

# Neural Style Transfer

## 1. Style Transfer이란?

P라는 이미지의 스타일을 A 이미지의 스타일로 바꾸어 새로운 이미 X를 생성하는 것



## 2. Style Transfer 방식

① pre-trained된 모델을 기반으로 content image와 style image을 입력으로 이용해 이미지를 학습하는 방법

- 장점
  - 이미지 2장(content image와 style image)만으로 style transfer가 가능
- 단점
  - 새로운 content 이미지를 변환하거나, 또는 새로운 style로 변환할 경우 다시 학습해야 함

② pre-trained된 모델을 기반으로 content image와 style image을 입력으로 이용해 이미지 변환 네트워크를 학습하는 방법

- 장점
  - 이미지 2장(content image와 style image)만으로 style transfer가 가능
  - 새로운 content 이미지를 변환할 경우, 재학습이 필요 없고 실시간 변환이 가능
- 단점
  - style image는 대체로 그림으로 한정되는 듯.

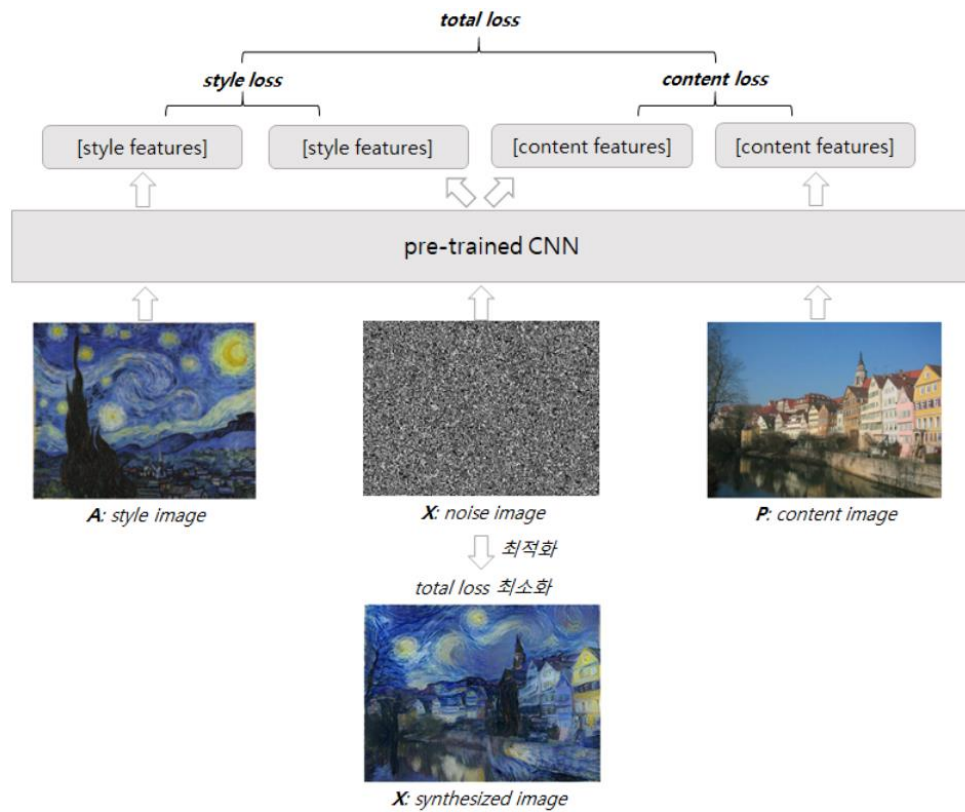
③ GAN을 기반으로 style transfer 모델을 직접 학습하는 방법

- 장점
  - 새로운 content 이미지를 변환할 때, 재학습이 필요 없고 실시간 변환이 가능
  - style image에 제한이 없음
- 단점
  - 학습을 위한 많은 데이터셋 확보가 필수적일 것

Style transfer의 시초가 된 ①번 방식에 대해 알아보자

### 3. 알고리즘 도식화

미리 학습된 CNN 모델을 통해 content image와 style image에서 각각 content, style feature를 뽑아내고 이를 기본 이미지 X(noise image)의 content, style feature와 비교해 loss를 계산하고 이를 최소화하는 방식으로 진행



#### 4. pre-trained CNN은 어떤 모델??

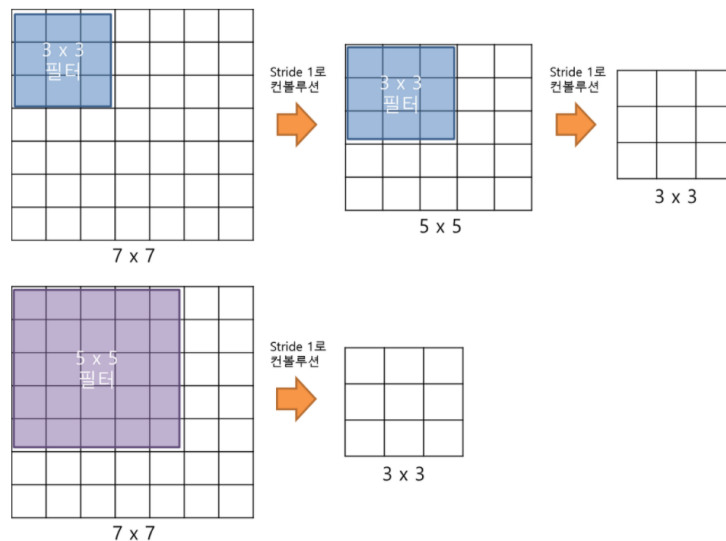
최초 style transfer를 다룬 논문에 따르면 VGG-19라는 CNN 모델을 활용한다.

※ VGG-19 (정규화, average pooling 적용)

- 옥스퍼드 대학의 연구팀 VGG에서 개발(19개의 층을 쌓았다고 해서 VGG-19)

ConvNet Configuration						NAME:
A	A-LRN	B	C	D	E	
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers	
input (224 × 224 RGB image)						
conv3-64	conv3-64 LRN	conv3-64 <b>conv3-64</b>	conv3-64	conv3-64	conv3-64	conv1_1
			conv3-64	conv3-64	conv3-64	conv1_2
maxpool						pool1
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	conv3-128	conv3-128	conv3-128	conv2_1
			conv3-128	conv3-128	conv3-128	conv2_2
maxpool						pool2
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3_1
conv3-256	conv3-256	conv3-256	conv3-256 <b>conv1-256</b>	conv3-256 <b>conv3-256</b>	conv3-256	conv3_2
					conv3-256	conv3_3
					<b>conv3-256</b>	conv3_4
maxpool						pool3
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv4_1
conv3-512	conv3-512	conv3-512	conv3-512 <b>conv1-512</b>	conv3-512 <b>conv3-512</b>	conv3-512	conv4_2
					conv3-512	conv4_3
					<b>conv3-512</b>	conv4_4
maxpool						pool4
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv5_1
conv3-512	conv3-512	conv3-512	conv3-512 <b>conv1-512</b>	conv3-512 <b>conv3-512</b>	conv3-512	conv5_2
					conv3-512	conv5_3
					<b>conv3-512</b>	conv5_4
maxpool						pool5
FC-4096						fc6
FC-4096						fc7
FC-1000						fc8
soft-max						

- 필터의 크기와 층 깊이: 필터 크기가 작을수록 층을 깊게 쌓을 수 있다. (필터 크기가 클수록 이미지 축소가 심하게 됨) → 필터를 작게 해서 층을 깊게 쌓는 것이 학습 파라미터 수를 줄여 속도를 높여주고 비선형성을 증가시켜 성능이 좋아진다.



그렇다면 19개의 층 중에 어느 레이어의 feature map을 사용할까?

## 5. pre-trained CNN 모델에서 사용 feature map의 layer

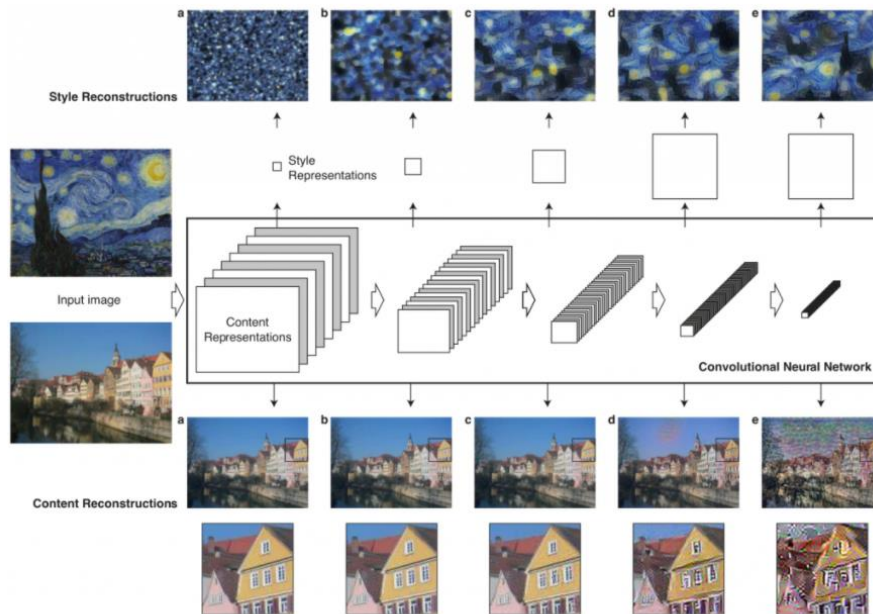


그림. 레이어별 feature

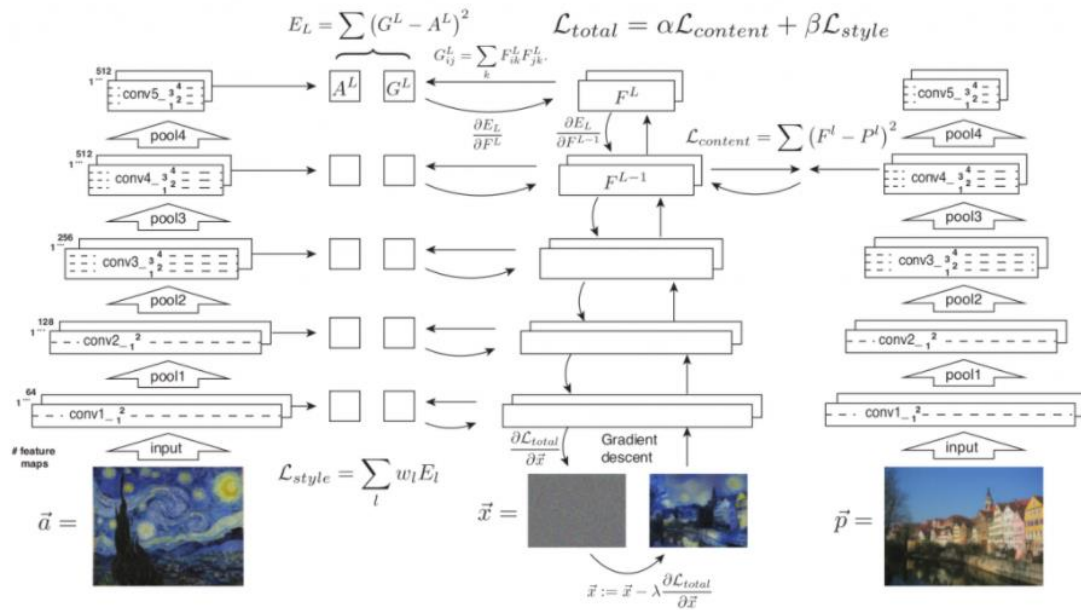
### ① content feature

layer가 깊어질수록 pixel들이 일그러져 style의 특성은 사라지고 대략적인 형태만 남음(style이 사라지고 content만 남음) → 깊은 layer의 feature map을 사용

### ② style feature

style feature는 공간적 정보에 기반하는 style이 아니라 그림 전체적으로 퍼져 있는 느낌을 살려야 함(건물의 특징, 다리의 특징에 따른 style 반영 X) → 이를 위해서 각 layer의 feature map 상관관계를 나타내는 Gram matrix를 사용 → 다수의 스케일을 고려한 style 추출이 가능

## 6. 구체적인 style transfer 알고리즘 도식화



- ① content image p, style image a에 대해, 합성할 이미지인 x를 noise image로 초기화한다.
- ② 각 이미지 p, a, x를 신경망을 통해 feed forward 한다.
- ③ p와 x에 대해서는 content feature 기반의 content loss를 계산한다.
- ④ a와 x에 대해서는 style feature 기반의 style loss를 계산한다.
- ⑤ content loss와 style loss를 합하여 total loss를 계산한다.
- ⑥ total loss를 back propagation을 통해 noise image x를 업데이트한다.

## 7. Loss 계산

### ① Content loss

$$\mathcal{L}_{\text{content}}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2$$

l층의 layer 전체 feature map 값의 차이(F는 content image, P는 noise image)

### ② Style loss

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2, \text{ where } N_l \text{ is \# of feature maps at layer } l$$

$M_l \text{ is height} \times \text{width of feature maps at layer } l$

$$\mathcal{L}_{\text{style}}(\vec{a}, \vec{x}) = \sum_{l=0}^L w_l E_l, \text{ where } w_l \text{ is weighting factors of the layer to the total loss}$$

w 가중치의 합은 언제나 0(어느 레이어를 많이 사용할 것인지 결정)

### ③ Total loss

$$\mathcal{L}_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{\text{style}}(\vec{a}, \vec{x})$$



$\alpha$ ,  $\beta$ 는 우리가 임의로 정하는 값이고 그 값의 차이에 따라 content가 얼마나 잘 유지되는지 style을 얼마나 더 중요시할 것인지 결정할 수 있다.

## 8. 출처

<https://www.popit.kr/neural-style-transfer-%EB%94%B0%EB%9D%BC%ED%95%98%EA%B8%B0/>  
(Neural Style Transfer)

<https://kyounju.tistory.com/3> (Neural Style Transfer)

<https://bskyvision.com/504> (VGG-19)

<http://blog.naver.com/PostView.nhn?blogId=atelierjpro&logNo=221180412283> (Gram matrix)