画像内容を考慮した質感表現に基づく画像変換

杉山 優 *1　　柳井啓司 *2
Yu Sugiyama　　Keiji Yanai

*1電気通信大学
The University of Electro-Communications

In recent years, deep learning has attracted attention not only as a method on image recognition but also as a technique for image generation and transformation. Above all, a method called Style Transfer [Gatys 16] is drawing much attention which can integrate two photos into one integrated photo regarding their content and style. In this paper, we propose to use words expressing photo styles instead of using style images for neural image style transfer. In our method, we take into account the content of an input image to be stylized to decide a style for style transfer in addition to a given word. We implemented the propose method by modifying the network for Unseen Style Transfer [Yanai 17]. By the experiments, we show that the proposed method has ability to change the style of an input image taking account of both a given word and its contents.

1. はじめに

画像変換の技術として代表的なものにスタイル変換 [Gatys 16] と呼ばれる技術がある。しかしこれはスタイル画像とコンテンツ画像の二枚をネットワークに入力して時間をかけて最適化しなければ変換が行えず、手軽な手法ではない。またスタイル画像が決まって いればどのような画像も似たような変換になってしまうため、コンテンツ画像に似合わないスタイル変換を行ってしまうことも多い。

スタイル変換の改良手法は数多く存在していて、変換を高速化するもの [Johnson 16] や、高精度に変換を行うもの [Liao 17] などが提案された。しかし、これらの多くは変換の前にスタイル画像単独での学習を必要とし、汎用的にスタイル画像を使用できない。そこでこの解決策として提案された手法に任意スタイル変換 [Yanai 17] がある。この手法では事前にスタイルを学習して変換自体は高速に行う、という方針は変えずに、学習したスタイルをベクトル表記していくことによってあらゆる画像をスタイル画像のベクトルとして変換に使用することができるようになるというものである。

しかしスタイル画像が適切ではなかったときに、変換がうまく行かなくなってしまうという欠点を抱えている。そこで入力された画像によって変換を変化させ、画像に適切な画像変換を行うような手法が必要となる。

2. 関連研究

画像変換の手法にはいくつかあるが、その中でもコンテンツ画像に対してのスタイル変換を行っているものとして下の研究がある。

2.1 スタイル変換

スタイル変換の手法として代表的で最もプリミティブなものに Gatys らの Imagestyle transfer using convolutional neural networks [Gatys 16] がある。この手法では、コンテンツ画像とスタイル画像の二種類を用意し、２つの画像どちらにも近づくような画像を作成するようにニューラルネットワークを学習させることで、コンテンツ画像に映っているものは維持しつつ、スタイル画像のような画像の風味に変換する。

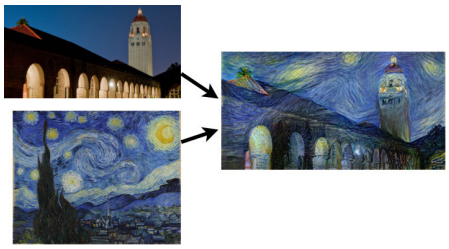

図 1: スタイル変換の様子

変換の学習は、スタイル画像とコンテンツ画像それぞれのロス関数を足し合わせたものを全体のロスとして学習を行う。また、変換には VGG ネットワーク [Simonyan 15] を使用する。VGG の FC 層を除いた部分のパラメータを使用する。画像識別ネットワークの深い層では画像の意味を持った部分だけが判別され、物体認識にはあまり影響しない模様等をスタイル画像に近づけることでスタイル変換を行うことができる。

コンテンツ画像のロス関数は下のようになっている。$\vec{p}$、$\vec{x}$ はそれぞれオリジナルの入力されたコンテンツ画像と生成画像を、P^l、F^l はそれぞれの画像のレイヤ l における特徴ベクトルを示す。

$$L_{content}(\vec{p},\vec{x},l) = \frac{1}{2}\sum_{i,j}(F^l_{ij} - P^l_{ij})^2 \tag{1}$$

スタイル画像のロス関数は下のように計算する。まずはグラム行列 G^l、A^l をスタイル画像と生成画像でそれぞれ計算し、スタイル画像のロスを計算する。ある層のみについて計算するのではなく、複数の層を加算して考える。

$$G^l_{ij} = \sum_k F^l_{ik}F^l_{jk} \tag{2}$$

$$E_l = \frac{1}{4N_l^2M_l^2}\sum_{i,j}(G^l_{ij} - A^l_{ij}) \tag{3}$$

連絡先: sugiya-y@mm.inf.uec.ac.jp

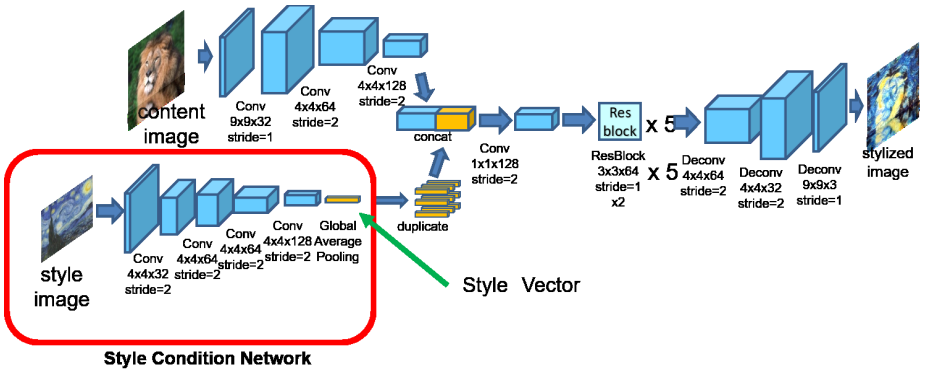

図 2: 任意スタイル変換のネットワーク図 ([Yanai 17] から引用)

$$L_{style}(\vec{a},\vec{x}) = \sum_{l=0}^{L} w_l E_l \tag{4}$$

これらを足し合わせ、学習で最適化していくロス関数は下のようになる。

$$L_{total}(\vec{p},\vec{a},\vec{x}) = \alpha L_{content}(\vec{p},\vec{x}) + \beta L_{style}(\vec{a},\vec{x}) \tag{5}$$

つまりスタイル画像とコンテンツ画像が互いに近づくように最適化していくため、画像の組が変わるたびに時間をかけて画像を最適化する必要がある。これでは画像を入力するたびに長い時間をかけることになるために、変換画像を直ぐに用意したいといった要求に答えることができない。それに対応する研究に高速スタイル変換というジャンルがある。

2.2 高速スタイル変換

スタイル変換を高速化する手法として、Fast Neural Style Transfer [Johnson 16] がある。事前にスタイルを学習しておくことで、新たなコンテンツ画像が入力された際に、学習済みのスタイルに変換するというネットワークを構築するものである。

しかし、高速スタイル変換では事前に学習したスタイル画像しか変換に使用できず、新たなスタイルで変換がしたいという要求にはその都度長い時間をかけてスタイル画像を学習しなければならない。

2.3 任意スタイル変換

スタイル変換の改良手法が多く提案されている中、学習の汎用性を持たせる研究として Yanai の Unseen Style Transfer [Yanai 17] がある。従来の Jonhson らの高速スタイル変換 [Johnson 16] では、学習に用いた 1 種類のスタイル画像のみのスタイル変換しかできなかったが、この研究では、事前に学習に使っていないスタイルも含めて任意のスタイルの変換が可能となるネットワークを提案している。ネットワークは図 2 のようになっている。

高速スタイル変換 [Johnson 16] と同様に、変換を行う前にスタイルだけを学習しておき、実際にコンテンツ画像を入力して変換する際には高速に動作させることができる。さらに高速スタイル変換 [Johnson 16] とは異なり、学習の際に複数画像を用いてスタイル画像から Style Vector を生成する Style Condition Network を学習することにより、任意のコンテンツ画像に対して変換を行うことができる。

事前に複数の画像に対してのスタイルを学習して、任意のスタイルに対応できるネットワークを作成しているために、変換の際にはスタイル画像とコンテンツ画像の両者はどんな画像でも対応でき、高速な変換が可能であることになる。

3. 手法概要

本研究では、Unseen Style Transfer のネットワークの一部を入れ替えるように学習を行うことで画像の変換を実装した。以下に本研究での画像変換の学習と変換の流れを示す。

1. スタイル画像とその特徴を示す単語の組になったデータセットを用意
2. Word2Vec [Mikolov 13] と VGG [Simonyan 15] による Style Selector Network(図 3) を用意
3. Style Selector Network を Unseen Style Transfer の中間ネットワークの Style Conditon Network(図 [Yanai 17]) と近似するように学習
4. 学習した Style Selector Network(図 3) を Unseen Style Transfer の中間ネットワークとして Style Conditon Network(図 2) の代わりに入力し、画像を生成

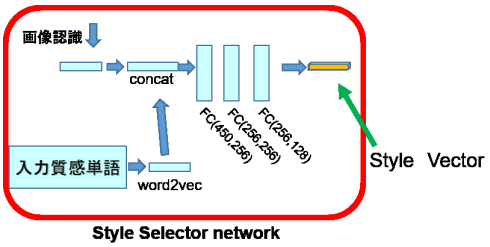

図 3: Style Selector Network

4. 手法詳細

4.1 データセットの構築

データセットとして画像を WikiArt から絵画画像を、Bing 画像 API と Yahoo 検索から質感画像を収集した。それぞれ画像は対応する単語にそれぞれ 500 枚から 1000 枚程度収集し、その中からスタイル画像に適した、画面全体が一様に同じスタイルを示しているような画像を手動で抽出し、各単語 50 枚程度の画像データセットを作成した。データセットに使用した画像の例を図 4 と図 5 に示す。

図 4: 絵画画像データセットに使用した画像例

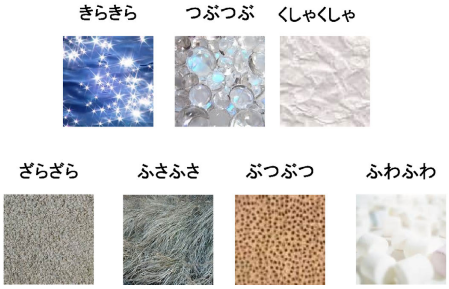

図 5: 質感画像データセットに使用した画像例

4.2 Word2Vec の学習

文字情報を変換のパラメータとして使用するためには、入力された文字列をベクトルで表現しなければならない。それを行うために Word2Vec [Mikolov 13] を使用する。

日本語の Word2Vec モデルは公開されていないため，モデルを学習する必要がある．本研究では，容易に入手可能である日本語 WikiPedia データ (3.1GB) を利用して，日本語 Word2Vec モデルの学習を行った。

4.3 VGG による画像認識

VGG16 [Simonyan 15] を使用して、画像特徴量を取得した。今回は画像の意味特徴を表現していることを期待してこのネットワークの fc7 層にある 4096 次元の全結合レイヤを使用した。

取り出した中間層のベクトルはそのままでは 4096 次元のベクトルだが、これでは Word2Vec [Mikolov 13] の 200 次元というベクトルの次元数と釣り合いが取れず、実際に学習した際に画像特徴を重視しすぎてしまうという良くない傾向が出たために、VGG からのベクトルを次元圧縮した。次元圧縮には PCA を使用し、250 次元までベクトルの次元を削減した。

4.4 Style Selector Network の構成

VGG 画像認識によって得た画像からの中間信号と単語を Word2Vec に入力して得たワードパラメータを結合し 450 次元のベクトルに整形し、それを図に示す FC 層が 3 層で各層に活性化関数 Relu を使用した Style Selector Network(図 3) に通した出力が Style Condition network(図 2) に近づくように収集した画像を用いて学習した。

学習したモデルを使用して画像を変換する際には、Unseen Style Transfer のネットワークのうち、Style Condition Network 部分を Style Selector Network(図 3) に差し替えたネットワークを使用して、スタイル画像を入力する代わりに、変換したい言葉を入力することによって変換を行った。

学習に使用した Style Selector Network のロス関数 L_{SNN} は下のようになっている。L2 距離を最小化することで学習を行った。v_i と w_i はそれぞれ画像特徴と単語特徴のベクトルを示し、s_i は Style Vector を示す。

$$L_{SNN} = \frac{1}{2}\sum_i (s_i - SNN(v_i, w_i))^2 \tag{6}$$

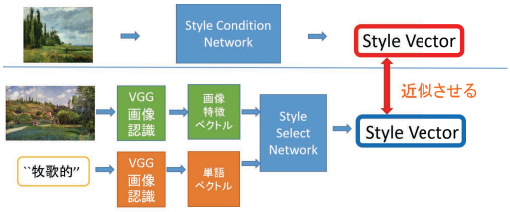

図 6: 学習の概略

5. 実験

実際に画像を学習したネットワークに通して変換を実験した。実験は下に示す 2 通り行った。各実験について、上手く変換できた例とできなかった例を示す。

1. 単語を固定し画像だけを変化させた場合
2. 画像を固定し単語だけを変化させた場合

5.1 単語を固定し画像だけを変化させた場合

まず、画像の変化によって変換が変化するかということを観察するための実験を行った。図 7 は元画像に対して “油絵” という単語で、絵画スタイル変換を行った様子。図 8 は元画像に対して “きらきら” という単語で質感スタイル変換を行った様子を示す。単純にスタイルが変換されているのではなく、コンテンツ画像によって異なる変換が行われたことが見て取れた。

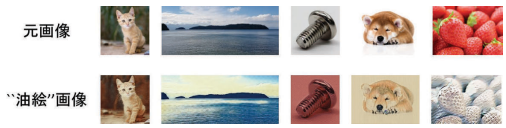

図 7: “油絵” で画像を変換した様子

図 8: “きらきら” で画像を変換した様子

逆に、変換がうまくいかなかった例では、次のようなものがあった。変換がうまく学習できずに、出力画像が入力画像とほぼ変わらないという失敗であったり、出力画像が元画像の意味を保持できていないものがあった。

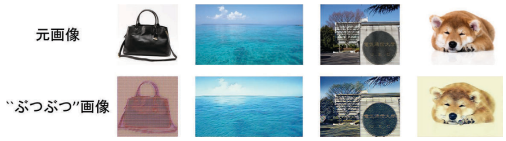

図 9: “ぶつぶつ” で画像を変換した様子

5.2 画像を固定し単語だけを変化させた場合

次に、単語によって画像変換が変わるか、ということを観察するための実験を行った。図 10 は絵画スタイルで学習した単語についての変換を、図 11 は質感画像スタイルで学習した単語についての変換を行った様子を示す。入力単語を変えることで変換の方向性を制御することができたことがわかる。

図 10: 絵画スタイルで画像を変換した様子

図 11: 質感画像スタイルで画像を変換した様子

次に、上手く変換が行えなかった場合について示す。言葉毎の変換の違いがほぼ出ておらず、どの変換でも同じような画像が出力されていたり、スタイル変換を上手く学習できずにコンテンツ画像を崩壊させるような変換になってしまっている事がわかる。

図 12: 絵画スタイルでの画像変換の失敗例

図 13: 質感画像スタイルでの画像変換の失敗例

6. 考察

上手く変換できた例とできなかった例を見比べると、建物のような画像では上手く変換できてい。これは VGG 画像認識によって画像の特徴を認識したからだと考える。VGG 画像認識は画像に写っている物体についての認識をするものであり、風景などの “シーン” を判別することは難しい。これによって “猫” や “いちご” などの物体は認識できても “海原” や “ビル” などの背景になるような画像は認識できないという減少が起こる。これを解決するためには VGG 画像認識によって作成しているベクトルを、シーン認識のネットワークを使用して生成することによって解決することが出来ると考えられる。

また、“スケッチ” のような画像の見た目特徴を示す単語では変換できるが “フラット” のような概念的な単語では上手く変換できていない。学習画像を各カテゴリ 50 枚程度のデータセットと小規模なものを使用してスタイル変換を学習したために、対応できるスタイルに偏りが出てしまっているために、画像変換を行うと元画像の意味を完全に無視し、画像として成り立っていないような変換を行ってしまうことがある。これは更に大規模でノイズの少ないデータセットを用意することができれば解決することができると考えられる。

7. Conclusion

本提案手法によって、“言葉による画像変換” と “コンテンツ画像に適した画像変換” の 2 つの目的は達成できた。特に VGG 画像認識で上手く認識できるような、画像の中心にコンテンツが表示されている画像に関しては良い結果が得られて、逆に背景画像のような物体認識の行えない画像に関しては上手く行かなかった。

今後はシーン認識によって背景を示す画像に対応すること、より大きなデータセットの構築や、ユーザー評価を行いより客観的な結果の考察などを行っていきたい。

参考文献

[Gatys 16] Gatys, L. A., Ecker, A. S., and Bethge., M.: Image style transfer using convolutional neural networks, in *Proc.of IEEE Computer Vision and Pattern Recognition*, pp. 2414–2423 (2016)

[Johnson 16] Johnson, J., Alahi, A., and Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution, in *Proc.of European Conference on Computer Vision*, pp. 694–711 (2016)

[Liao 17] Liao, J., Yao, Y., Yuan, L., Hua, G., and Bing, S. K.: Visual Attribute Transfer through Deep Image Analogy, in *arXiv:1705.01088* (2017)

[Mikolov 13] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean., J.: Distributed representations of words and phrases and their compositionality, in *Proc.of Advances in Neural Information Processing Systems 25*, pp. 3111–3119 (2013)

[Simonyan 15] Simonyan, K., Vedaldi, A., and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, in *Proc.of International Conference on Learning Representation* (2015)

[Yanai 17] Yanai, K.: Unseen Style Transfer Based on a Conditional Fast Style Transfer Network, in *Proc.of International Conference on Learning Representation Workshop Track (ICLR WS)* (2017)

謝辞 本研究はJSPS 科研費15H05915, 17H01745, 17H05972, 17H06026, 17H06100 の助成を受けたものです.