1P-02

畳み込みニューラルネットワークによる特定部分へのスタイル転移

新家　歩†　　グエン　トゥン‡　原田　智広†　ターウォンマット　ラック†
知能エンターテインメント研究室
立命館大学情報理工学部知能情報学科†　　立命館大学情報理工学研究科人間情報科学コース‡

概要–画像内の特定のオブジェクトへのスタイル転移を提案する．画像分類の分野で高い性能を示した畳み込みニューラルネットワークにVGG-19と呼ばれるネットワークがある．学習済みVGG-19の特定の層のフィルタによって出力される特徴マップを利用し，一方の画像のコンテンツ情報を保ちつつ，もう一方の画像のスタイル情報を合わせ持った画像を生成することができるスタイル転移と呼ばれる研究がある．本研究では学習済みのVGG-19の各フィルタを最大化する画像を出力することで,どのフィルタがどのようなオブジェクトに対し反応するかを確認し，スタイル転移させたい特定のオブジェクトに対して大きな反応を示したフィルタのみを使用した目的関数を新たに設定することで，対象のオブジェクトへのスタイル転移を行った．

1 .はじめに

近年画像認識の分野で高い性能を示した畳み込みニューラルネットワーク（CNN）にVGG-19と呼ばれるネットワークがある[1]．この学習済みVGG-19の中間層の出力を利用したスタイル転移という研究が存在する．スタイル転移とはGatysらが提案したアルゴリズムで，コンテンツ自動生成の研究分野の一つである[2]．スタイル転移ではコンテンツ画像とスタイル画像の二つの画像を入力とし，コンテンツ画像に書かれた物体（コンテンツ）に対し，スタイル画像の画風（スタイル）を転移した画像を生成する．

Gatysらの手法では画像全体にスタイル転移を行うことを目的としているため,画像内の特定の物体へのスタイル転移を行うことが出来ない（図1）．そこで本論文では従来のスタイル転移手法をベースに画像内の特定の物体へのスタイル転移を実現する．

2 .関連研究

Gatysらが提案した手法では，まず事前に学習したVGG-19にコンテンツ画像とスタイル画像を入力し，コンテンツ画像のコンテンツ表現とスタイル画像のスタイル表現をそれぞれ特定の層から抽出する．次に生成された画像をVGG-19に入力し，生成画像のコンテンツ表現とスタイル表現を抽出する．そして，コンテンツ画像のコンテンツ表現と生成画像のコンテンツ表現の差と，スタイル画像のスタイル表現と生成画像のスタイル表現の差を，最小化するように生成された画像のピクセルを調整することで，スタイル転移を実現している．

図1：Gatysらの手法により生成された画像の例 左：コンテンツ画像 中：スタイル画像 右：生成画像

このとき，コンテンツ表現とは中間層の出力である特徴マップを指す．コンテンツ表現同士の差のことをcontent lossと言い，以下の式で表される．

$$\mathcal{L}_{\text{content}}(x_c, x) = \frac{1}{N^l M^l}\sum_{i,j}(F_{ij}^l(x_c) - F_{ij}^l(x))^2 \quad (1)$$

ここでx_c，xはそれぞれコンテンツ画像と生成された画像を示し，lは特定の層を表す．$F^l(x_c)$，$F^l(x)$は特定の層lにおけるx_c，xそれぞれのコンテンツ表現である．またN^l，M^lはそれぞれ層lにおける特徴マップの数と特徴マップのサイズを表す．

一方で，スタイル表現は同じ層のコンテンツ表現の内積であり，以下の式で表されるグラム行列である．

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l \quad (2)$$

ここでF_i^l，F_j^lはそれぞれ特定の層lにおける特徴マップを示し，kは特徴マップの各要素を表す．スタイル表現同士の差のことをstyle lossと言い，ある層におけるstyle lossは以下の式で表される．

$$E_l = \frac{1}{N_l^2 M_l^2}\sum_{i,j}(G_{i,j}^l(x_s) - G_{i,j}^l(x))^2 \quad (3)$$

ここでx_sはスタイル画像，$G_{i,j}^l(x_s)$，$G_{i,j}^l(x)$はそれぞれある層lにおけるx_s，xの前記のスタイル表現を表す．該当する全層のstyle lossの和を取る最終的なstyle lossは

$$\mathcal{L}_{\text{style}(x_s,x)} = \sum_{l=1}^{L} w_l E_l \quad (4)$$

で表される．w_lはE_lに対する重みである．以上のcontent lossとstyle lossより目的関数は以下の式で表される．

$$L_{total}(x_c, x_s, x) = \alpha L_{content}(x_c, x) + \beta L_{style}(x_s, x) + \gamma L_{tv}(x) \quad (5)$$

ここでα，β，γは各lossに対応する重みである．L_{tv}はトータルバリエーション正則化項と呼ばれる正則化項であり，以下の式で表される．

$$\mathcal{L}_{tv}(x) = \sum_{i,j}\left(\left(x_{i+1,j} - x_{i,j}\right)^2 + \left(x_{i,j+1} - x_{i,j}\right)^2\right) \quad (6)$$

ここで$x_{i,j}$は生成画像の各ピクセルを表す．

3 .提案手法

既存の手法では入力画像に対して特定の層のすべてのフィルタを適用したときのコンテンツ表現とスタイル表現を使用している．そのため生成画像の背景にもスタイル転移がされる．本手法では背景などの，転移させたいオブジェクト以外のものにはスタイル転移を行わないようにする必要がある．そこで転移させたいオブジェクトに対して反応が大きいフィルタを選択し，style loss を新たに設定する．

フィルタの選択には，Yosinski らが公開している Deep Visualization Toolbox を使用する[3]．Deep Visualization Toolbox は複数枚の入力画像から，ネットワークの各フィルタに対して反応を最大化させる画像を，活性度の大きい順に任意の枚出力することができる．今回は入力画像に ILSVRC2012 のテストデータセット画像約 5 万枚を使用し，各フィルタを最大化させる画像 9 枚を出力させた．このように出力させた 9 枚の画像の中で，9 枚中 9 枚すべてが転移させたいオブジェクトであることを目視で確認できたものを利用する．

このようにして選択されたフィルタを使用した新たなグラム行列$\mathcal{G}$は以下の式で表される．

$$\mathcal{G}_{ij} = \begin{cases} F_i(x_c) \cdot F_j(x_c) & i \notin \mathcal{M} \text{ and } j \notin \mathcal{M} \\ \mathcal{F}_i(x_s) \cdot \mathcal{F}_j(x_s) & i \in \mathcal{M} \text{ or } j \in \mathcal{M} \end{cases} \quad (7)$$

ここで$\mathcal{M}$は選択されたフィルタによって出力される特徴マップのセットを表す. このグラム行列$\mathcal{G}$を導入した style loss は以下のようになる．

$$E_l = k_1 \sum_{\substack{i \notin \mathcal{M} \\ and \\ j \notin \mathcal{M}}} \left(G^l_{ij}(x) - \mathcal{G}^l_{ij}\right)^2 + k_2 \sum_{\substack{i \in \mathcal{M} \\ or \\ j \in \mathcal{M}}} \left(G^l_{ij}(x) - \mathcal{G}^l_{ij}\right)^2 \quad (8)$$

ここで$G^l_{i,j}(x)$は層lにおけるxのスタイル表現，k_1，k_2は各項の該当するフィルタ数に比例する重みでありそれぞれ以下の値を取る．

$$k_1 = \frac{1}{1-r} \ , \ k_2 = \frac{1}{r} \quad (9)$$

またrは以下の式で定義される．

$$r = \frac{2 \times |\mathcal{M}| \times N^l - |\mathcal{M}| \times |\mathcal{M}|}{N^l \times N^l} \quad (10)$$

4 .実験

本実験では犬が映った画像を対象として，画像内の犬に対してのみスタイル転移を行う．提案手法でフィルタの選別を行った結果,conv4_1 が 6 枚,conv5_1 が 16 枚となった．既存手法ではスタイル表現の抽出に，conv1_1，conv2_1，conv3_1 のフィルタの出力も使用しているが，犬を視認できる画像が，Deep Visualization Toolbox の出力において確認できなかったため使用しなかった．また既存の手法では conv4_1，conv5_1 はそれぞれ 512 枚のフィルタすべてを使用している．content loss の重みは 5.0，style loss の重みは 0.002，正則化項の重みを 0.001，style loss における各層の重みは 0.5 とし，最適化のイテレーション数を 1000 とした。また最適化手法には L-BFGS を使用した．

図 2：使用したコンテンツ画像

図 3：実験結果
上段：使用したスタイル画像　中段，下段：図 2 コンテンツ画像と上段のスタイル画像を入力とした際の生成画像

5 .結果と今後の課題

図 3 の結果より 1 枚目のコンテンツ画像において，画像内の犬にのみスタイルが転移され，背景にはスタイルが転移されていないことが確認できる．一方で，2 枚目のコンテンツ画像に対しては背景へのスタイルの転移は見られないが，犬全体にスタイルが転移されず，犬の一部にのみスタイルが転移されている．これは Deep Visualization Toolbox の出力結果を目視で確認しているため，適当なフィルタを選別出来ていないという可能性があげられる．今後の課題として，スタイル転移させたいオブジェクト毎のフィルタの選別を機械的に行うことで，より適当なフィルタの選別が可能になり，精度の高い特定部分へのスタイル転移を行うことが出来るようになると考えられる．

参考文献

[1] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).
[2] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." *arXiv preprint arXiv:1508.06576* (2015).
[3] Yosinski, Jason, et al. "Understanding neural networks through deep visualization." arXiv preprint arXiv:1506.06579 (2015).