Abstract

우리가 흔히 아는 Image transformation task는 noisy, low-resolution, grayscal의 degraded image를 denoising, super-resolution, colorization과 같은 image processing을 취하여 최종적으로 높은 퀄리티의 color image를 추출하는 것이다. 이번에 알아볼 주제는 style transfer로 하나의 이미지가 있을 때, 그 이미지의 스타일을 다른 스타일로 바꾸는 것이다. 본래의 style transfer는 texture transfer로 여겨졌으며, target image의 content의 source image의 texture를 합성하는 것이였다. 하지만, 이 주제의 모든 연구는 target image의 low level feature만을 사용했다는 것에 한계를 갖는다. 반면 style transfer는 Deep Convolutional Neural Network를 활용하여 high level feature를 추출하고 이를 활용하여 image transformation task를 수행한다. 아래의 사진은 texture transfer [1]의 연구결과로 추후 다룰 연구와 확연하게 perceptual한 부분에서 차이가 느껴질 수 있다.

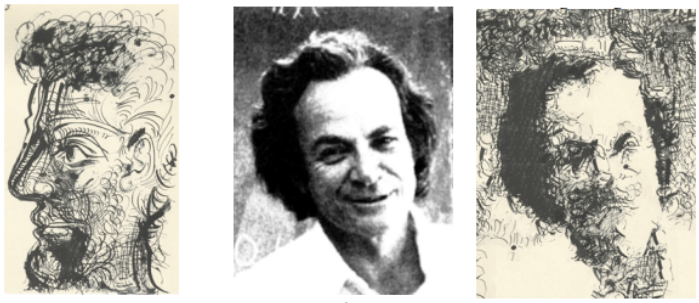

Figure 1: Result of texture transfer

1 First Style Transfer

최초로 딥러닝을 활용하여 Style transfer이 제안된 연구는 [2]이다. Style transfer는 한 image의 content 정보를 유지하고 다른 이미지의 스타일 정보를 가져와서 반영하는 것으로, 스타일을 전송하는 것과 같다. 이를 위한 주요 방법론은 다음과 같다. style transfer는 Pre-traninded CNN 모델이 사용된다. 이를 활용하여 content를 유지할 이미지와 적용하고자 하는 style이 담긴 이미지의 high level feature를 추출해 낼 수 있다. 이때 임의의 이미지를 noise값으로 초기화 한 뒤 앞서 추출한 feature map에 대하여 loss function을 정의한 뒤 이를 optimization함으로써 style transfer를 수행한다.

2 Style transfer Algorithm

2.1 prerequisite

Style transfer method에 대해 알아보기 이전에 cnn에서 추출되는 high-level feature과 low-level feature에 대해 이해할 필요가 있다. cnn의 더 깊은 layer에서 추출되는 feature를 high-level feature라고 한다. 먼저, low-level feature 일수록 이미지의 minor details, 즉 세부적인 내용을 담고 있다고 할 수 있다. 점과 선 같은, 즉 edges와 같은 추상적인 정보를 많이 담고 있다. 반면, high-level는 이미지 내의 더 큰 모양들과 물체들을 감지한다. 즉 content의 정보를 앞쪽 layer에서 더 잘 보존할 수 있으며, high level일수록 color, texture, exact shape은 유지되지 않고 전체적인 공간 정보가 보존된다. 따라서, content정보와 style정보를 추출하는 것은 다른 layer에서의 feature map을 사용할 것으로 추측할 수 있다.

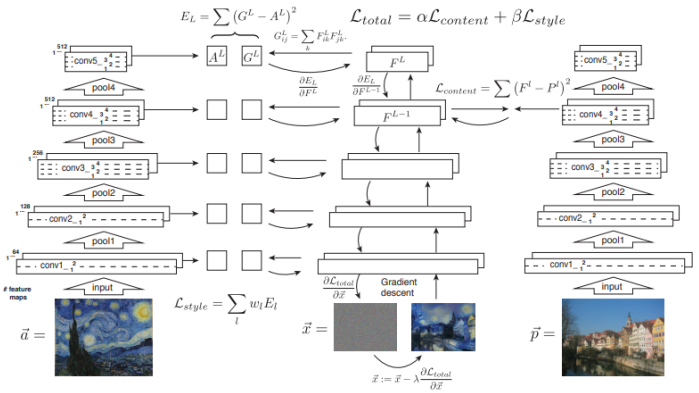

Figure 2: Style transfer flowchart

2.2 Content Loss

Content loss는 두 이미지의 activation 값이 동일하도록 만들어내기 위하여 정의된다. Figure2에서 noise image를 왼쪽 위부터 오른 쪽아래까지 하나의 vector $\vec{x}$, Content Image를 $\vec{p}$로 나타내었다. 이미지 $\vec{x}$를 CNN 모델에 넣었을 때, l번 째 layer에서의 output인 feature map을 F^l, content image $\vec{p}$의 l번 째 feature map을 P^l이라 한다. i를 feature map의 channel numbering, j를 각 feature map의 activation value의 위치라 할때, Content loss는 다음과 같이 정의된다.

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

2.3 Style Loss

이 논문에서는 stlye은 서로다른 feature의 correlation(Style Represetation)으로 정의한다. 이 correlation은 Gram Matrix로 나타나는데, 이는 다음과 같다.

$$G_{ij} = \sum_k F_{ik}F_{jk} \tag{2}$$

(2)의 방정식에서 k는 activation value의 위치를 의미하며 i, j는 서로다른 feature를 의미한다.
style loss는 두 이미지의 feature의 correlation을 유사하게 만들어주기 위해 사용된다. 즉, 두 이미지의 Gram matrix가 유사해질 수 있도록, 스타일 정보가 유사해질 수 있도록 style loss를 업데이트 하게 된다. 식으로 나타내면 아래와 같다.

$$L_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l \tag{3}$$

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

E_l은 두 이미지의 Gram Matrix가 유사해질 수 있도록 정의된 L2 norm 이며, N_l, M_l은 normalization term이다. style loss function에서L은 Gram matrix를 구하는 layer의 개수를 의미하고 w_l은 해당 layer의 가중치이다.

2.4 Difference between Content loss and Style Loss

content loss는 하나의 layer에서 feature map 자체가 같아질 수 있도록 하는 반면, style loss는 여러 layer (Figure 2에서는 5개의 layer)에서 모두 Gram matrix를 구하고 각 layer에서 Gram Matrix의 L2 norm을 구하여 각 layer에 가중치를 준 후 이를 합하여 구한다. 또한, layer가 깊어질수록 구체적인 픽셀 정보가 소실되므로 Content loss는 앞쪽에 있는 layer에 대해서 정의된다. 반면 style loss는 깊은 layer까지 같이 사용하였을 때, 해당 style image의 style이 더 잘 담길 수 있다. 이때, style loss에서 해당 layer의 feature의 개수에 따라 Gram Matrix의 크기가 결정된다.

2.5 Total Loss

이미지 $\vec{x}$를 업데이트 하기 위하여 content loss와 style loss를 통해 loss function을 다음과 같이 정의한다.

$$L_{total} = \alpha L_{content} \beta L_{style} \tag{5}$$

이렇게 구해진 total loss를 이용하여 backpropagation를 반복하는 것으로 이미지를 만들어 낸다. 이때 backpropagation에서 사용하는 gradient는 total loss를 $\vec{x}$로 미분하여 얻어낸다.

2.6 Conclusion

loss function의 hyperparameter를 다른 값으로 설정하여, content와 style 중 어느 쪽에 더 비중을 둘지 정할 수 있어 다양한 style transfer를 시도할 수 있다. 이 논문의 style transfer는 backprop를 통해 직접 이미지를 업데이트를 하는 방식으로 이미지를 만들어내는 과정에서 속도가 정해지며, 512*512 해상도의 이미지를 생성하는데 1시간까지 걸릴정도로 속도적으로 한계가 있다.

3 make use of Perceptual Loss for style transfer

앞서 소개한 연구의 Style transfer의 속도적 한계를 해결하기 위해 backprop가 없이 하나의 forward과정으로 이미지를 생성해내는 feed-forward network를 이용하는 방식이 고안되었다.
[3]가 연구되기 이전 image transformation problems에 대해서 per-pixel loss function을 사용한 feed-forward cnn이 사용되었다. 이 방법의 장점은 한 번의 forward pass를 통해 빠른시간에 Inference(학습을 통해 만들어진 모델을 실제로 새로운 입력 데이터에 적용하여 결과를 내놓는 것으로, 현재 데이터에 대해서 해당 모델이 원하는 작업을 수행해 주는 것)를 할 수 있다. 하지만, 이 방법은 원본이미지와 output의 차이를 각 픽셀 값으로 비교한 것으로, ground-truth와 prediction 사이 perceptual difference를 잡아낼 수 없다. 즉 이미지 전체에 대한 인식이 고려되지 않아 동일한 이미지가 한 픽셀 단위만 움직여도 다른이미지로 판단하여 신뢰도가 낮아지는 단점이 존재한다.
이미지 전체에 대한 인식을 고려하기 위한 perceptual loss function은 pre-trained CNN에서 추출된 high-level image feature representations의 차이에 기반하여 높은 퀄리티의 이미지를 생성하지만, 이는 inference에서 최적화 문제를 풀어야함에 있어 시간이 오래걸린다. 이에 따라 연구에서는 어떤 방식을 취하는지 자세히 알아본다.

3.1 Method

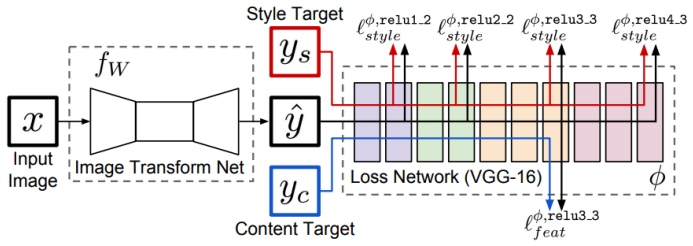

Figure 3: Perceptual network flowchart

이 연구에서는 두 접근법의 이점을 결합하여, feed-forward transformation network를 학습하는데 있어, perceptual loss function을 사용한다. 이에 따라 per-pixel loss 보다 더욱 robustly하고 유사한 이미지를 얻을 수 있다. 또한 이는 속도적으로 test-time에서 real-time으로 동작한다. 이 연구에서 제안하는 Networks architecture는 image transformation networ와 loss network로 구성되어 있다. content target을 y_c, style target을 y_s로 할때, style transfer에서 content target y_c는 input image인 x이며, output image $\hat{y}$은 $x = y_c$의 content에 style y_s를 결합한 것이다.

3.2 Image Transformation Network

Image Transformation Network는 style transfer를 위해 입력과 출력이미지의 resolution을 동일하게 한다. 이를 위해 downsampling과 upsampling이 각각 입력단과 출력단 쪽에서 진행된다. 이때, downsampling과 upsampling을 위해 어떠한 pooling layers를 사용하지 않고 strided and fractionally strided convolutions만을 이용한다. 이는 computational cost와 receptive field에 이점이 있다. input을 down sampling하기 위해 두 번의 stride-2인 convolutions을 취하며 output 추출하기 전, stride-$\frac{1}{2}$인 convolution이 수행된다.
downsampling과 upsampling을 수행하는 convolution layer 사이에 5개의 residual blocks가 있다. 이는 출력이미지와 입력이미지의 구조가 share될 수 있기 때문에 사용된다.
처음과 마지막 layer는 9*9 kernel이며, 나머지 layer는 3*3 kernel이다. non-residual convolution layers는 output layer를 제외하고 spatial batch bormalization과 ReLU를 사용한다. 이때, output layer에서는 출력 이미지의 pixel 범위가 [0, 255]로 하기 위해서 scaled tanh를 사용한다.

3.3 Loss network

Loss Network에서는 images의 high-level perceptual, semantic 차이를 측정하기 위해 두 perceptual loss functions를 정의한다. 이때, 이미 학습된 네트워크인 VGG-16을 loss network로 활용한다.
Feature Reconstruction Loss는 아래와 같이 정의된다.

$$l^{\phi,j}_{feat}(\hat{y}, y) = \frac{1}{C_j H_j W_j} ||\phi_j(\hat{y}) - \phi_j(y)||_2^2 \tag{6}$$

두 feature representations 사이 squared, normalized Euclidean distance를 사용한다. ϕ_j는 j번 째 convolution layer의 feature map, $C_j H_j W_j$는 feature map의 shape 을 의미한다.
Style Reconstruction Loss는 [2]와 동일한 gram matrix를 사용하며, target images와 output의 gram matrice 사이 suqared fronebius norm으로 계산된다. style loss는 아래와 같다.

$$l^{\phi,j}_{style}(\hat{y}, y) = ||G^{\phi}_j(\hat{y}) - G^{\phi}_j(y)||_F^2 \tag{7}$$

$$G^{\phi}_j(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'} \tag{8}$$

추가적으로, gram matrix는 channel에 영향을 받기 때문에 target과 output의 이미지 사이즈가 달라도 잘 정의된다.

3.4 Generated image $\hat{y}$

feature와 style reconstruction을 진행하여 생선되는 이미지는 최종적으로 아래와 같은 식으로 구해진다.

$$\hat{y} = \arg\min_y \lambda_c \ell^{\phi,j}_{feat}(y, y_c) + \lambda_s \ell^{\phi,J}_{style}(y, y_s) + \lambda_{TV} \ell_{TV}(y)$$

4 Conclusion

style transfer는 한 이미지에 다른 이미지의 style을 추출하여 입히는 것으로, 다양한 이미지를 생성해낼 수 있다. 또한, content loss function과 style loss function에 각각 다른 가중치를 주면서 같은 style image에서도 다양한 image를 생성해 낼 수 있다. 이렇게 생성된 이미지는 예술적으로 보여질 수 있으며, Photorealistic하게 보이기도 한다. 이러한 인공지능 모델은 창작·예술 분야에서 유효하게 쓰일 것으로 생각된다.

[1] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001.

[2] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[3] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.