

## 第5章 東海道五十X プロジェクト(大島圭佑)

### はじめに

筆者がZAFにおいて発表するのは今回で3度目になります。1度目は2019年の夏に「AI開発に必要なもの」という題で、2度目は2019年の冬に「TOEIC500点台の僕が1日10本以上英語論文を読めるようになるまで」という題で発表させていただきました。この2度の発表では「AIの開発において大事なのは気合と根性」というお話をした気がします(正直な話をしてるとAIという言葉はあまり好きではないので自分は基本的に人工知能と言っています)。この発表を聞いた方の中には「AIの開発って想像以上に地味で地道だな」と感じた方もいるかもしれません。発表では主に学習データ作成を頑張ったお話をしましたが、実はそれ以外でもたくさんの方業を試しています。



AIエンジニアと呼ばれる人の中にはアカデミックな知識を持って実世界の現状の課題解決であったり、現在世にあるDeepLearningのモデルアーキテクチャの改善手法などを考えたりしている人たちがいます。彼らは筆者からすると雲の上の存在で、平たく言うと「超賢い人が超タフなことをやっている」というイメージでいます(本当のところはよくわからないですが)。かたや筆者はというと、基本的な理論を抑えているつもりではいますが超賢いわけではないので最後の最後は力業に頼りがちです。さすがに雲の上の存在と張り合うつもりはありませんが、ある程度の頭脳、理論の理解の差は力業で埋められるのがDeepLearningのいいところでもあります。真面目に試行錯誤をすれば精度を上げられるため日本の技術者に向いている技術と言われたりもします。要はやる気があれば決してハードルが高い技術ではないわけです。先の2回の発表ではそのこともあって気合と根性が大事だというお話をしました(発表当時はそこまで真面目に考えていたわけではないですが)。

と、散々言ったわけですが今回は珍しく力業ではなく、「いろいろ試してみました」というある意味前回々回と比べると一番ZAFらしい発表となりました。特に今回はZAFへの発表を見据えた開発をしているので、前回までの発表と比べると技術的には少し難しいかもしれませんが面白い発表になっていると思います。

# 1. 東海道五十 X プロジェクト

## 1. 1. 東海道五十 X プロジェクトとは？

平たく言うと「AI を使って東海道五十三次に続く存在しない浮世絵を作ろう」というプロジェクトになります。東海道五十三次については名前だけでも読者はなんとなくでもご存じだとは思いますので割愛します。おそらく Nao Honda 氏が説明してくれていると思いますので。



## 1. 2. 取り組みのきっかけ

Nao 氏が紹介してくれているとは思いますが、今回の取り組みのきっかけを筆者目線で簡単に説明いたします。筆者が初めて今回の取り組みとなる「東海道五十 X プロジェクト」について聞かされたのは去年の夏ごろ（現在は 2021 年 4 月）だったと記憶しています。そのお話を聞いたときはやろうやろうという話にはなったのですが、個人的に忙しかったために当時はあまり手を付けられず（すみません）。先月の頭に ZAF で発表するというお話を Nao 氏から聞いて取り組みを再開したわけです。

## 1. 3. 開発の方針

AI で浮世絵を作るというお話を聞いたとき、筆者の頭には以下の 2 通りのやり方が浮かんでいました。

1. ゼロから浮世絵を生成する
2. 写真、絵などを浮世絵に変換する

Nao 氏は「GAN で浮世絵作ろう」とだけ言っていましたが、どうやら彼女の中ではゼロから浮世絵を作る気だったようです。ただ線画から浮世絵を作ることもできるということを話したところそちらの方が琴線に触れたようで、めでたく（？）2 の「写真、絵などを浮世絵に変換する」方法で開発を進めることになりました。

技術はスタイル変換系の DeepLearning の技術である CycleGAN を試しています。選定した理由は「よく聞くから」という適當なものなのですが、そこはご愛嬌ということで。ただ、試してみたところ期待以上に「いい感じ」にしてくれるので最初に触ってみるにはちょうどいい技術ではないでしょうか。逆に言うと適当にやってもそれなりにいい絵を出力してくれるのによく聞くでしょう。ここではまず CycleGAN について説明していきます。基本的に中身に関しては知らなくても問題ないので興味がない方は飛ばしちゃって大丈夫です。

## 2. CycleGAN について

### 2. 1. GAN

CycleGAN は、主に画像生成等に使われる GAN と呼ばれる技術の一つです。GAN をご存じの方にとっては耳タコかもしれません、ご存じない方のために少しだけ説明します。  
2. 1. 1. GAN の大枠

GAN とは敵対的生成ネットワークと呼ばれるもので、以下の二つのモデルがお互いに敵対し、競い合うことで学習を進めていく仕組みとなっています。

- Generator : 学習画像をもとに架空の画像を生成する生成器
- Discriminator : Generator が生成した画像か本物の画像かを見破る画像認識器

流れとしては基本的に以下の通りです。

- ① Generator に値を入力し、生成画像を出力（多くの場合入力は乱数ベクトル）。
- ② ①の架空画像、学習画像（本物画像）Discriminator に入力、本物か生成画像を識別。
- ③ Generator は生成画像を Discriminator に本物画像と誤認させるように、Discriminator は生成画像を本物画像と見分けられるようにそれぞれ学習、①へ。

この一連の流れを繰り返すことで人でも見分けがつかないような完成度の高い架空の画像を生成することができます。

### 2. 1. 2. ルパン VS 銭形警部

先ほどの説明ではいまいちイメージがつかなかった方のためにもう少しだけ。GAN の基本的な仕組みはアニメ「ルパン三世」に出てくる主人公ルパンと銭形警部の関係性で説明できます。（※ルパン三世をご存じない方はスルーしてください。数分時間を無駄にすること

になります。)

ルパンは一流の怪盗でありながら変装の名手であり、本人と寸分たがわぬ変装でたいでいの警察官を巻いてしまいます。しかしながら銭形警部はそんなルパンの変装をわずかな違和感で見破る術に長けており、彼だけはほぼ毎回変装を見破ってしまいます。これは最初から銭形警部がルパンの変装を見破られたわけではなく、何度も何度もルパンを追い詰めた経験の賜物とみることができます。そしてルパンもまた、そんな銭形警部の目を欺くために変装の腕を上げていているに違いありません。

このときルパンを Generator、銭形警部を Discriminator とみることができます。まさにルパン (Generator) は銭形警部 (Discriminator) を欺くように変装し、銭形警部 (Discriminator) はルパン (Generator) の変装を見破れるように目を光らせるわけです。そうやって、ルパンと銭形警部の欺き、見破るいたちごっこは私たちのような第三者（ルパン三世で言うなら普通の警官でしょうか）から見ると何がどうなっているかわからないようなレベルの高い変装（生成画像）に見えるわけです。

## 2. 1. 2. Mode Collapse

発表でもお話ししたので、GAN 特有の問題として Mode Collapse について簡単に書いておきます。Mode Collapse とは Generator がある特定の画像を出力し続ける、いわば学習が崩壊してしまった状態を指します。これは GAN が二つのモデルを同時に学習させ、しかも両モデルの実力の均衡を保ちながら学習を進めないとうまく学習が進まないという繊細な問題から来ています。例えば 0~9 の 10 種類の手書き数字を生成しようとした場合、「1」の画像は縦棒が一本で最も単純な形をしています。もしも Generator 様が「1 は簡単に出力できて、Discriminator を簡単に欺けるから 1 だけ出力してればいいや！」となるとそこで学習が止まっちゃうわけです。GAN の技術の進歩はこの Mode Collapse との闘いでもあります。

## 2. 2. CycleGAN

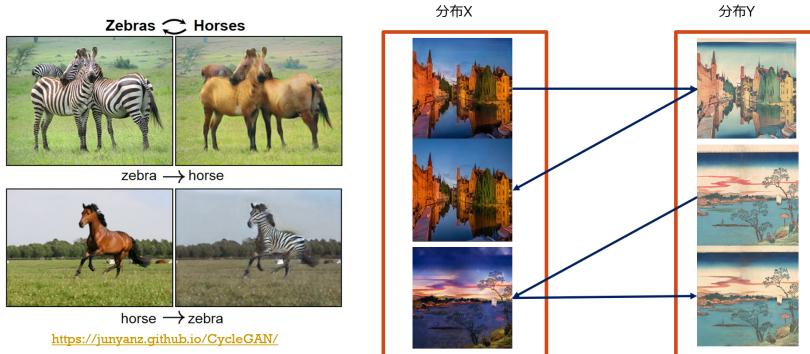
GAN についてはこの辺にして、CycleGAN について説明します。CycleGAN は GAN の仕組みを使った画像のスタイル変換技術です。スタイル変換には CycleGAN が出た時点での Pix2Pix をはじめとした GAN 系の技術がすでにありましたが、CycleGAN は学習画像に対する画像が必要ないという大きな利点がありました。要するにそれ以前は線画と写真の変換を学習させる場合写真と、その写真をなぞったような線画が必要だったわけです。

### 2. 2. 1. CycleGAN の技術概要

CycleGAN の学習方法は、ある分布 X の画像から分布 Y の画像へ変換する場合、、、と説明するとよくわからなくなるので、写真と浮世絵の変換を考えてみます。通常のスタイル変換技術では写真を浮世絵に変換する際は写真と浮世絵の組を用意し、写真→浮世絵の変換の学習を行うという素直なアプローチをとります。一方 CycleGAN では写真→浮世絵の変換と同時に浮世絵→写真の変換を学習させます。この操作は Mode Collapse を防ぐ働きを

します。Mode Collapse は Generator がどんな入力に対しても同じような画像を出す現象を引き起こしました。CycleGAN では写真→浮世絵→写真という変換を行い、変換前の写真と変換後の写真の差がなくなるように学習させます。これにより写真→浮世絵の変換において、写真によって別の画像を出力させる制約を加えるわけです。この時写真→浮世絵→写真の変換と同時に浮世絵→写真→浮世絵の変換も学習しますので、写真→浮世絵の変換器(Generator)と浮世絵→写真の変換器が同時に作られることになります。少し乱暴な言い方ですがこの枠組みでは写真と、それを描きなおした浮世絵の組が必要ないことがわかります。雰囲気だけでも理解していただけたでしょうか。なおこの CycleGAN にも問題があるて、写真→浮世絵の GAN と浮世絵→写真の GAN が必要になるので単純に 2 組の GAN となりリソースが通常の 2 倍必要になります。。。

やったこと : CycleGANでスタイル変換



### 3. 開発、検証

Nao 氏は線画から浮世絵を作りたいと言っていましたが、線画を持ってないのでちょっと順を追って検証を進めることにします。学習画像は以下のようなものを使っています。

今回使う学習画像 : 歌川広重



今回使う学習画像 : 情景画像

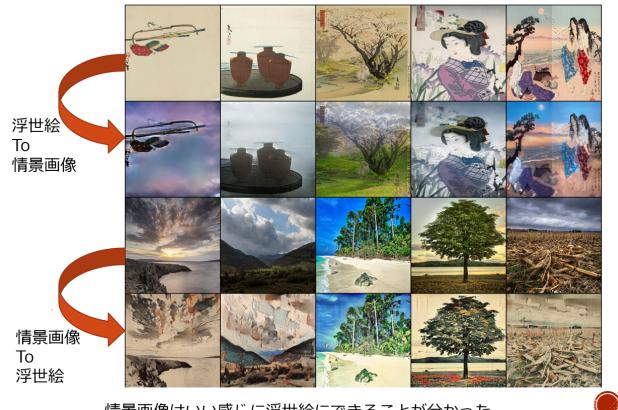


#### 3. 1. 情景画像→浮世絵

線画は持っていないので、まずは情景画像(画像処理の世界では風景などの写真を慣習的に

情景画像と呼びます) を浮世絵に変換するという学習をしてみました。これは予想以上にうまくいっているように見えます。

#### 検証1：浮世絵to情景画像 & 情景画像to浮世絵



#### 3. 2. 情景画像→歌川広重

情景画像→浮世絵が予想以上にうまくいっていましたが、いきなり線画から！ではなく少しづつ検証を進めます。ここでは浮世絵を歌川広重（東海道五十三次の作者）が描いたものに限定して学習してみました。これも期待通りの出来です。次はいよいよ線画から変換を試してみます。

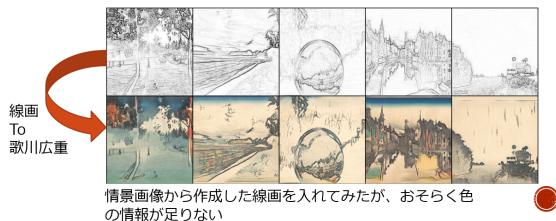
#### 検証2：歌川広重to情景画像 & 情景画像to歌川



#### 3. 3. 線画→歌川広重

いよいよ線画から歌川広重の画像を作ってみます。この時線画を持っていないので情景画像から線画を生成しました。筆者はこの学習に用いた線画を事前に Nao 氏に見せて「このくらいの線画ならすぐ描けたりしますか?」と聞いたところ「すぐ描けるんじゃない?」という回答が。つまり、この学習がうまくいけば大方の開発は終了というわけです。結果は御覧の通り。

#### 検証3：線画to歌川 & 歌川to線画



決して悪くはないものの、やはり線画だけでは色の情報が足りないようです。最初から分かっていたことではあります。

#### 3. 4. 線画→歌川広重（生成画像）

次に試したのは 4. 2 で試した、情景画像→歌川広重で生成した歌川広重画像を学習データに加える手法です。期待はしていたものの、ちょっと筆者の思惑とは違う画像が出力されています。やっぱり線画から自然な色を付けさせるのは難しいようです。市來さんはじめ何人かの方からは「これはこれで味があっていい」という風に言っていただきました。確かに少し陰鬱な雰囲気が漂っていてこれはこれで雰囲気があるよう見えます。ただ筆者からするとこれは歌川広重ではなく、東海道五十景ではありません！

また、写真→歌川広重じゃダメなの？というお話もされました。そこはやはり線画にこだわりたいです。何しろ Nao 氏が線画からがいいと言っているので（彼女が言うには自分で描いた絵が浮世絵になるのがいいらしいです）。

今回の技術開発はここまで来て時間切れとなりましたので、次回発表する機会があればそれまでに何かしら進展させようと思っています。

#### 検証4：線画to歌川（フェイク） & 歌川（フェイク）to線画



## 4. 実践

線画から浮世絵の開発は志半ばではありますが、写真から浮世絵はある程度うまくいっているということでいろいろ試してみました。

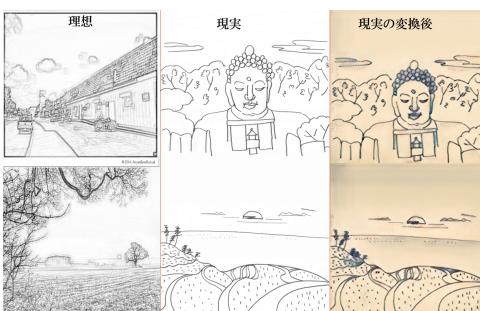
#### 4. 1. 写真→浮世絵

Nao 氏が紹介してくれていると思うので、ここでは発表でお見せしたダイジェスト画像を貼ってみます。下図の左が情景画像、右が変換した浮世絵です。やはり情景画像を浮世絵にすると絵になります。特にタージマハルは何かのアイコンにしたくなるくらいきれいです。これはこれで開発の成果としては十分だと思います。



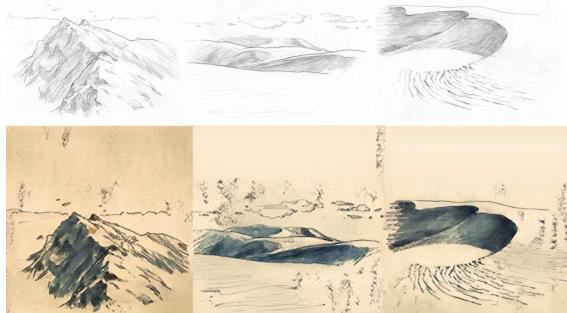
#### 4. 2. 線画→浮世絵

一応線画から浮世絵の変換も試してみましょう。筆者が提示した線画のクオリティに対して Nao 氏は「これならすぐ描けるんじゃない？」と言っていたので書いていただきました。それがこちらになります。並べてみると想像以上に理想と現実が乖離していました（下図左が理想、中央が現実）。ですがせっかく描いていただいたので変換してみましょう。結果は下図右側になります。どうやら書き込みが少なすぎて何を描いてあるのか、どうやって色を付けていいのかわからなかったようです。次回描いていただく際は書き込みを徹底してもらうことにします。



ちなみにですが、それなりに書き込まれている絵を変換すると下図のようになりました。描いていただいたのは Nao 氏と共に知り合いであるデザイナーの佐伯有果氏です。下図上段が変換前、下段が変換後です。影の部分が青く塗られ、文字っぽい黒い模様？が出ただけ

ではありますが、やはり書き込みをすると色を塗ってくれるようです！描いた本人は山の稜線と砂漠を描いたと言っていましたが、問答無用で青色になったのは砂漠を海と判断しているからでしょうか。



## 5. 今後

今後何をするか決まっているわけではありませんが、ゆるーく思っていることを描いておこうと思います。

- ① LightWeightGAN を試す
- ② TransGaGa を試す
- ③ (ブロックチェーンと NFT トークンで XX)

①について、軽量で完成度の高い絵を出力する LightWeightGAN を CycleGAN にどこにかこーにか導入できないか考えています。モデル構造に工夫があるみたいです。②は CycleGAN の発展形になるのですが、CycleGAN と違って大きく輪郭を変える様な変換、例えば馬→キリンの様な変換でも対応できる技術になります。③については Nao 氏と一緒にブロックチェーンの NFT プラットフォームで絵を売ってみよー！という取り組みです。すでに「NamcO」というユニット名も決まっています。できれば 6 月に発表できるように取り組みを進めたいと思います。

今回参考にしたソースコード：<https://github.com/jesa7955/CycleGAN-PyTorch>

※学習データに使った情景画像、検証 1 の浮世絵画像は上記 URL をたどって入手しています。歌川広重の学習画像は市來さんよりいただいたものを使用しています。

## おまけ：ベクシンスキー

せっかく CycleGAN をそれなりに動かせるようになったので、浮世絵以外の変換も試し

てみました。試すのは情景画像→ベクシンスキーの変換です。ベクシンスキーとはポーランドの画家で、ナチスへの恨みを絵にした画家だそうです（Nao 氏曰く）。画風はひたすら陰鬱で退廃的な雰囲気で家に飾るのは少し気が引けます。



これと情景画像の組を学習し、情景画像→ベクシンスキーを試してみましたが、色味がくすむ以外はたいして面白い画像は出力されませんでした。

#### 情景画像toベクシンスキー



しかしながら、ベクシンスキー→情景画像の変換では筆者の琴線に触れる画が出力されていました。もともとベクシンスキーの描く絵は筆者にとってどこか別の星の映像にも見えていましたが、情景画像に変換されたベクシンスキーの絵はまるで人が減んだあの生命にあふれる世界に見えます。AI とアートでは開発者が意図しない産物が生まれるところが面白いのですが、まさにそれが見れた試みでした。



筆者の記事はここまでになります。読んでくださった方、ありがとうございました。次回また記事を書く機会をいただければもう少し読みやすい記事を書こうと思います。