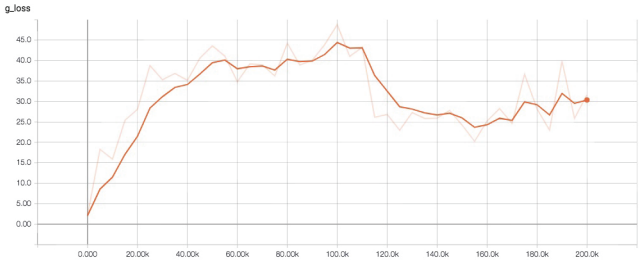

図 7　generator の損失関数

15 × 26，2 つ目の Generator の出力サイズを 30 × 52 としている．

4.2　損 失 関 数

図 6 から discriminator の損失関数は収束していることが分かる．図 7 は generator の損失関数の収束を示したものであるが，12,000 回目の学習あたりで値が小さくなっているのが分かる．これは，3 つ目の Generator の値が変化しなくなってきたので，次に大きい 2 番目の Generator の学習が進んで，その結果値が小さくなったためである．

この Generator の損失関数を見ると，学習回数が不足していることが分かる．

図 8 にあるように 100,000 回転では最終的な生成画像は収束していることが分かるが途中の生成画像 (Stage2(1)) は全く学習できていないことが分かる．この画像が 150,000 回転目では学習できている．これが図 7 で 120,000 回転あたりで損失関数が落ちているところである．これ以降，最終的な生成画像もより鮮明なものになっている．この学習が進めば，Stage1 の生成画像も鮮明になるはずで，それが進めばより鮮明な画像が生成されるようになる．

よって学習時に各過程の画像の学習の重み付けをどのようにするかが鮮明な画像を生成するために必要であるということがこれらのグラフから読み取ることができる．

5.　まとめと今後の課題

コンテキストを考慮した画像生成を行うために GAN への入力としてテキストに加え，コンテキスト画像を識別器に入力した出力結果を与える Context-aware GAN モデルを提案した．本研究における提案手法では従来手法が入力とすることのできないコンテキスト情報を反映した画像を生成した．

課題を以下に列挙し，今後これらについて検討する予定である．

- 実写画像を学習データにする．その上で，実写の画像の適切なキャプションに加え，オブジェクト情報や背景情報が適切に付与されているデータセットが必要である．
- 学習速度の向上．複数の GAN を組み合わせているため，収束速度にばらつきがあり，そのため，どれか GAN が収束するたびに他の GAN が影響を受けるので，最終的に収束するまでにかなりの時間がかかってしまう．

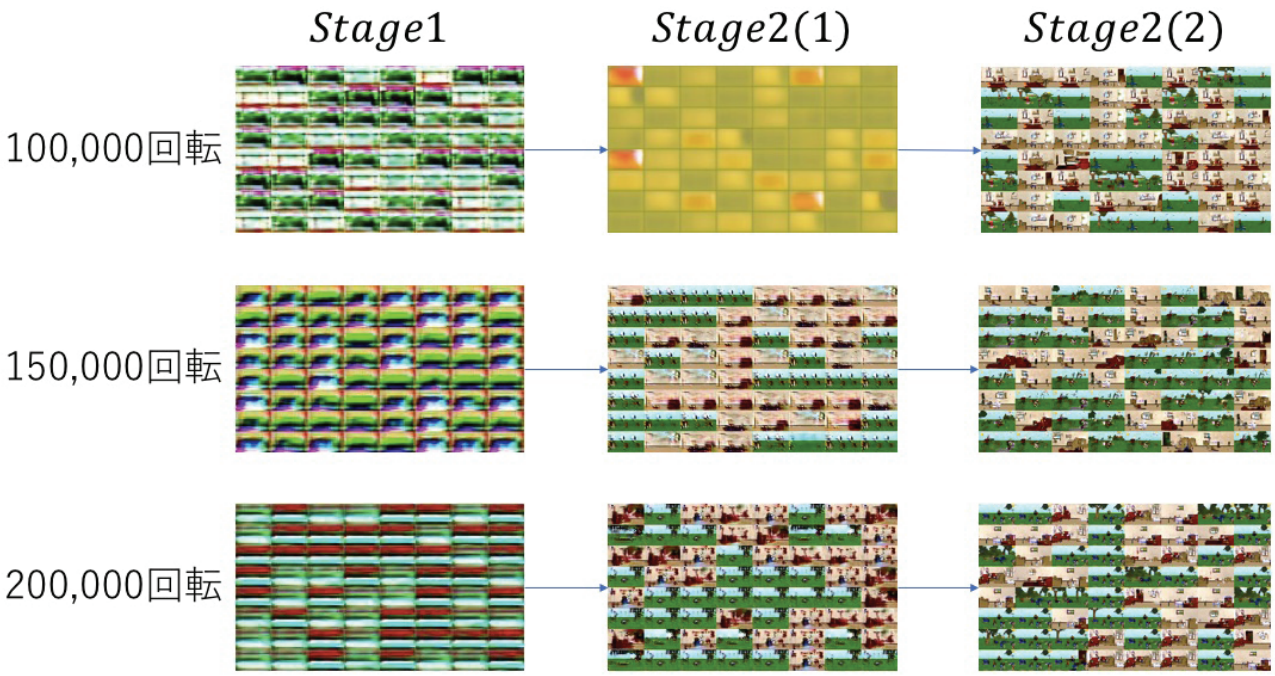

図 8　Context-aware GAN の学習経過

- コンテキスト情報の拡大．本手法では背景のみをコンテキスト情報としているが，映っているオブジェクトや人物などもコンテキスト情報として扱いたい．そのため，これらの情報を抽出し，入力テキストとの関連からコンテキストとして適切な情報を入力として与え，よりコンテキストを反映した画像生成を行いたい．

6. 謝　　辞

本研究の一部は総務省 SCOPE(172307001) による．

文　　献

[1] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. (2014). Learning deep features for scene recognition using places database. In Proceedings of Advances in neural information processing systems (pp. 487-495).

[2] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3156-3164).

[3] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Proceedings of Advances in neural information processing systems (pp. 2672-2680).

[4] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

[5] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016). Generative adversarial text to image synthesis. arXiv preprint arXiv:1605.05396.

[6] 中村玄貴, 馬強, “コンテキストアウェア敵対的生成ネットワークによる画像生成”, 第 9 回データ工学と情報マネジメントに関するフォーラム

[7] Kenki Nakamura, & Qiang Ma.(2017). Context-aware Image Generation by Using Generative Adversarial Networks. In Proceedings of the IEEE International Symposium on Multimedia (pp. 516-525).

[8] Ferraro, F., Mostafazadeh, N., Misra, I., Agrawal, A., Devlin, J., Girshick, R., ... & Parikh, D. (2016). Visual storytelling. arXiv preprint arXiv:1604.03968.

[9] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan. arXiv preprint arXiv:1701.07875.

[10] Mathieu, M., Couprie, C., & LeCun, Y. (2015). Deep multi-scale video prediction beyond mean square error. arXiv preprint arXiv:1511.05440.

[11] Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., & Metaxas, D. (2016). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. arXiv preprint arXiv:1612.03242.

[12] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., & Parikh, D. (2015). Vqa: Visual question answering. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2425-2433).

[13] Redmon, J., & Farhadi, A. (2016). YOLO9000: better, faster, stronger. arXiv preprint arXiv:1612.08242.

[14] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint arXiv:1703.10593.

[15] Choi, Y., Choi, M., Kim, M., Ha, J. W., Kim, S., & Choo, J. (2017). StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. arXiv preprint arXiv:1711.09020.

[16] Yu, L., Zhang, W., Wang, J., & Yu, Y. (2017, March). SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In Proceedings of AAAI conference on Artifical Intelligence (pp. 2852-2858).

[17] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[18] Reed, S., Akata, Z., Lee, H., & Schiele, B. (2016). Learning deep representations of fine-grained visual descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 49-58).