Stylization

2019~2020 논문 구조 요약

201810977 이정인

Contents

01 Collaborative Distillation for Ultra-Resolution Universal Style Transfer

02 Two-Stage Peer-Regularized Feature Recombination for Arbitrary Image Style Transfer

03 SRM: A Style-Based Recalibration Module for Convolutional Neural Networks

04 Multimodal Style Transfer via Graph Cuts

05 Total Style Transfer With A Single Feed-forward Network

06 ETNet: Error Transition Network for Arbitrary Style Transfer

Contents

07 MetaStyle : Three-Way Trade-off among Speed, Flexibility, and Quality in Neural Style Transfer

08 Attention-Aware Multi-Stroke Style Transfer

09 Learning Linear Transformations for Fast Image and Video Style Transfer

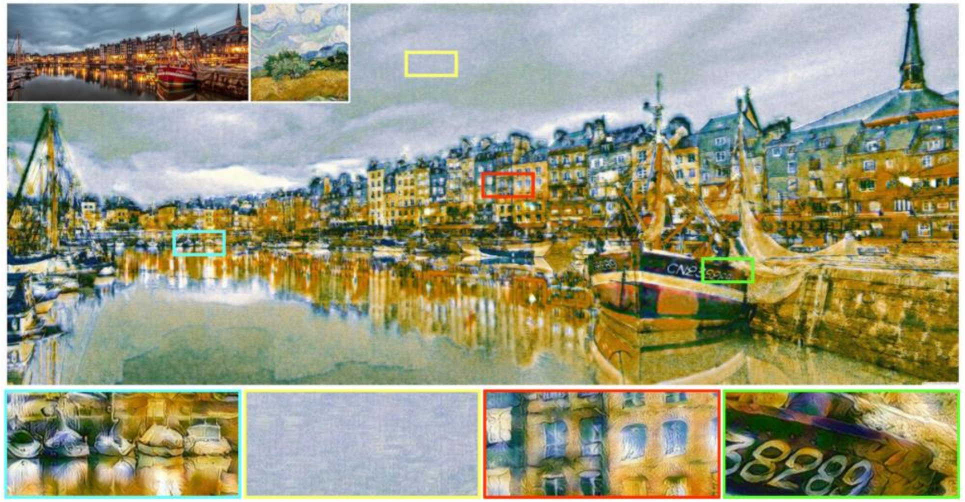
10 Arbitrary Style Transfer With Style-Attentional Networks

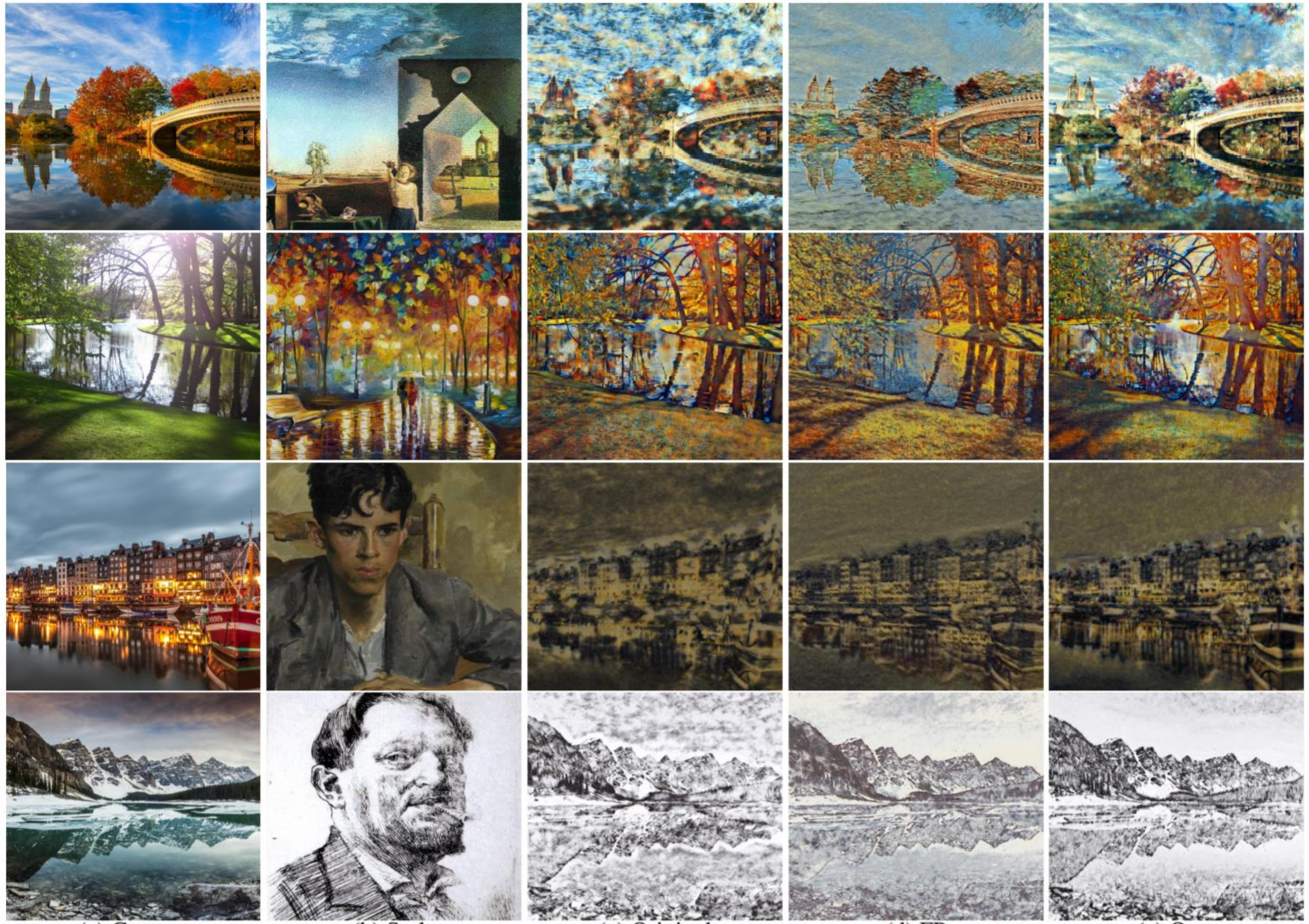
11 A Flexible Convolutional Solver for Fast Style Transfers

12 A Content Transformation Block for Image Style Transfer

01 Collaborative Distillation for Ultra-Resolution Universal Style Transfer

Huan Wang, Yijun Li, Yuehai Wang, Haoji Hu, Ming-Hsuan Yang, CVPR, 2020

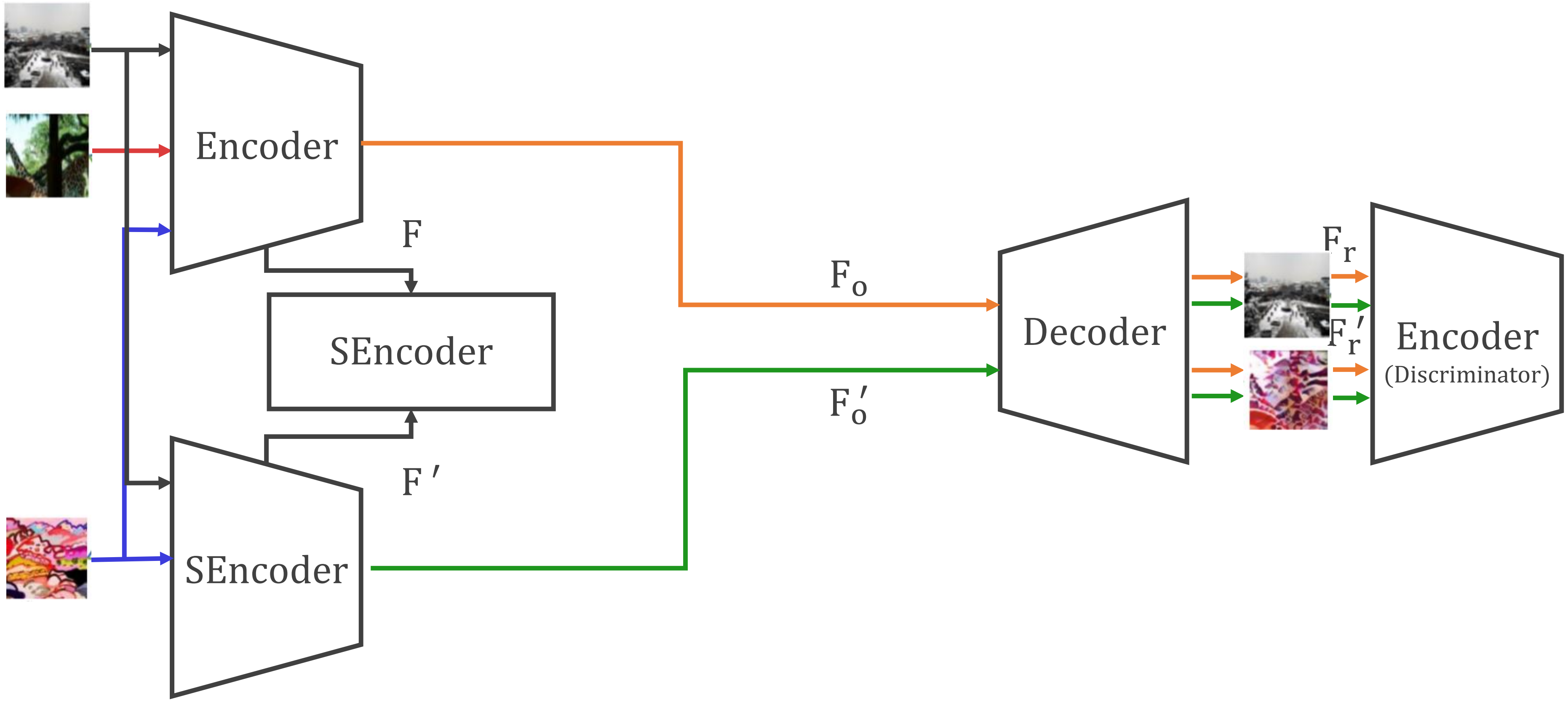




(a) Content (b) Style (c) Original (d) FP (e) Ours

01 Collaborative Distillation for Ultra-Resolution Universal Style Transfer

Huan Wang, Yijun Li, Yuehai Wang, Haoji Hu, Ming-Hsuan Yang, CVPR, 2020



SEncoder : model의 크기를 줄인 Small Encoder, 수용할 수 있는 해상도를 크게 하여 고해상도의 스타일 변환을 시도

01 Collaborative Distillation for Ultra-Resolution Universal Style Transfer

Huan Wang, Yijun Li, Yuehai Wang, Haoji Hu, Ming-Hsuan Yang, CVPR, 2020

$$\boxed{L_{total} = \beta \sum_{i=1}^{k} L_{embed} + L_{collab}}$$

L_{embed} : feature domain information (for small encoder)

L_{collab} : for image reconstruction

01 Collaborative Distillation for Ultra-Resolution Universal Style Transfer

Huan Wang, Yijun Li, Yuehai Wang, Haoji Hu, Ming-Hsuan Yang, CVPR, 2020

$$L_{total} = \beta \sum_{i=1}^{k} L_{embed} + L_{collab}$$

$$\mathcal{L}_r^{(k)} = \|\mathcal{I}_r - \mathcal{I}_o\|_2^2 + \lambda_p \sum_{i=1}^{k} \|\mathcal{F}_r^{(i)} - \mathcal{F}_o^{(i)}\|_2^2, \quad (1)$$

$$\mathcal{L}_{\text{st}} = \|\mathcal{F}_{\text{st}}^{(4)} - \mathcal{F}_c^{(4)}\|_2^2 + \lambda_s \sum_{i=1}^{4} \|\mathcal{G}_{\text{st}}^{(i)} - \mathcal{G}_s^{(i)}\|_2^2, \quad (2)$$

$$\mathcal{G} = \mathcal{F} \cdot \mathcal{F}^T, \quad (3)$$

$$\mathcal{F} = Q \cdot \mathcal{F}', \quad (4)$$

$$\mathcal{L}_{\text{embed}} = \|\mathcal{F} - Q \cdot \mathcal{F}'\|_2^2. \quad (5)$$

Huan Wang, Yijun Li, Yuehai Wang, Haoji Hu, Ming-Hsuan Yang, CVPR, 2020

01 Collaborative Distillation for Ultra-Resolution Universal Style Transfer

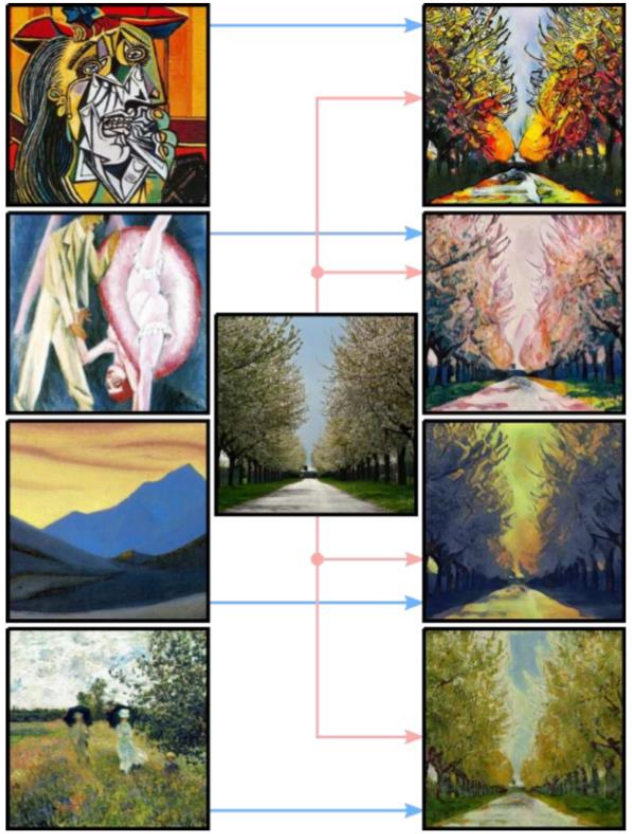
Huan Wang, Yijun Li, Yuehai Wang, Haoji Hu, Ming-Hsuan Yang, CVPR, 2020

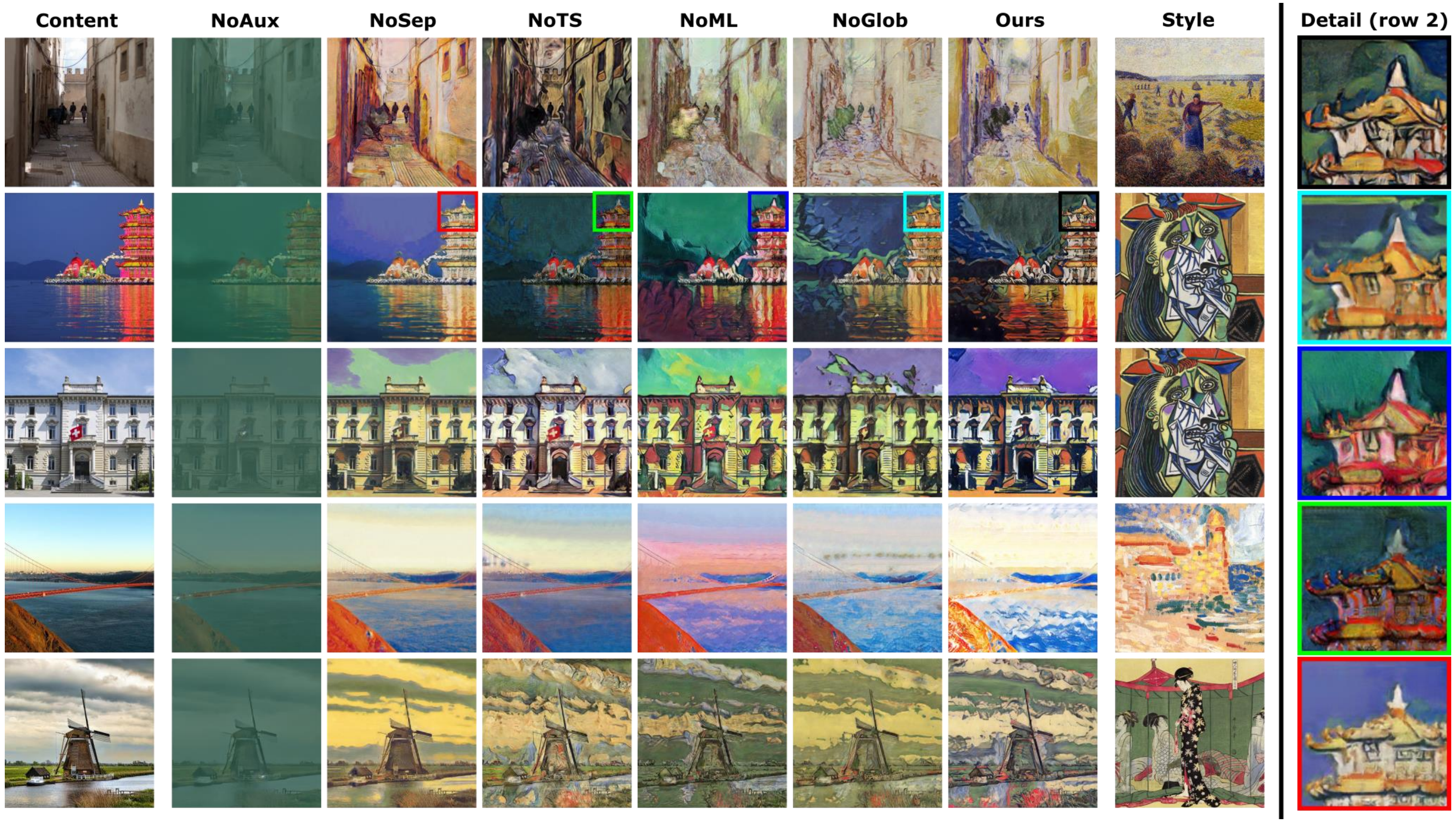
2차 피드백 내용

Q. embed, collab loss의 차이점
A. embed는 sencoder를 만들기 위해 존재(feat loss), collab은 encoder를 만들기 위해 존재(rec loss)

02 Two-Stage Peer-Regularized Feature Recombination for Arbitrary Image Style Transfer

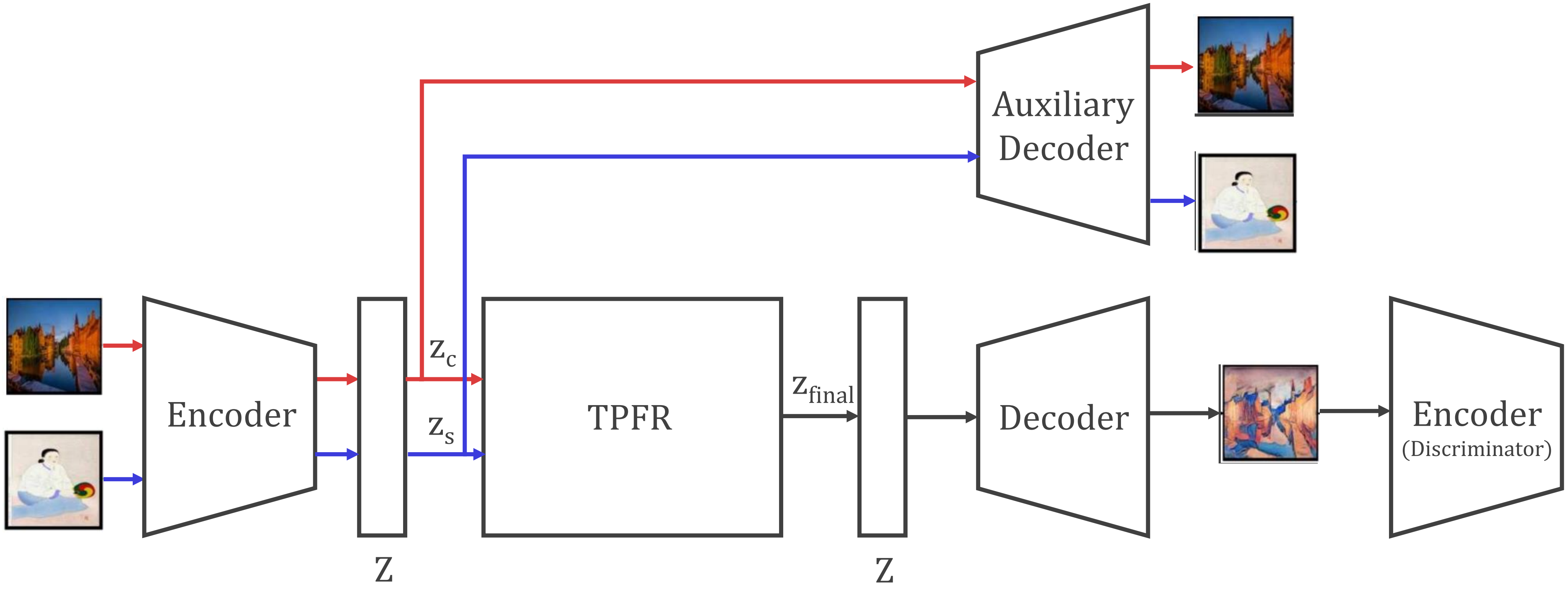
Jan Svoboda, Asha Anoosheh, Christian Osendorfer, Jonathan Masci, CVPR, 2020





02 Two-Stage Peer-Regularized Feature Recombination for Arbitrary Image Style Transfer

Jan Svoboda, Asha Anoosheh, Christian Osendorfer, Jonathan Masci, CVPR, 2020



TPFR : latent space에서 효과적으로 style transfer 하기 위해 사용하는 모듈, input pixel이 어떤 style과 가장 가까운 지 매칭하기 위해 k-NN graph를 사용하고 recombination으로 각각의 mapping에 따라 style를 강화시켜 도출 된 latent의 style이 decoder의 입력으로 사용됨

02

Two-Stage Peer-Regularized Feature Recombination for Arbitrary Image Style Transfer

Jan Svoboda, Asha Anoosheh, Christian Osendorfer, Jonathan Masci, CVPR, 2020

$$L_{total} = L_C + L_D + L_{\widetilde{D}}$$

L_C : Discriminator

L_D : Main decoder

$L_{\widetilde{D}}$: Auxiliary decoder

02 Two-Stage Peer-Regularized Feature Recombination for Arbitrary Image Style Transfer

Jan Svoboda, Asha Anoosheh, Christian Osendorfer, Jonathan Masci, CVPR, 2020

$$L_{total} = L_C + L_D + L_{\tilde{D}}$$
$$L_{\tilde{D}} = L_{z_{style}} + \tilde{L}_{z_{style}} + \tilde{L}_{z_{style}}$$

$$\begin{aligned} L_{z_{cont}} &= f[E(D(T(z_i, z_t)))_C - (z_i)_C] \\ &+ f[E(D(T(z_i, z_i)))_C - (z_i)_C] \end{aligned} \quad (2)$$

Content feature cycle loss

$$\begin{aligned} L^{pos}_{z_{style}} &= f[(z_{i_1})_S - (z_{i_2})_S] + f[(z_{t_1})_S - (z_{t_2})_S] \\ L^{neg}_{z_{style}} &= f[(z_{i_1})_S - (z_{t_1})_S] + f[(z_{i_2})_S - (z_{t_2})_S] \\ L_{z_{style}} &= L^{pos}_{z_{style}} + max(0.0, \mu - L^{neg}_{z_{style}}). \end{aligned} \quad (3)$$

Metric learning loss : latent의 style part에 clustering을 적용하기 위해 사용된다.
Posivite loss에서는 첫번째 latent와 두번째 latent의 style part가 유사해지길 원하며(결과적으로 지향)
Negatice loss에서는 latent가 자신의 본래 style part와 유사해지길 원한다. (결과적으로 지양)

$$\tilde{L}_{idt} = f[\tilde{D}(E(x_i)) - x_i] + f[\tilde{D}(E(x_t)) - x_t]. \quad (4)$$

Classical reconstruction loss : 오토인코더에서 주로 사용하는 reconstruction loss

$$\tilde{L}_{z_{cycle}} = f[E(\tilde{D}(z_i)) - (z_i)] + f[E(\tilde{D}(z_t)) - (z_t)], \quad (5)$$

Latent cycle loss
: 입력 이미지의 latent code와 그 이미지의 reconstruction 의 latent code 차이

02 Two-Stage Peer-Regularized Feature Recombination for Arbitrary Image Style Transfer

Jan Svoboda, Asha Anoosheh, Christian Osendorfer, Jonathan Masci, CVPR, 2020

$$L_{total} = L_C + L_D + L_{\tilde{D}}$$
$$L_D = L_{gen} + L_{z_{transf}} + \lambda L_{idt}$$

$$L_{gen} = \mathbb{E}_{x_i \sim \mathbb{P}}\left[\left(C(x_i) - \mathbb{E}_{x_f \sim \mathbb{Q}} C(x_f) + 1\right)^2\right] + \mathbb{E}_{x_f \sim \mathbb{Q}}\left[\left(\mathbb{E}_{x_i \sim \mathbb{P}} C(x_i) - C(x_f) + 1\right)^2\right], \quad (7)$$

Decoder adversarial loss
- P : 실제 데이터의 분포
- Q : 생성된 데이터의 분포(fake)
- C : discriminator

$$L_{z_{transf}} = f[E(D(T(z_i, z_t)))_C - (z_i)_C] + f[E(D(T(z_i, z_t)))_S - (z_t)_S]. \quad (8)$$

Transfer latent cycle loss : 스타일화 된 이미지의 Iinput contet와 target style을 적절히 recombining(재결합) 하여 나타내기 위해 사용

$$L_{idt} = f[D(T(z_i, z_i)) - x_i] + f[D(T(z_t, z_t)) - x_t]. \quad (9)$$

Classical reconstruction loss

02 Two-Stage Peer-Regularized Feature Recombination for Arbitrary Image Style Transfer

Jan Svoboda, Asha Anoosheh, Christian Osendorfer, Jonathan Masci, CVPR, 2020

$$L_{total} = L_C + L_D + L_{\widetilde{D}}$$
$$L_C$$

$$\begin{aligned} L_C = \; & \mathbb{E}_{x_i \sim \mathbb{P}} \left[\left(C\left(x_i\right) - \mathbb{E}_{x_f \sim \mathbb{Q}} C\left(x_f\right) - 1 \right)^2 \right] \\ & + \mathbb{E}_{x_f \sim \mathbb{Q}} \left[\left(\mathbb{E}_{x_i \sim \mathbb{P}} C\left(x_i\right) - C\left(x_f\right) - 1 \right)^2 \right]. \end{aligned} \tag{11}$$

C : discriminator

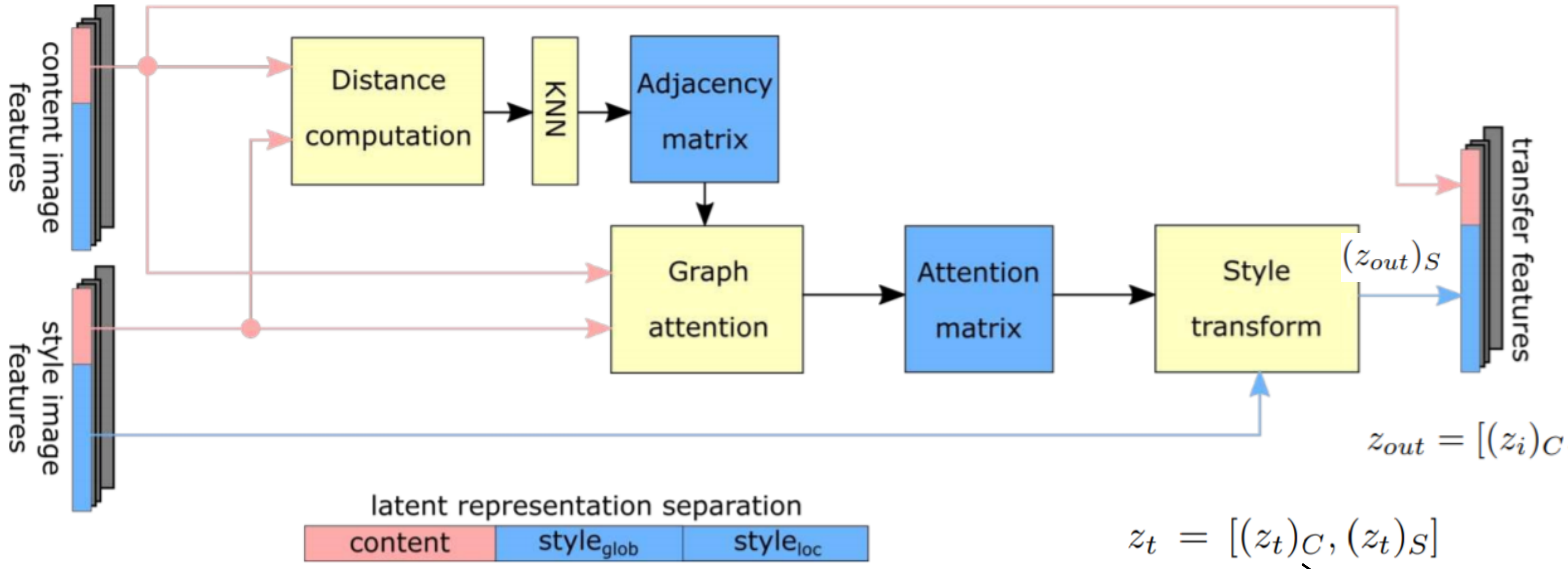
실제 이미지에 대한 판별 결과와(1에 가까움)
가짜 이미지에 대한 판별 결과의 차이를 구함(1에 가까울수록 좋음)

02 Two-Stage Peer-Regularized Feature Recombination for Arbitrary Image Style Transfer

Jan Svoboda, Asha Anoosheh, Christian Osendorfer, Jonathan Masci, CVPR, 2020

2차 피드백 내용

Q. Z_{final} ?
A.



$$z_{out} = [(z_i)_C, (z_{out})_S].$$

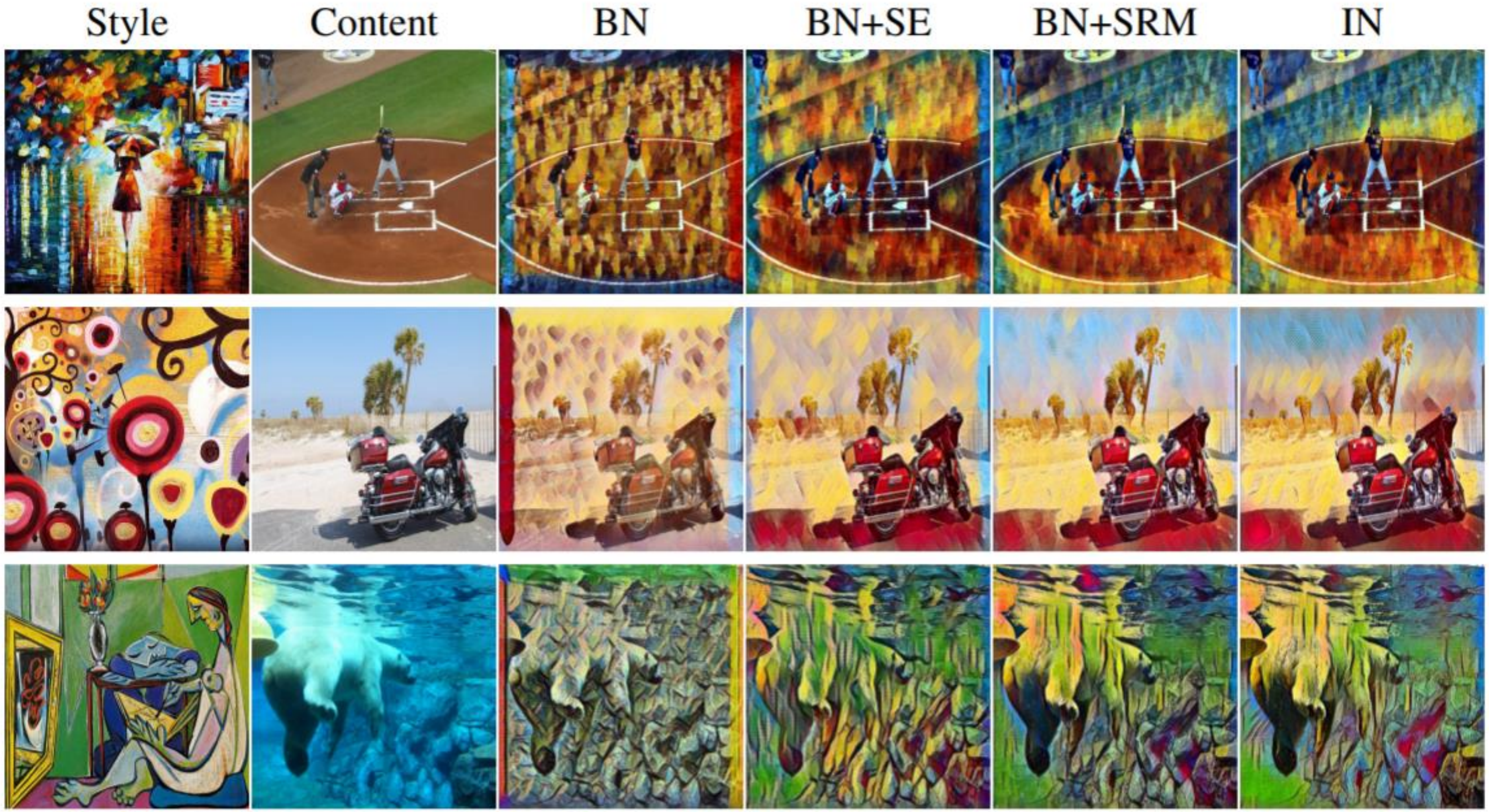
$$z_t = [(z_t)_C, (z_t)_S]$$

$$(z_{out})_S$$

$$z_{final} = [(z_{final})_C, (z_{out})_S]$$

03 SRM: A Style-Based Recalibration Module for Convolutional Neural Networks

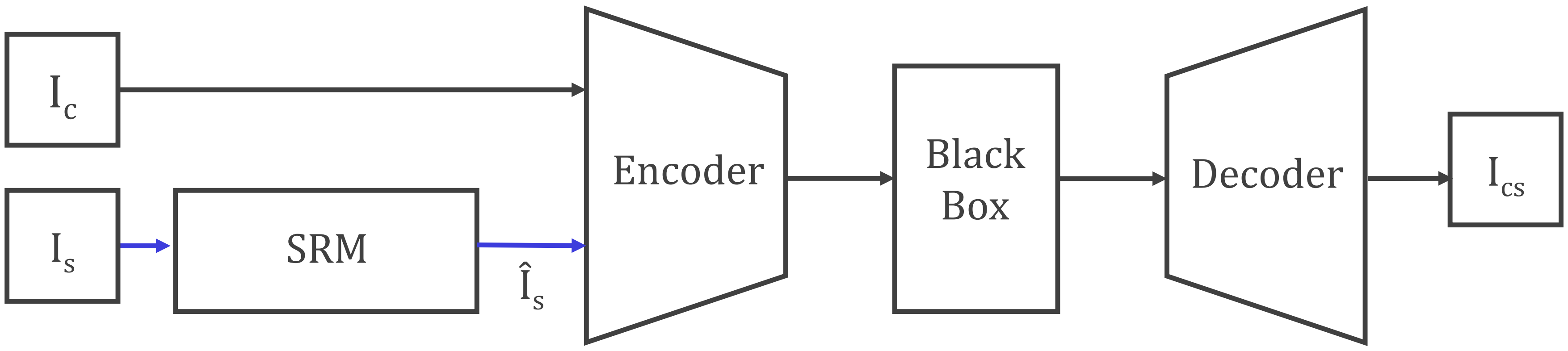
HyunJae Lee, Hyo-Eun Kim, Hyeonseob Nam, ICCV, 2019



03

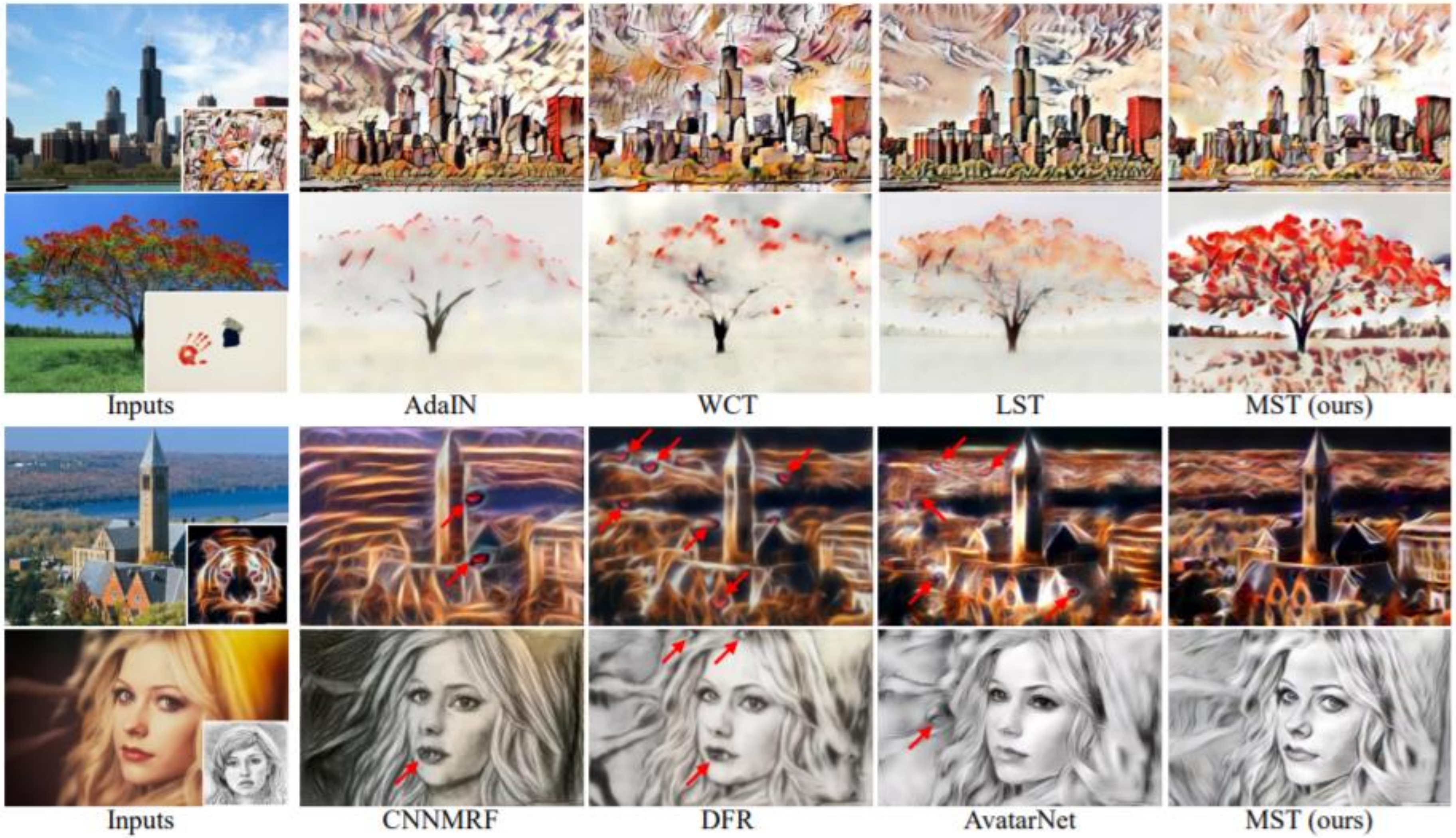
SRM: A Style-Based Recalibration Module for Convolutional Neural Networks

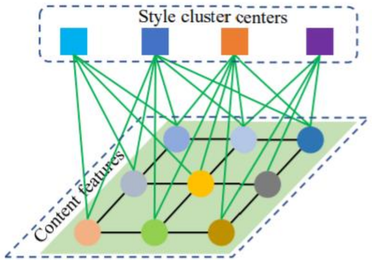
HyunJae Lee, Hyo-Eun Kim, Hyeonseob Nam, ICCV, 2019



04 Multimodal Style Transfer via Graph Cuts

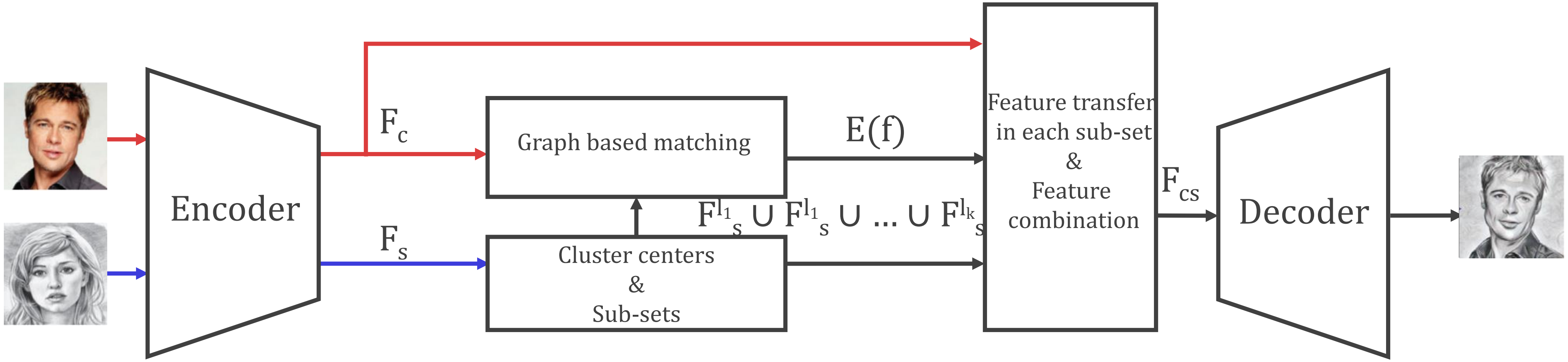
Yulun Zhang, Chen Fang, Yilin Wang, Zhaowen Wang, Zhe Lin, Yun Fu, Jimei Yang, ICCV, 2019





04 Multimodal Style Transfer via Graph Cuts

Yulun Zhang, Chen Fang, Yilin Wang, Zhaowen Wang, Zhe Lin, Yun Fu, Jimei Yang, ICCV, 2019



Graph based matching : 라벨이 붙은 subset과 content feature를 매칭시킴
Cluster centers & Subset : style feature를 cluster center 방식을 통해 subset 으로 나누어 라벨을 부여함
Feature transfer & combination : 각각의 style 에 따라 content 를 subset 으로 분리하고 결합하여 content style feature 를 생성함

04 Multimodal Style Transfer via Graph Cuts

Yulun Zhang, Chen Fang, Yilin Wang, Zhaowen Wang, Zhe Lin, Yun Fu, Jimei Yang, ICCV, 2019

$$\boxed{L_{total} = l_C + \gamma l_S}$$

l_C : Content loss

l_S : Style loss

04 Multimodal Style Transfer via Graph Cuts

Yulun Zhang, Chen Fang, Yilin Wang, Zhaowen Wang, Zhe Lin, Yun Fu, Jimei Yang, ICCV, 2019

$$\boxed{L_{total} = l_C + \gamma l_S}$$

$$l_c = \left\| \phi_{4\_1}\left(I_c\right) - \phi_{4\_1}\left(I_{cs}\right) \right\|_2 , \qquad (11)$$

→ φ4_1 (·) : Conv_4_1에서 추출한 feature

$$l_s = \sum_{i=1}^{4} \left(\left\| \mu\left(\phi_{i\_1}\left(I_s\right)\right) - \mu\left(\phi_{i\_1}\left(I_{cs}\right)\right) \right\|_2 \right) + \sum_{i=1}^{4} \left(\left\| \sigma\left(\phi_{i\_1}\left(I_s\right)\right) - \sigma\left(\phi_{i\_1}\left(I_{cs}\right)\right) \right\|_2 \right) , \qquad (12)$$

→ φi_1 (·) : Conv_i_1에서 추출한 feature

04 Multimodal Style Transfer via Graph Cuts

Yulun Zhang, Chen Fang, Yilin Wang, Zhaowen Wang, Zhe Lin, Yun Fu, Jimei Yang, ICCV, 2019

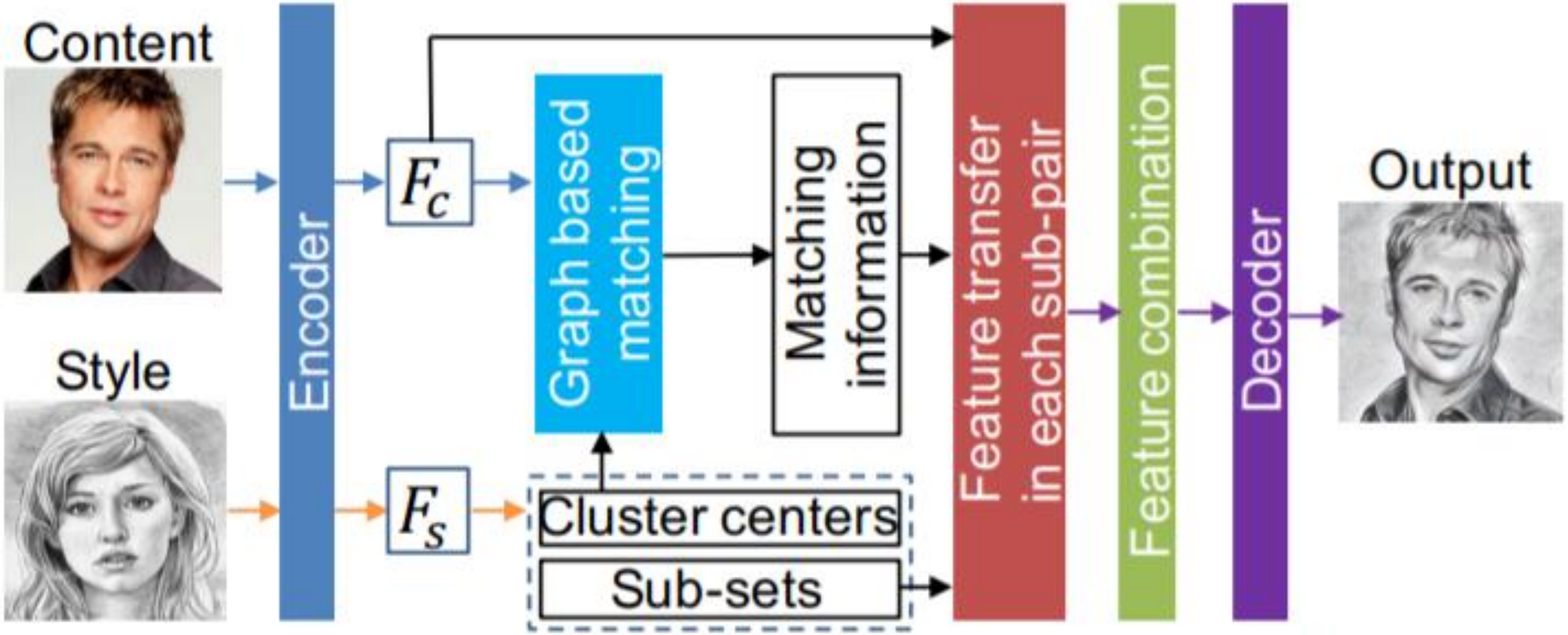


Figure 3: An overview of our MST algorithm.

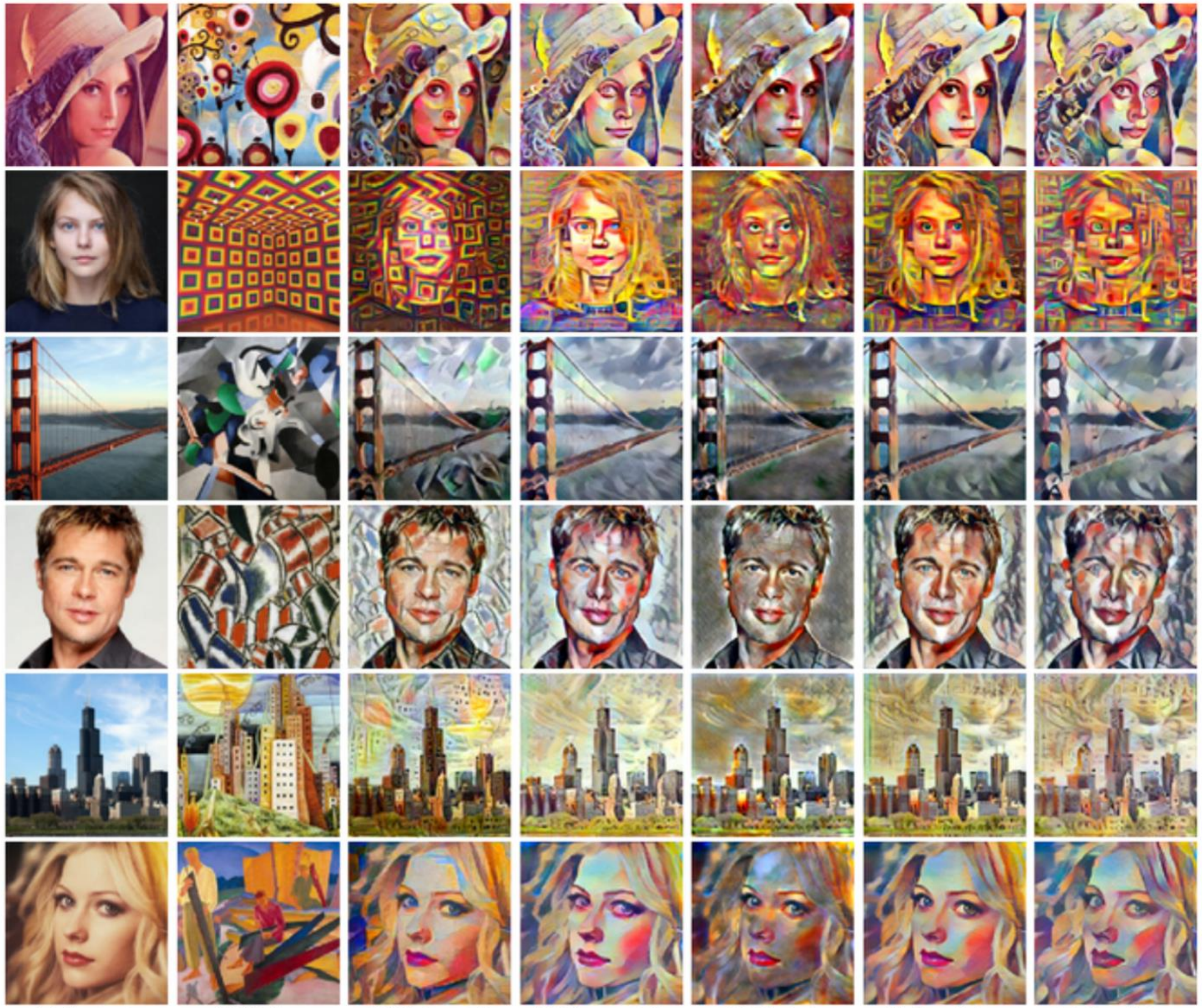
2차 피드백 내용

Q. 무엇으로 Clustering을 하나?
A. style feature

05

Total Style Transfer With A Single Feed-forward Network

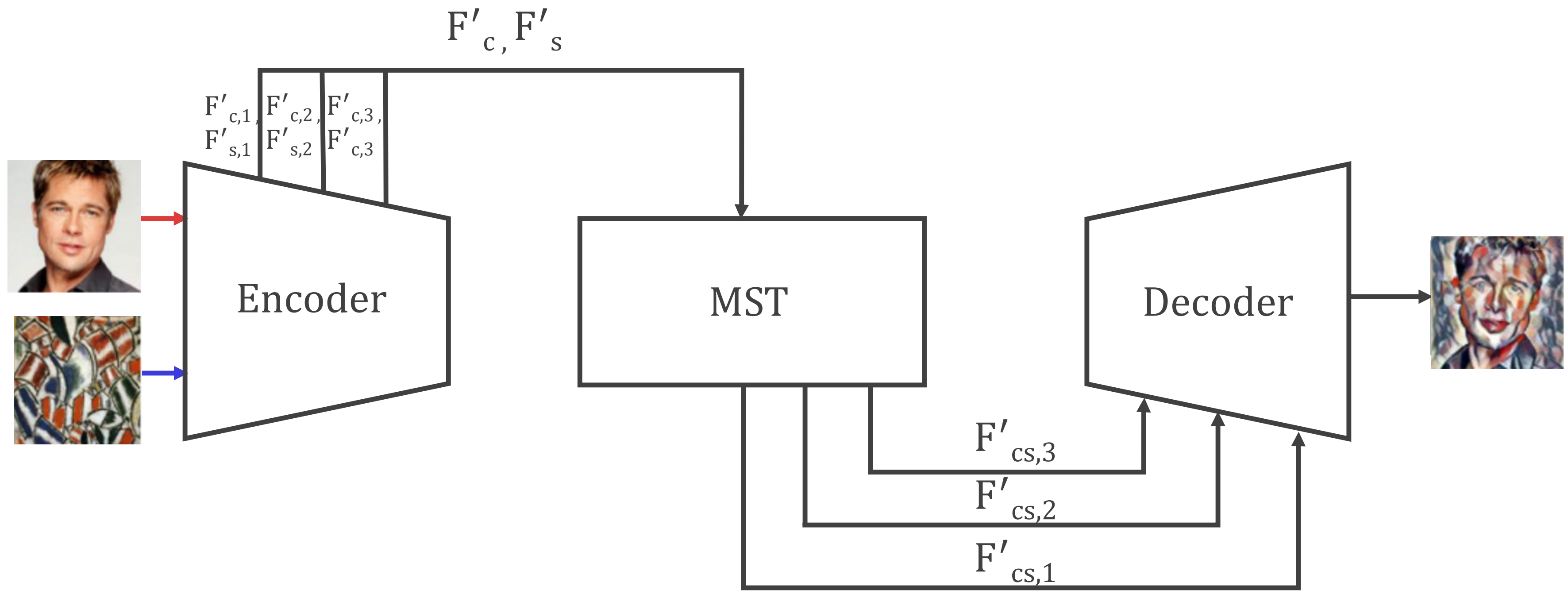
Minseong Kim, Hyun-Chul Choi, ICLR, 2019



(a) Content (b) Style (c) Gatys et al. (d) Huang et al. (e) Li et al. (f) Ours-intra (e) Ours-inter

05 Total Style Transfer With A Single Feed-forward Network

Minseong Kim, Hyun-Chul Choi, ICLR, 2019



MST(Multi-scaled style transformer) : 내부 SST와 skip-connected decoder를 사용하여 multi-scaled style을 transfer 함

05

Total Style Transfer With A Single Feed-forward Network

Minseong Kim, Hyun-Chul Choi, ICLR, 2019

$$L_{content} = \ Gatys\ et\ al. (2016)$$
$$L_{style_{intra}},\ L_{style_{inter}}$$

$\mathrm{L_{content}}$: Content loss

$\mathrm{L_{style_{intra}}}$: Correlations between channels in each scale

$\mathrm{L_{style_{inter}}}$: Correlations between scales

05 Total Style Transfer With A Single Feed-forward Network

Minseong Kim, Hyun-Chul Choi, ICLR, 2019

$$L_{content} = \ Gatys\ et\ al.(2016)$$
$$L_{style_{intra}},\ L_{style_{inter}}$$

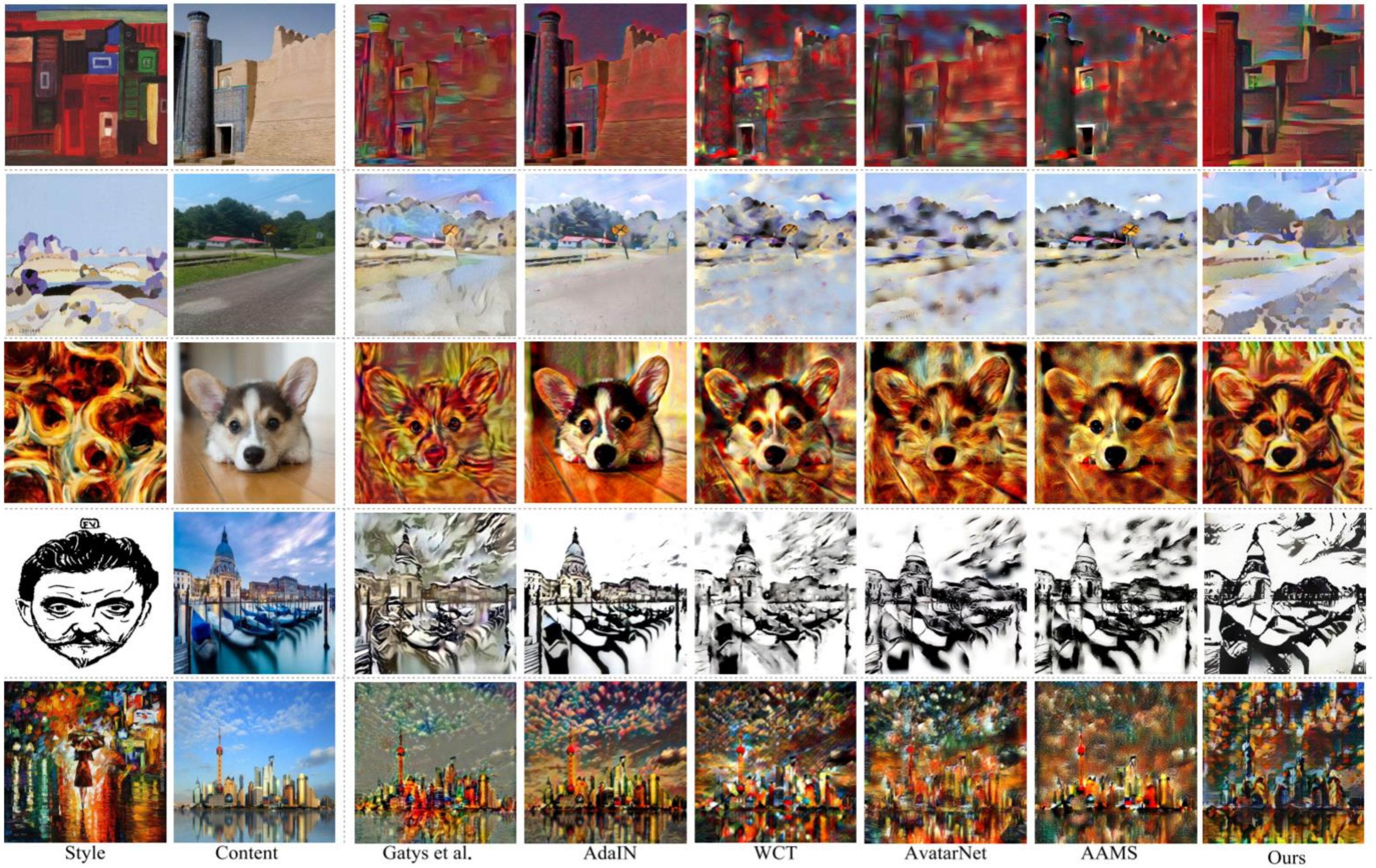
$$L_{style_{intra}} = \sum_i ||\mu_{s,i} - \mu_{o,i}|| + \sum_i ||cov(\bar{F}_{s,i}) - cov(\bar{F}_{o,i})||, \quad (5)$$

$$L_{style_{inter}} = ||\mu'_s - \mu'_o|| + ||cov(\bar{F'}_s) - cov(\bar{F'}_o)||, \quad (6)$$

$\bar{F}s, i$: i th zero − centered feature of style image (o : output)
$\bar{F}'_s$: concatenated feature of style image
cov : covariance matrix

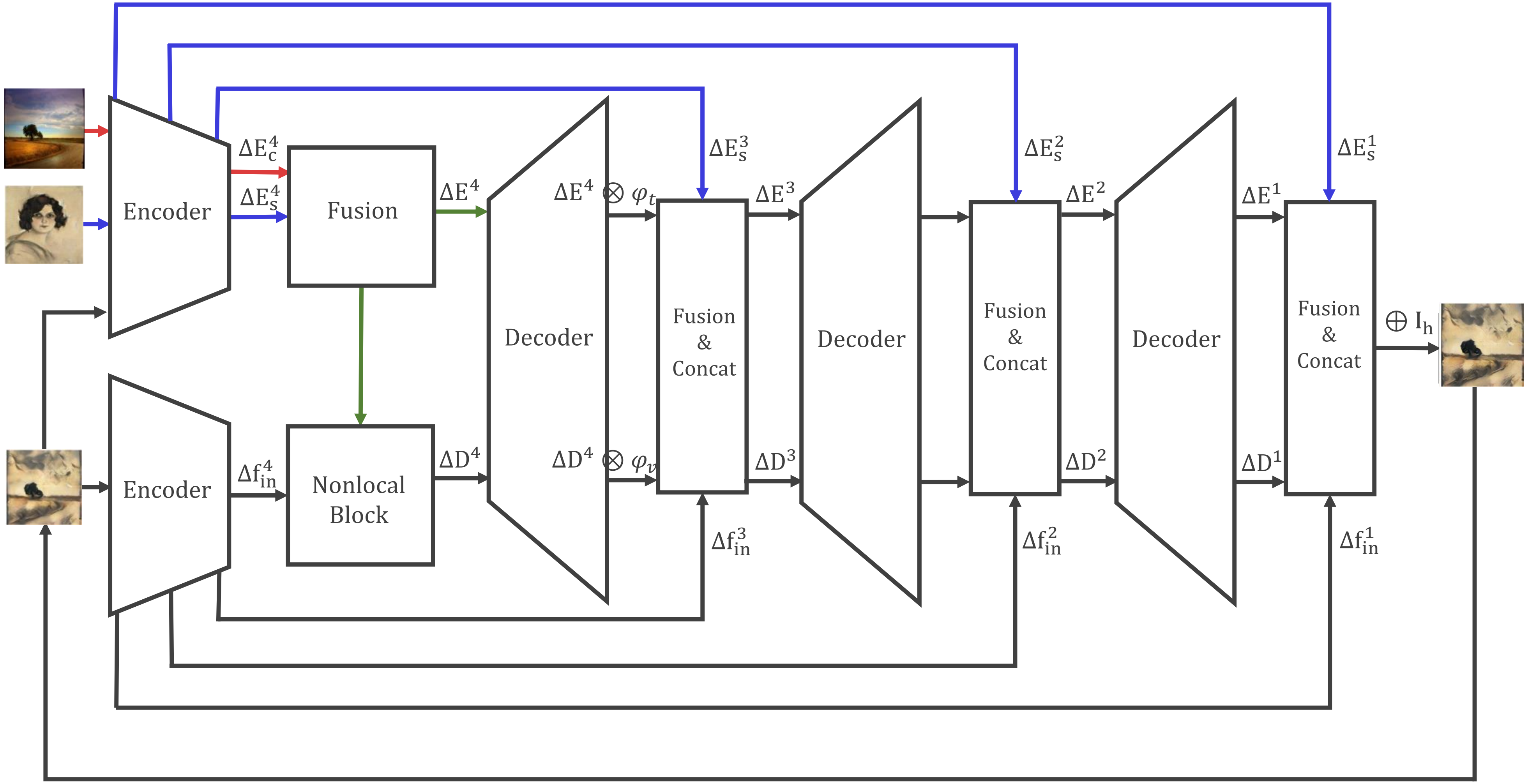
ETNet: Error Transition Network for Arbitrary Style Transfer

Chunjin Song, Zhijie Wu, Yang Zhou, Minglun Gong, Hui Huang, NIPS, 2019



06 ETNet: Error Transition Network for Arbitrary Style Transfer

Chunjin Song, Zhijie Wu, Yang Zhou, Minglun Gong, Hui Huang, NIPS, 2019



1st Fusion : 두 E를 합침
Nonlocal Block : globel residual feature를 추출
2nd Fusion and Concat : 더 lower한 level의 information 을 추출하기 위한 모듈

06

ETNet: Error Transition Network for Arbitrary Style Transfer

Chunjin Song, Zhijie Wu, Yang Zhou, Minglun Gong, Hui Huang, NIPS, 2019

$$\boxed{L_{total}^{k} = \lambda_{pc}^{k} L_{pc}^{k} + \lambda_{ps}^{k} L_{ps}^{k} + \lambda_{tv}^{k} L_{TV}}$$

* Total Loss at k-th level of a pyramid (k =1,2,3)

L_{pc}^{k} : content loss

L_{ps}^{k} : style loss

L_{TV} : Total variation loss (means adding to acheieve the piece $-$ wise smoothness)

06 ETNet: Error Transition Network for Arbitrary Style Transfer

Chunjin Song, Zhijie Wu, Yang Zhou, Minglun Gong, Hui Huang, NIPS, 2019

$$L^k_{total} = \lambda^k_{pc} L^k_{pc} + \lambda^k_{ps} L^k_{ps} + \lambda^k_{tv} L_{TV}$$

$$L^k_{pc} = \|\Phi_{vgg}(I^k_c) - \Phi_{vgg}(I^k_{cs})\|_2 + \sum_{j=k+1}^{K} \|\Phi_{vgg}(I^j_c) - \Phi_{vgg}(\tilde{I}^j_{cs})\|_2, \tag{7}$$

$$L^k_{ps} = \sum_{i=1}^{L} \|G^i(I^k_s) - G^i(I^k_{cs})\|_2 + \sum_{j=k+1}^{K} \|G^L(I^j_s) - G^L(\tilde{I}^j_{cs})\|_2, \tag{8}$$

Φ_{vgg} : VGG Encoder

G^i : Gram matrix for features extracted at layer i in the encoder network

I^k_c : input image

$\tilde{I}^j_{cs}$: (j-k) repeated applications of downsampling operation $\mathrm{d}(\cdot)$ on I^k_{cs}

(e.g. k=1. $\tilde{I}^3_{cs} = \mathrm{d}(\mathrm{d}(I^1_{cs}))$ (To capture large patterns)

set $\mathrm{L} = 4$

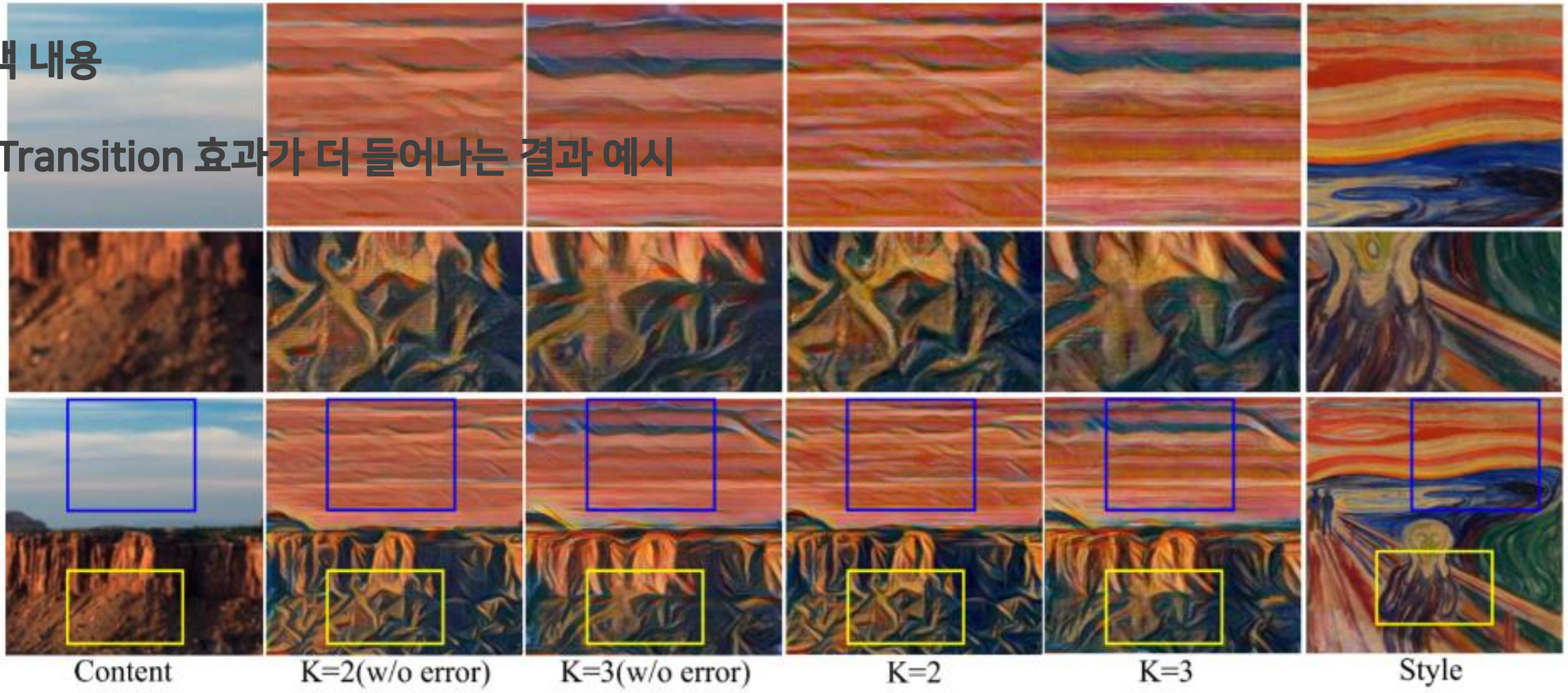
06

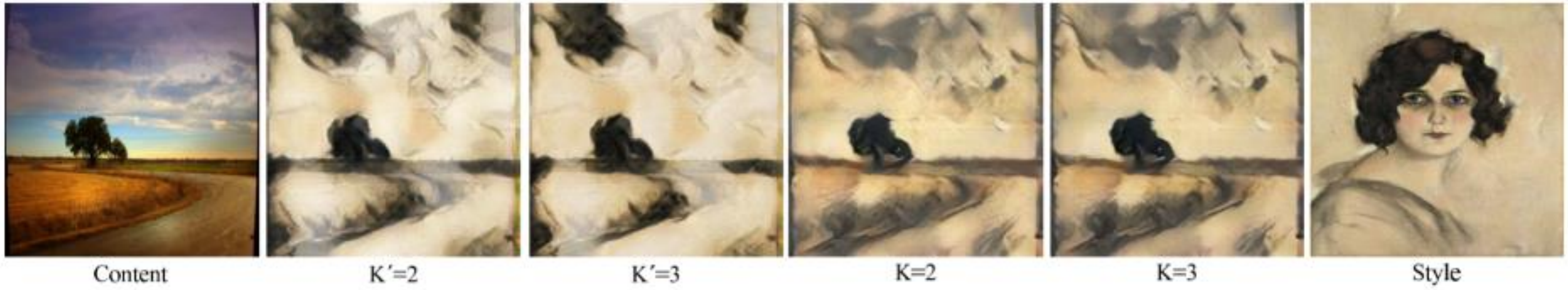
ETNet: Error Transition Network for Arbitrary Style Transfer

Chunjin Song, Zhijie Wu, Yang Zhou, Minglun Gong, Hui Huang, NIPS, 2019

2차 피드백 내용

Q. Error Transition 효과가 더 들어나는 결과 예시

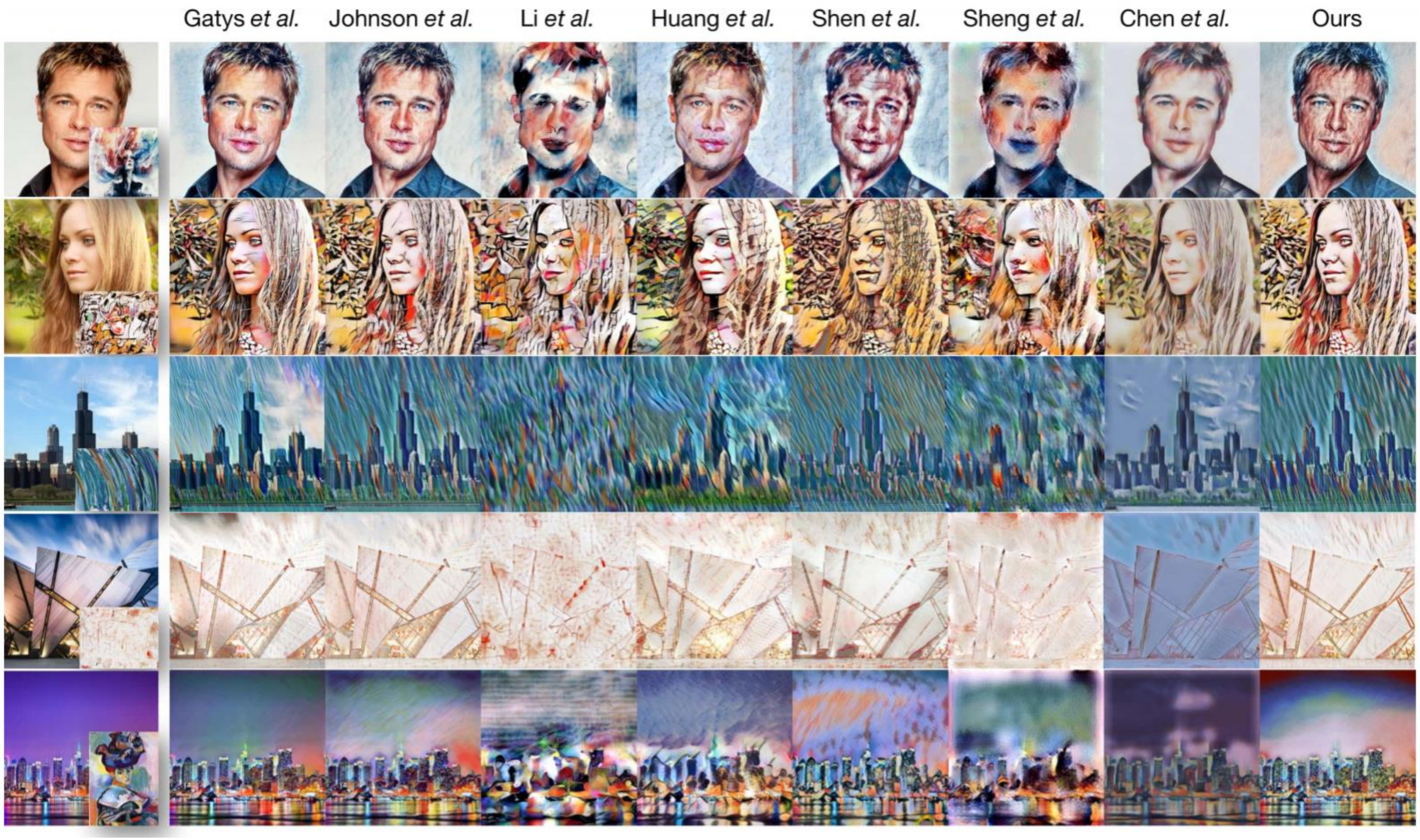




07

MetaStyle
: Three-Way Trade-off among Speed, Flexibility, and Quality in Neural Style Transfer

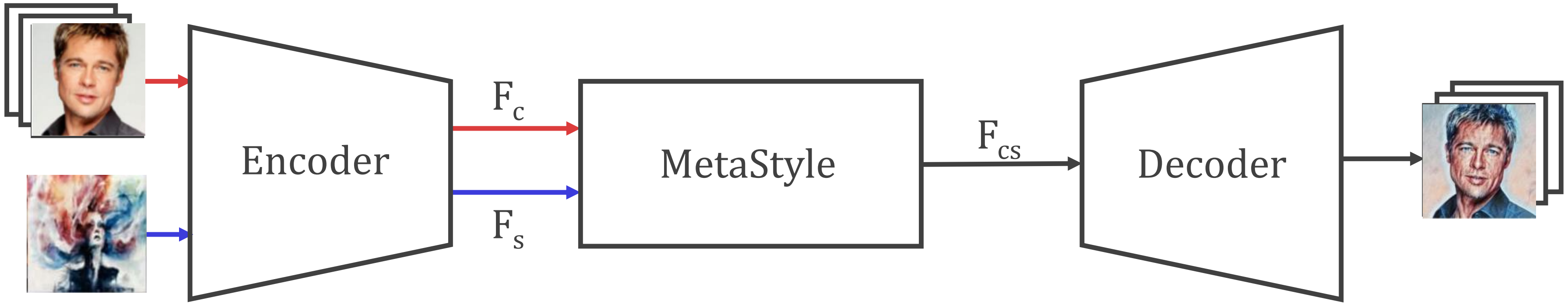
Chi Zhang, Yixin Zhu, Song-Chun Zhu, AAAI, 2019



07

MetaStyle
: Three-Way Trade-off among Speed, Flexibility, and Quality in Neural Style Transfer

Chi Zhang, Yixin Zhu, Song-Chun Zhu, AAAI, 2019



MetaStyle : 각각의 이미지에 대한 inner loss와 outer loss를 계산하여 스타일에 중립적인 표현을 배우고 새로운 스타일을 적용을 위한 일정 업데이트를 거침

07

MetaStyle
: Three-Way Trade-off among Speed, Flexibility, and Quality in Neural Style Transfer

Chi Zhang, Yixin Zhu, Song-Chun Zhu, AAAI, 2019

$$E_{c,s}[\, l(I_c, I_s, M\big(I_c; w_{s,T}\big)]$$
$$E_c[\, l(I_c, I_s, M\big(I_c; w_{s,t-1}\big)]$$

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \mathbb{E}_{c,s}[\ell(I_c, I_s, M(I_c; w_{s,T}))] \\ \text{subject to} \quad & w_{s,0} = \theta \\ & w_{s,t} = w_{s,t-1} - \delta \nabla \mathbb{E}_c[\ell(I_c, I_s, M(I_c; w_{s,t-1}))], \end{aligned} \tag{7}$$

$E_{c,s}$: outer loss, respect style and content image in the validation set

L_θ , E_c: inner loss, respect content image in the training set

→ single style 에 특화 & 일반적인 initialization 유지

M : model

δ : learning rate of inner class

07 MetaStyle
: Three-Way Trade-off among Speed, Flexibility, and Quality in Neural Style Transfer

Chi Zhang, Yixin Zhu, Song-Chun Zhu, AAAI, 2019

2차 피드백 내용

Q. Speed And Flexibility

Method	Param	256 (s)	512 (s)	# Styles
Gatys *et al.*	N/A	7.7428	27.0517	∞
Johnson *et al.*	**1.68M**	**0.0044**	**0.0146**	1
Li *et al.*	34.23M	0.6887	1.2335	∞
Huang *et al.*	7.01M	0.0165	0.0320	∞
Shen *et al.*	219.32M	**0.0045**	**0.0147**	∞
Sheng *et al.*	147.22M	0.5089	0.6088	∞
Chen *et al.*	**1.48M**	0.2679	1.0890	∞
Ours	**1.68M**	**0.0047**	**0.0145**	$\infty^{\star}$

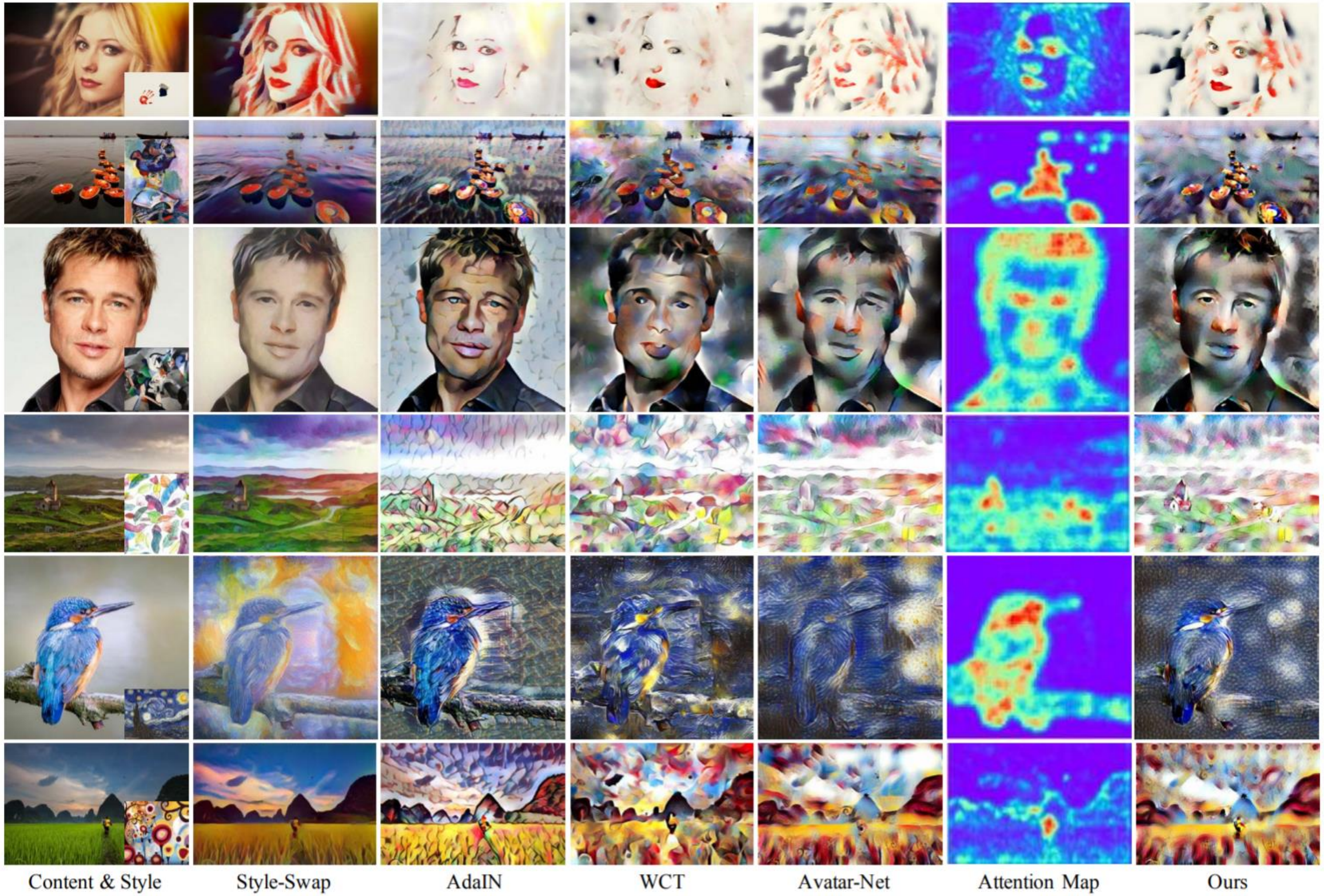
- 256/ 512 : 256 * 256 input, 512 * 512 input
  - # Style : 모델이 처리 할 수 있는 스타일의 수

➔ MetaStyle은 적은 업데이트 단계 이후 특정 스타일을 적용 가능 함

08

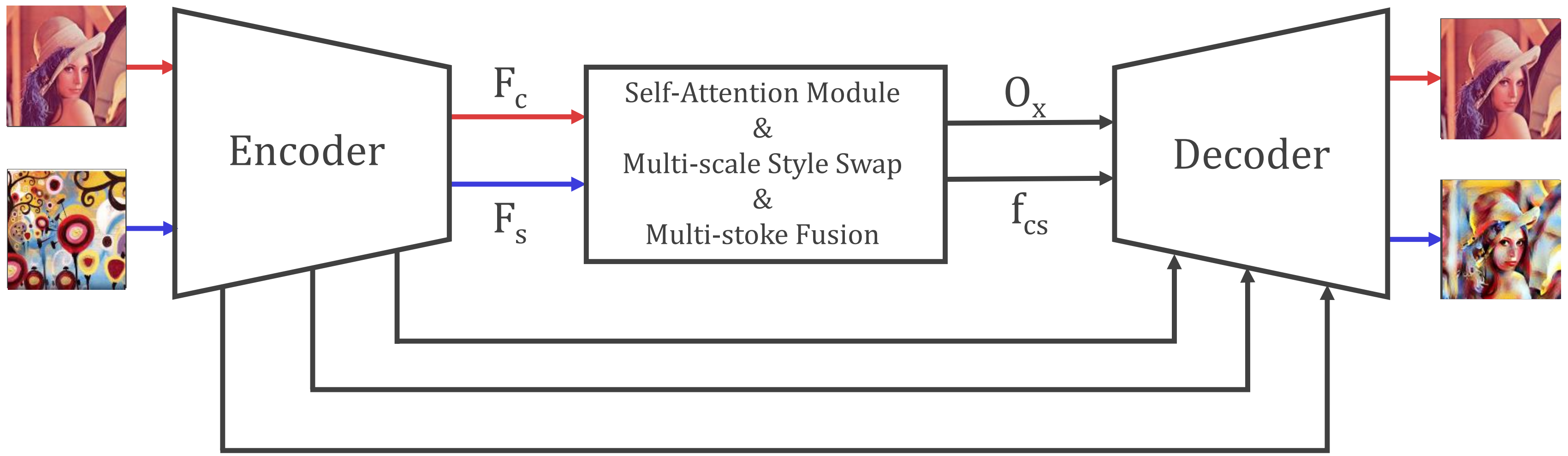
Attention-Aware Multi-Stroke Style Transfer

Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, Jun Wang, CVPR, 2019



08 Attention-Aware Multi-Stroke Style Transfer

Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, Jun Wang, CVPR, 2019



Self-Attention Module : Content의 두드러진 특징을 학습하고 모든 이미지의 content에 대한 activation feature map A_c를 생성하고 f_c와 연산되어 O_x를 출력
Multi-scale Style Swap : Style의 activation feature scale을 변경하며 패치 크기를 수정, content feature와 K style feature에 대한 style swap을 수행
K : user가 부여한 Clustering 번호로 Patch 별 가장 가까운 style feature로 content featurㄷ를 교체(Swap)
Multi-stroke fusion : arbitrary stroke를 통합하기 위해 attention map $\hat{A}_c^k$를 여러 cluster로 분할하여 유연하게 $\hat{f}_{cs}^k$와 융합

08 Attention-Aware Multi-Stroke Style Transfer

Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, Jun Wang, CVPR, 2019

$$\boxed{L = \lambda_{con} L_{con} + \lambda_{att} L_{att} + \lambda_{tv} L_{tv}}$$

* Total Loss at k-th level of a pyramid (k =1,2,3)

L_{con}: Semantic content loss(perceptual loss + pixel loss)
L_{att} : Sparse loss of self attention map A_x (for attention to smaller region)
L_{tv} : Total variation regularization loss (for spatial smoothness)

08 Attention-Aware Multi-Stroke Style Transfer

Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, Jun Wang, CVPR, 2019

$$L = \lambda_{con} L_{con} + \lambda_{att} L_{att} + \lambda_{tv} L_{tv}$$

$$\mathcal{L}_{con} = \sum_{l \in l_c} ||\phi_l(\hat{x}) - \phi_l(x)||_2^2 + \lambda_p ||\hat{x} - x||_2^2 , \quad (5)$$

$$\mathcal{L}_{att} = ||A_x||_1 . \quad (6)$$

$\hat{x}$: reconstruction of the input image

$\varphi_l(x)$: activations of the lth layer of the pre-trained VGG-19 network when processing image x

A_x : self-attention feature map (for awareness to small regions instead of the whole image)

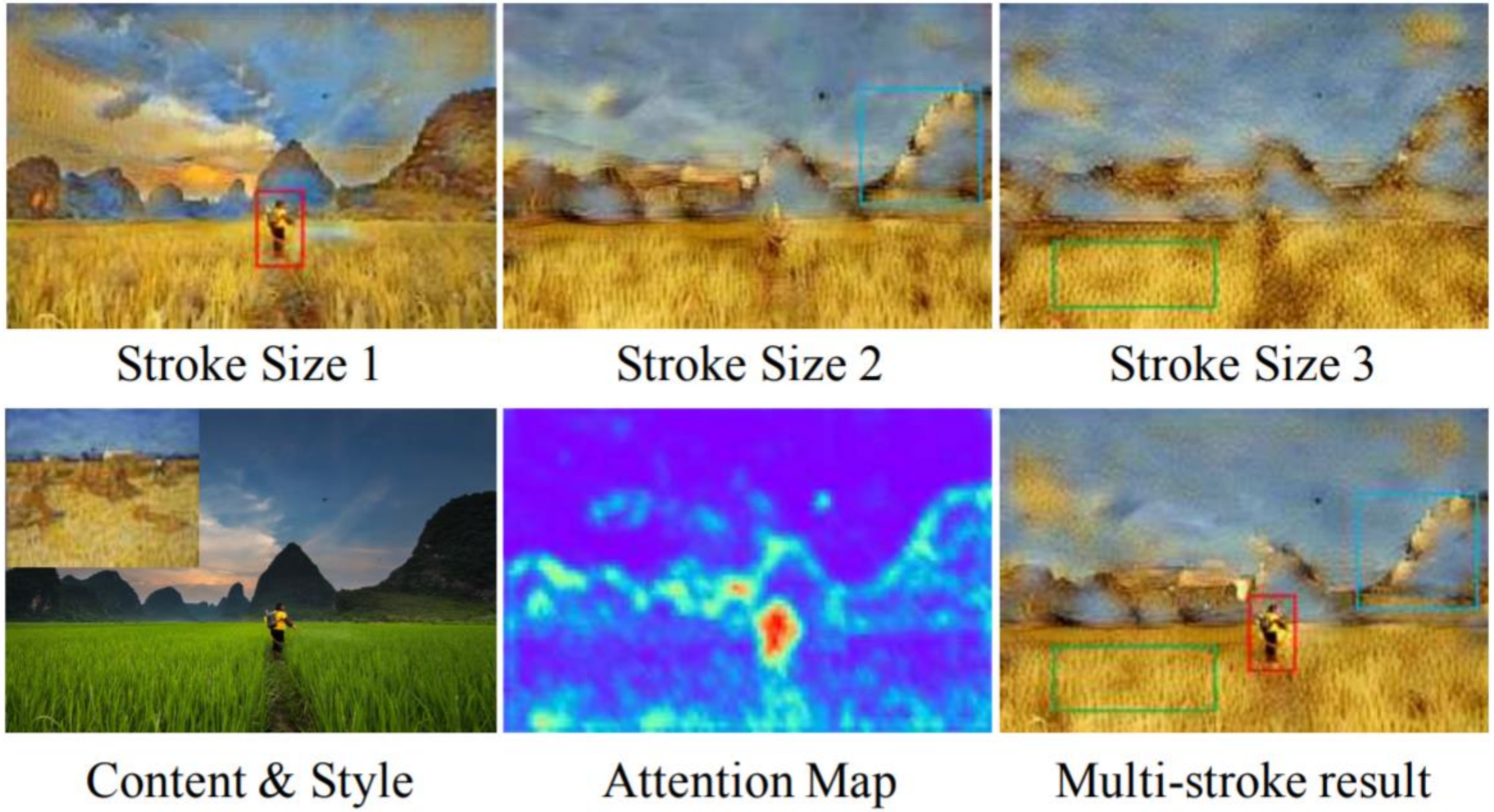
L_{tv} : J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In ECCV, 2016

08 Attention-Aware Multi-Stroke Style Transfer

Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, Jun Wang, CVPR, 2019

2차 피드백 내용

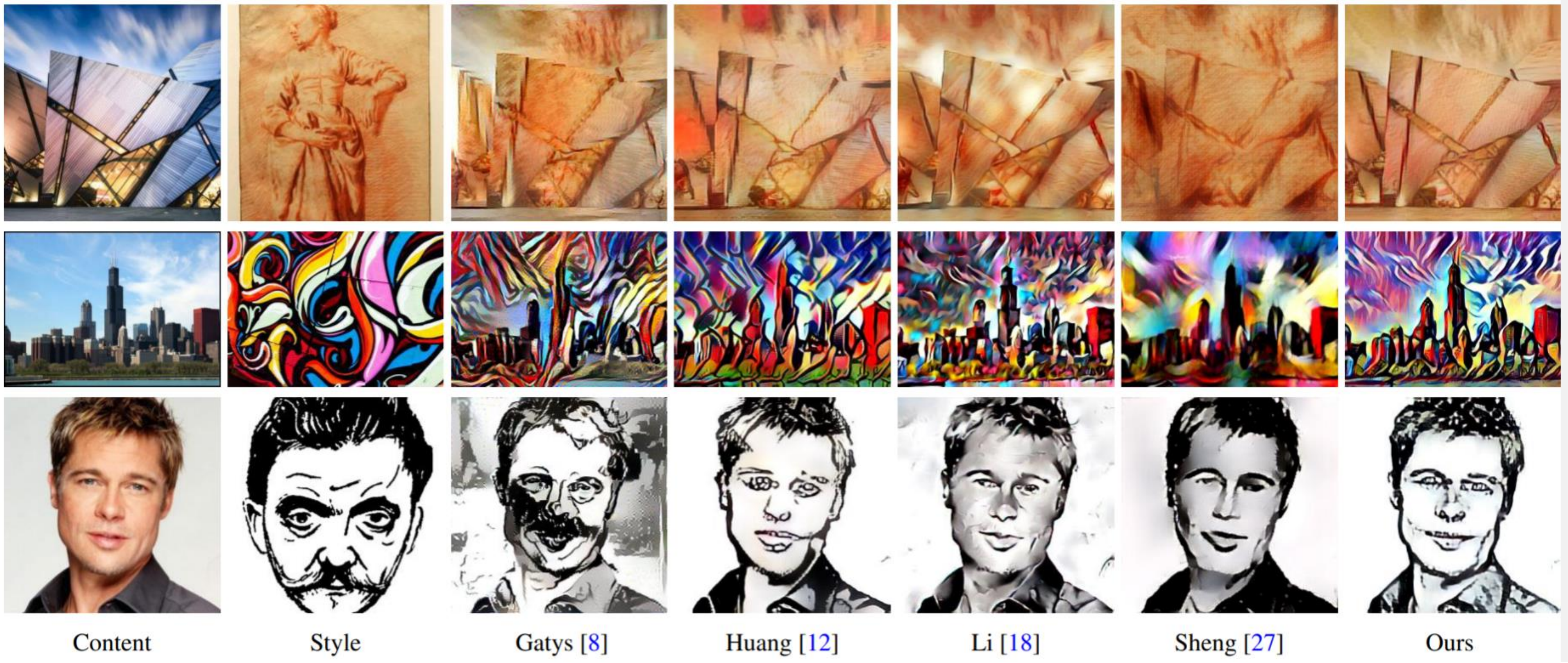
Q. Stroke차이
A.



09

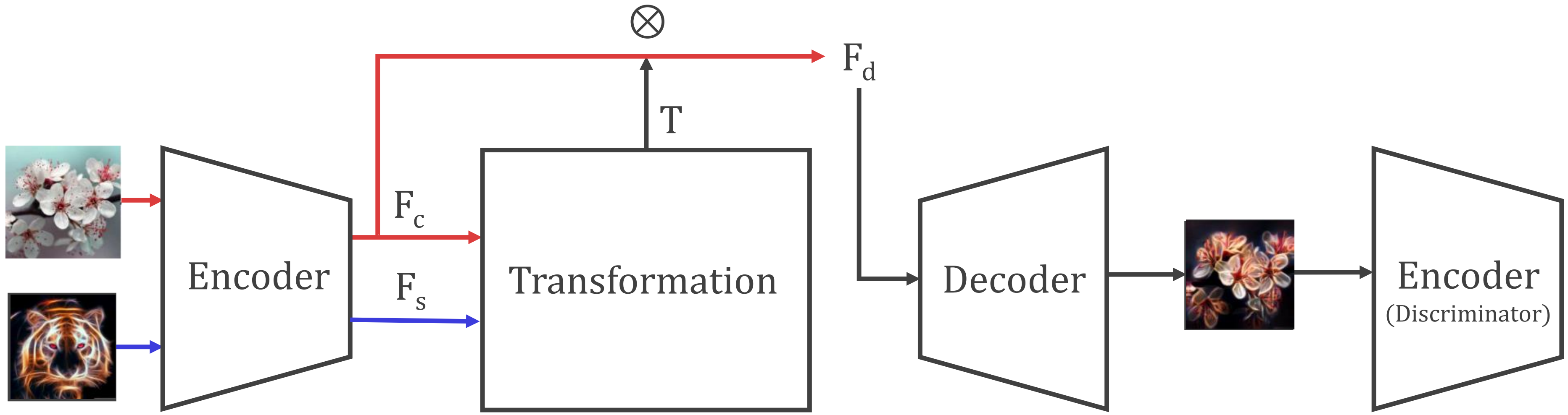
Learning Linear Transformations for Fast Image and Video Style Transfer

Xueting Li, Sifei Liu, Jan Kautz, Ming-Hsuan Yang, CVPR, 2019



09 Learning Linear Transformations for Fast Image and Video Style Transfer

Xueting Li, Sifei Liu, Jan Kautz, Ming-Hsuan Yang, CVPR, 2019



Transformation : 두 개의 독립적인 CNN을 사용하여 F_c , F_s에 대한 공분산 행렬을 이용하여 두 feature를 multiplication 함
* T는 출력 F_d에서 style이 content 이미지의 global & local 한 부분 모두 고르게 적용되길 바람

Learning Linear Transformations for Fast Image and Video Style Transfer

Xueting Li, Sifei Liu, Jan Kautz, Ming-Hsuan Yang, CVPR, 2019

$$F_d^* = \arg\min_{F_d} \frac{1}{NC} \|\bar{F_d}\bar{F_d}^\top - \bar{\phi_s}\bar{\phi_s}^\top\|_F^2$$
$$s.t. \bar{F_d} = T\bar{F_c}.$$

$$T = \left(V_s D_s^{\frac{1}{2}} V_s^\top\right) U \left(V_c D_c^{-\frac{1}{2}} V_c^\top\right), \qquad (3)$$

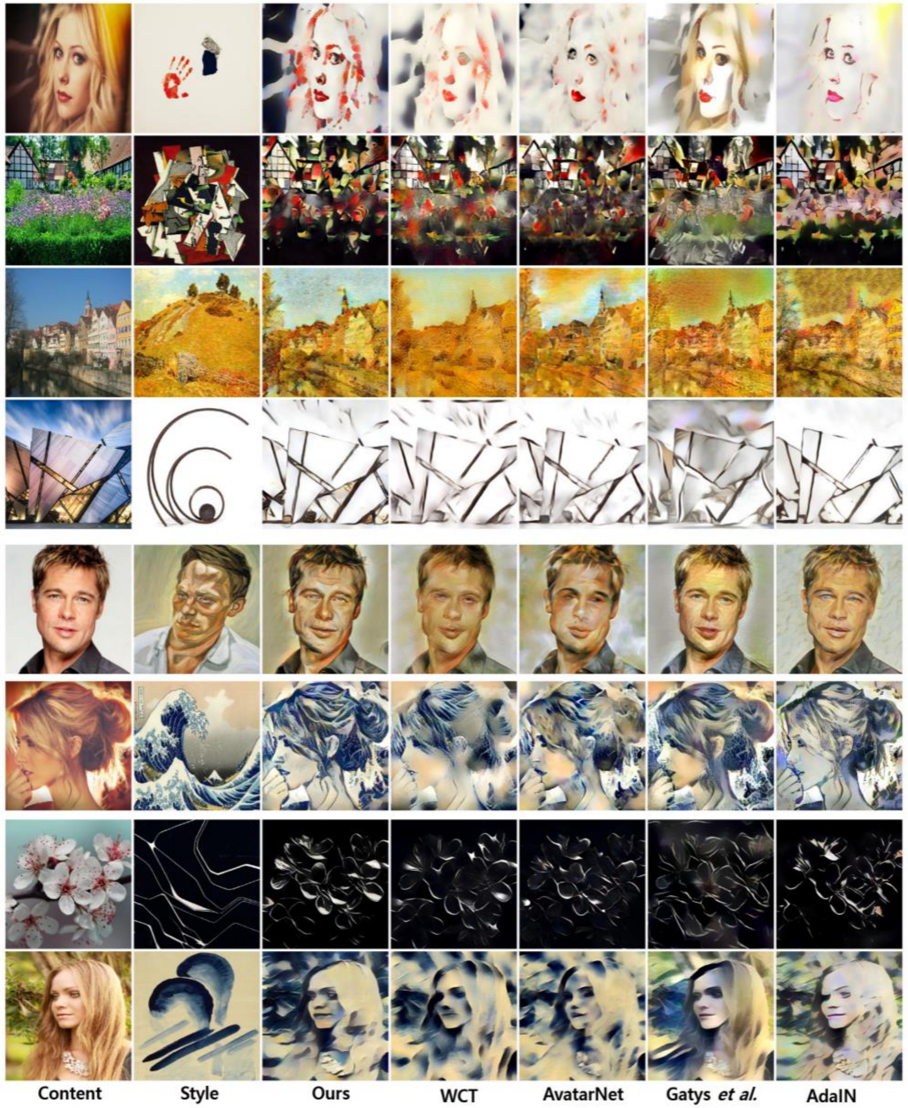
* Difference of centered covariance between F_d and φ_s

U : c-channel orthogonal group
VDV : singular value decomposition(SVD)
F_d : transformed feature vector
$\bar{F}_c$: vectorized feature map F_c with zero mean

L_c : J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In ECCV, 2016,
X. Huang and S. Belongie. Arbitrary style transfer in realtime with adaptive instance normalization. In ICCV, 2017.

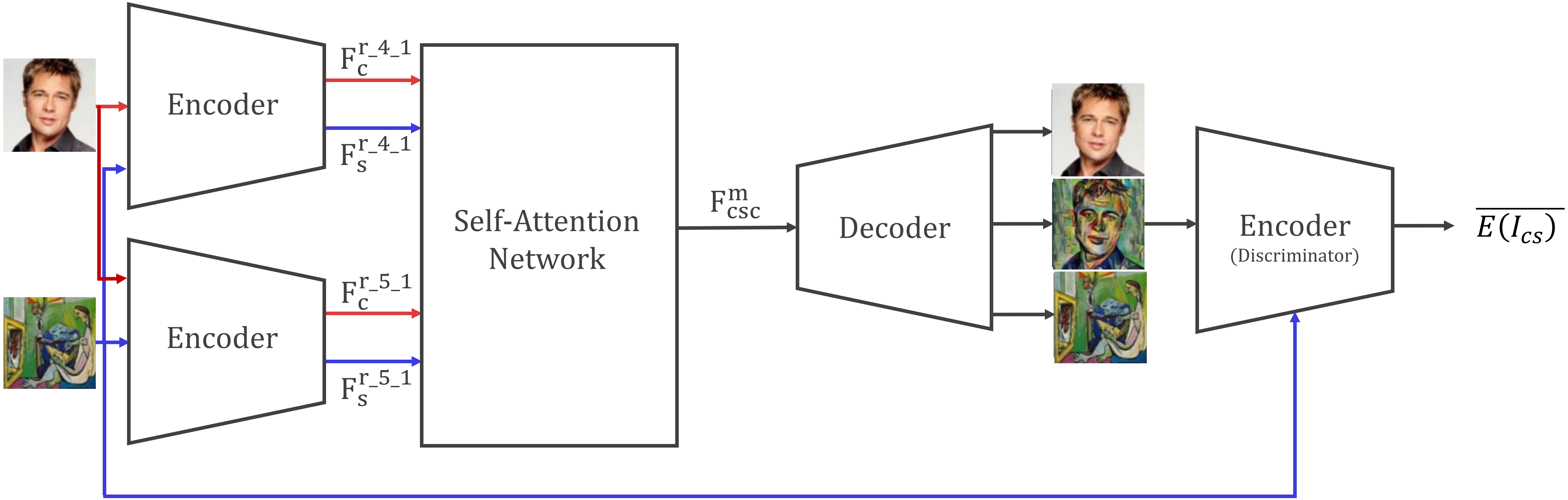
10 Arbitrary Style Transfer With Style-Attentional Networks

Dae Young Park, Kwang Hee Lee, CVPR, 2019



10 Arbitrary Style Transfer With Style-Attentional Networks

Dae Young Park, Kwang Hee Lee, CVPR, 2019



Self-Attention Network : global style pattern 과 local style pattern 을 적절히 융합하기 위해 서로 다른 레이어에서 나온 feature map 을 결합

10 Arbitrary Style Transfer With Style-Attentional Networks

Dae Young Park, Kwang Hee Lee, CVPR, 2019

$$\boxed{L_{total} = \lambda_c L_c + \lambda_s L_s + L_{identity}}$$

L_c : content loss (Euclidean distance between the mean-variance channel-wise normalized target feature)

L_s : style loss

$L_{identity}$: identity loss

Arbitrary Style Transfer With Style-Attentional Networks

Dae Young Park, Kwang Hee Lee, CVPR, 2019

$$L_{total} = \lambda_c L_c + \lambda_s L_s + L_{identity}$$

$$\mathcal{L}_c = ||\overline{E(I_{cs})^{\text{r\_4\_1}}} - \overline{F_c^{\text{r\_4\_1}}}||_2 + ||\overline{E(I_{cs})^{\text{r\_5\_1}}} - \overline{F_c^{\text{r\_5\_1}}}||_2. \quad (7)$$

$\overline{E(I_{cs})^{r\_4\_1}}, \overline{E(I_{cs})^{r\_5\_1}}$

: mean-variance channel-wise normalized features of the output image VGG features

$$\mathcal{L}_s = \sum_{i=1}^{L} ||\mu(\phi_i(I_{cs})) - \mu(\phi_i(I_s))||_2 + ||\sigma(\phi_i(I_{cs})) - \sigma(\phi_i(I_s))||_2, \quad (8)$$

φ : feature map of the layer in the encoder used to compute the style loss

$$\mathcal{L}_{identity} = \lambda_{identity1}(||(I_{cc} - I_c)||_2 + ||(I_{ss} - I_s)||_2) + \lambda_{identity2} \sum_{i=1}^{L} (||\phi_i(I_{cc}) - \phi_i(I_c)||_2 + ||\phi_i(I_{ss}) - \phi_i(I_s)||_2), \quad (9)$$

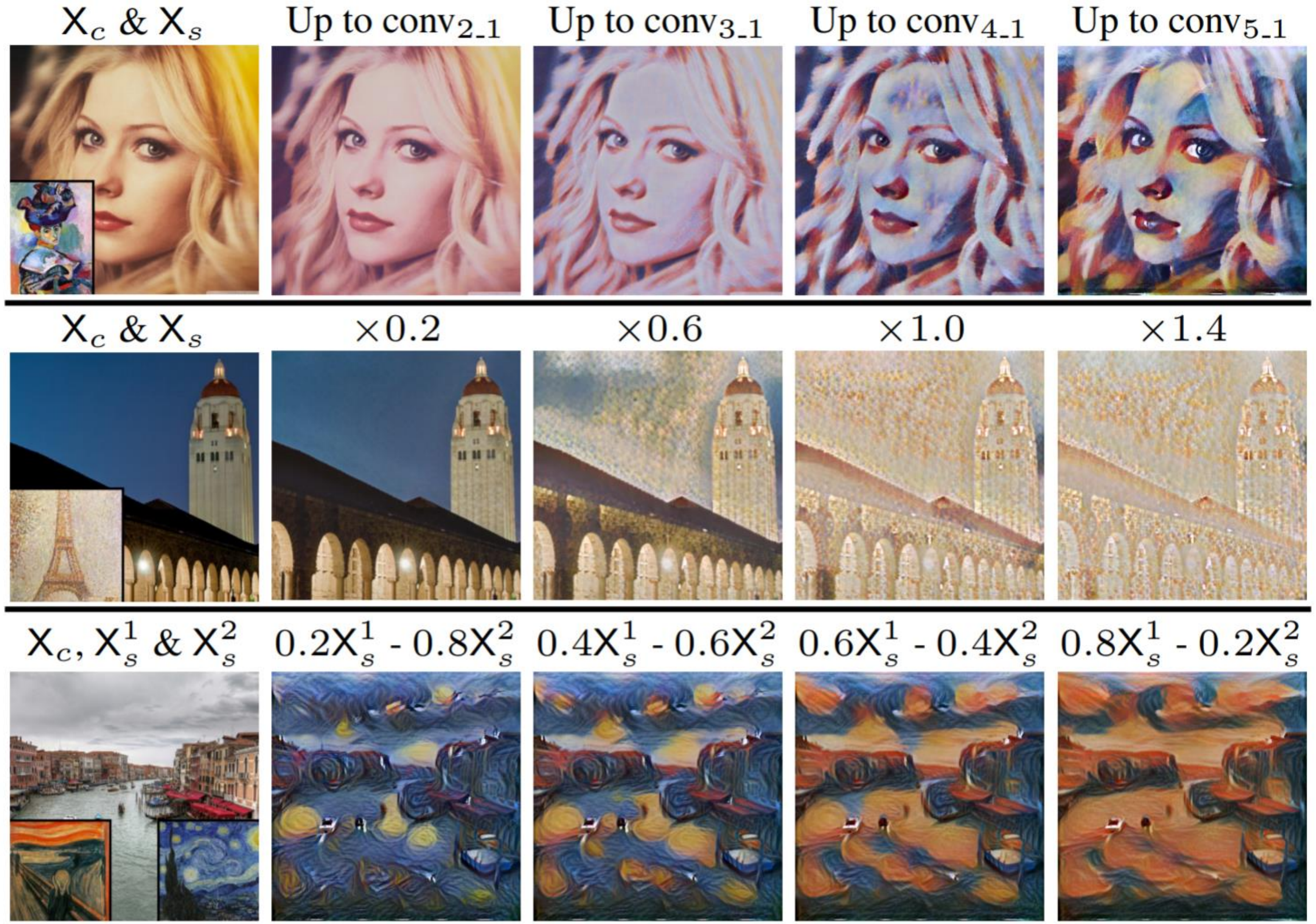
I_{cc} , I_{ss} : output image synthesized from two same content(or style) image

φ_i : layer in the encoder

11

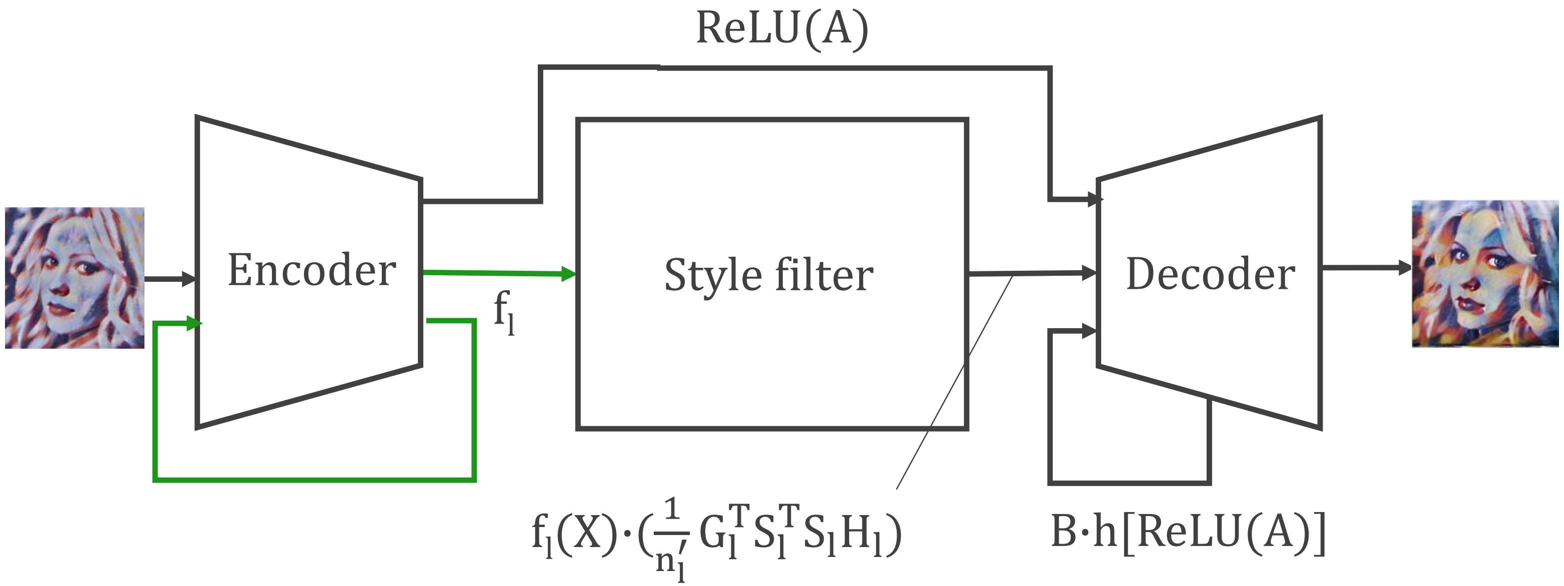
A Flexible Convolutional Solver for Fast Style Transfers

Gilles Puy, Patrick Perez, CVPR, 2019



A Flexible Convolutional Solver for Fast Style Transfers

Gilles Puy, Patrick Perez, CVPR, 2019



$$f_\ell(X)\cdot\left(\frac{1}{n'_\ell}G_\ell^T S_\ell^T S_\ell H_\ell\right) \rightarrow \frac{1}{n'_\ell} f_\ell(X)\cdot\left[G_\ell^T\left(\sum_i \alpha_i S_\ell^T[X_s^i]S_\ell[X_s^i]\right)H_\ell\right]. \quad (12)$$

Style filter : style control을 위한 필터로 각 scale에서 특성 f_l을 독립적으로 필터링하여 제어 가능함

A Flexible Convolutional Solver for Fast Style Transfers

Gilles Puy, Patrick Perez, CVPR, 2019

$$\boxed{\nabla L_s = \sum_{l \in I_s} \lambda_s^l \nabla L_s^l}$$

$$g(\mathsf{X}, \mathsf{X}_s) = \sum_{\ell \in \mathcal{I}_s} \lambda_s^\ell \, b_\ell \left(\frac{1}{n'_\ell} f_\ell(\mathsf{X}) \times \mathsf{G}_\ell^\intercal \, \mathsf{S}_\ell^\intercal \mathsf{S}_\ell \, \mathsf{H}_\ell \right). \quad (10)$$

$$\sum_{\ell \in \mathcal{I}_s} \frac{\lambda_s^\ell}{c_\ell^2} \sum_i \left\| \frac{1}{n_\ell} (\mathsf{M}_\ell^i \mathsf{F}_\ell)^\intercal (\mathsf{M}_\ell^i \mathsf{F}_\ell) - \frac{1}{n'_\ell} \mathsf{S}_\ell^\intercal [\mathsf{X}_s^i] \mathsf{S}_\ell [\mathsf{X}_s^i] \right\|_{\mathrm{F}}^2 .$$

** top : Archetecture of g_t*

** bottom : Runtime modification of the style loss*

$G_l^T S_l^T S_l H_l$: style filter, governs the style that is applied to the input image (S_l : style feature)

$f_l(X)$: l layer feature

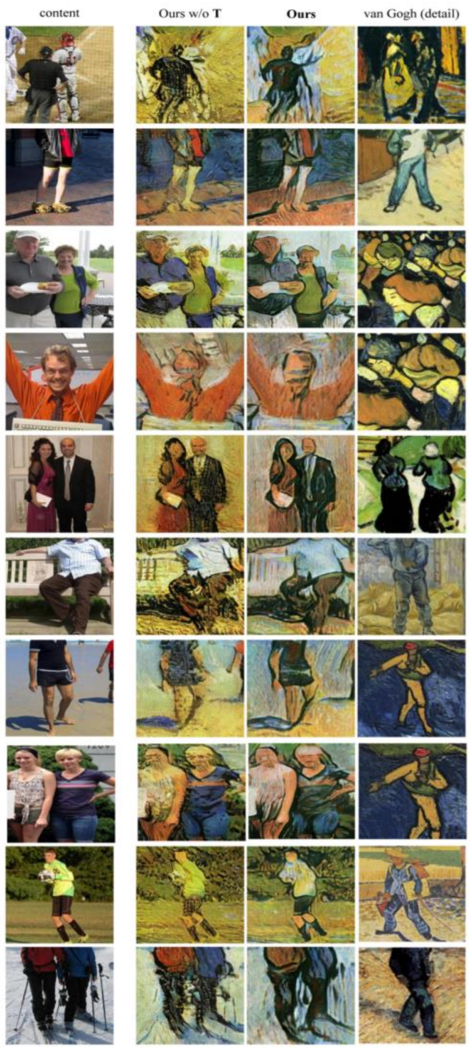
M_l^i : mask at layer l for the ith style and ith region

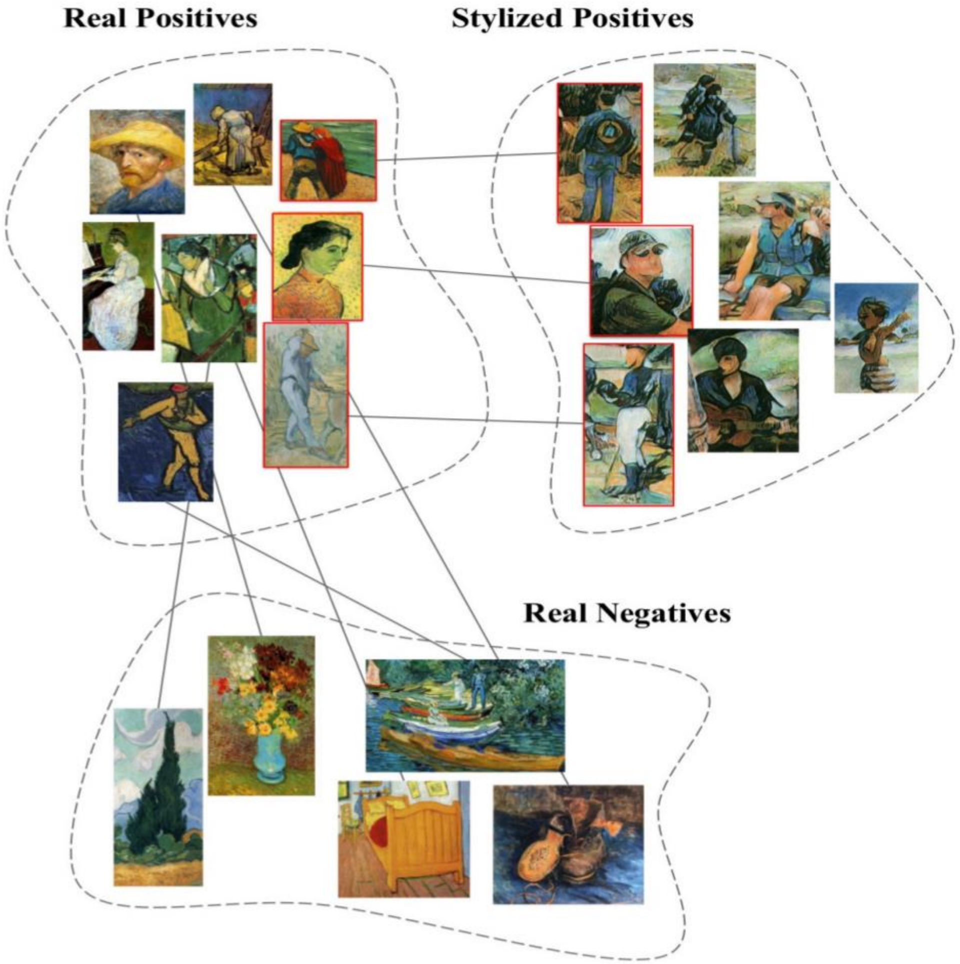
F_l : result feature of $f_l(X)$

X : image

12 A Content Transformation Block for Image Style Transfer

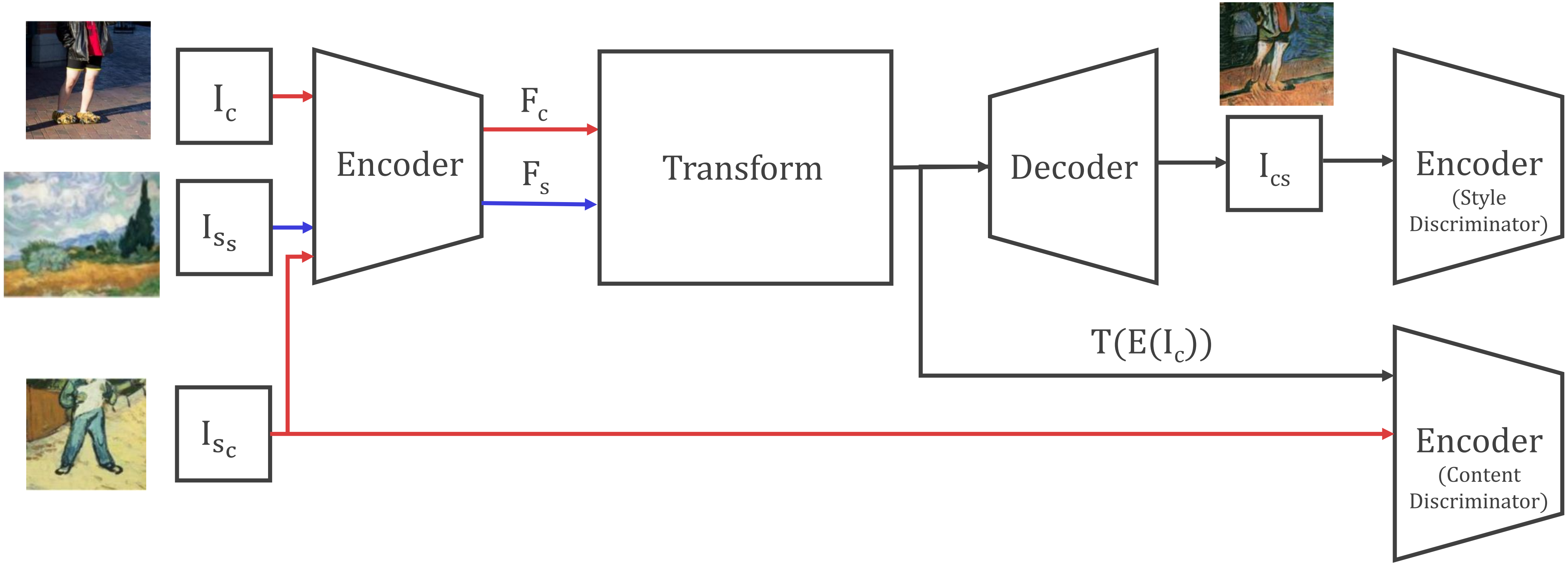
Dmytro Kotovenko, Artsiom Sanakoyeu, Pingchuan Ma, Sabine Lang, Bjorn Ommer, CVPR, 2019





12 A Content Transformation Block for Image Style Transfer

Dmytro Kotovenko, Artsiom Sanakoyeu, Pingchuan Ma, Sabine Lang, Bjorn Ommer, CVPR, 2019



Transform : 스타일의 부분적인 특징에 따라 content 를 변환하도록 학습

A Content Transformation Block for Image Style Transfer

Dmytro Kotovenko, Artsiom Sanakoyeu, Pingchuan Ma, Sabine Lang, Bjorn Ommer, CVPR, 2019

$$\text{Step 1}: \min_{\theta_E,\theta_D} \max_{\theta_{E(D)_S}} \lambda_{L_{pxl}} L_{pxl} + \lambda_{L_{FP}} L_{FP} + \lambda_{L_{adv-style}} L_{adv-style}$$

$$\text{Step 2}: \min_{\theta_T} \max_{\theta_{D_C}} \lambda_{L_{adv-cont}} L_{adv-cont} + \lambda_{L_{adv}} L_{adv-style}$$

$$\mathcal{L}_{adv-style} := \mathop{\mathbb{E}}_{(y,c)\sim\mathbf{Y}}[log(\mathcal{D}_s(y))] + \mathop{\mathbb{E}}_{(x,c,s)\sim\mathbf{X}}[log\left(1-\mathcal{D}_s\left(D(\mathcal{T}(E(x)))|s\right)\right)] \quad (8)$$

$$\mathcal{L}_{adv-cont} := \mathop{\mathbb{E}}_{(y,c)\sim\mathbf{Y}}[log(\mathcal{D}_c(E(y)|c))] + \mathop{\mathbb{E}}_{(x,c,s)\sim\mathbf{X}}[log\left(1-\mathcal{D}_c\left(\mathcal{T}(E(x))|c\right)\right)] \quad (9)$$

$$\mathcal{L}_{FP} := \mathop{\mathbb{E}}_{(x,c,s)\sim\mathbf{X}}[\|E(D(\mathcal{T}(E(x)))) - E(x)\|_2^2]. \quad (10)$$

Y : Style Image

X : Photo Image(Content)

L_{pxl} : Pixel similarity

L_{FP} : Fix Point loss (Residual of encoding space)

$L_{adv\text{-}cont}$: Adversarial loss of content

$L_{adv\text{-}style}$: Adversatial loss of style

c : content class

s : scene class

Thank You