DEVIEW 2019

신호처리 이론으로

실용적인 스타일 변환 모델 만들기

Style transfer your image in **"photographic way"**, e.g., day2sunset.

Code, generated images, and pre-trained models are all available at github.com/clovaai/WCT2

유재준
Search & Clova AI
NAVER

스타일 변환

이미지에서 스타일은 뭘까?

반대로 컨텐츠는 뭐지?

그리고 어떻게 옮기는데?

스타일 변환? Artistic

Gatys et al. CVPR '16

이미지를 "예술적"으로 변환하는 문제 (예: 사진을 고흐풍 그림으로)

스타일 변환? Photorealistic

이미지를 “사실적”으로 변환하는 문제 (예: 낮에서 저녁, 낮에서 밤)

왜 중요한데?

스타일 변환과 관련된 연구 주제들

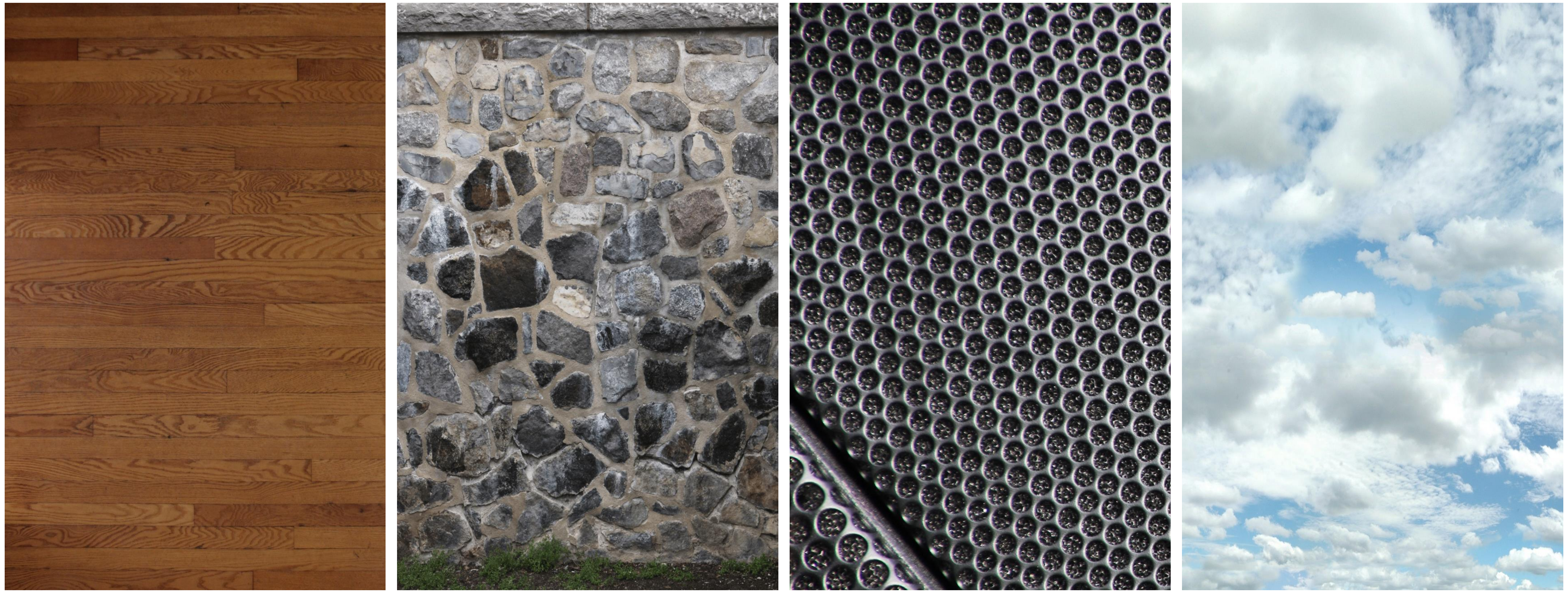
Style Transfer	Style Transfer
•Unsupervised Learning	•Domain Translation
•Representation Learning	•Domain Adaptation
•Feature extraction	•Domain Augmentation

Style Transfer: Generative model

옛날에는 어떻게 했을까?

질감 생성 (Texture Synthesis)

“질감이란 일반적으로 이미지 전반에 걸쳐 색이나 방향, 크기, 위치가 조금씩 임의로 바뀌면서 반복되는 간단한 이미지 요소(elements)라고 정의할 수 있다.”



www.aarondpate.com (CC0)

옛날에는 어떻게 했을까?

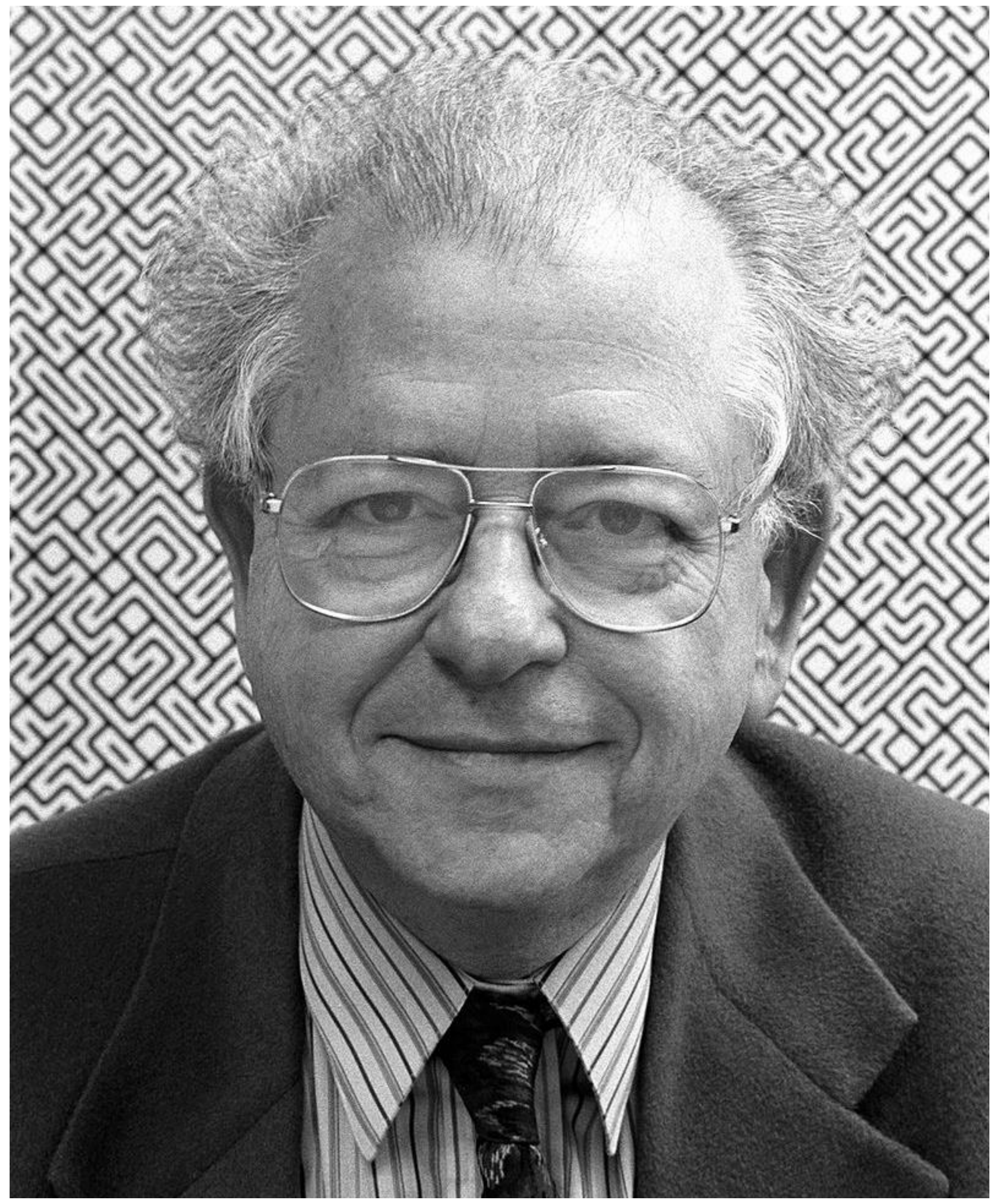
Julesz Conjecture (IRE Information Theory, 1962)

"임의의 질감(texture)이 두 개가 있을 때, 이 둘의 2nd-order statistics까지만 같아도 사람은 이 둘을 구별하지 못한다."

Texton (Nature, 1982)

"**어떤 선형 필터들의 조합으로 표현**되는 인간이 질감을 인지하는 최소 단위."

1. **The theory of Markov random fields**
2. **The use of oriented linear kernels** (e.g., multi-scale wavelets)



Béla Julesz

visual neuroscientist &
experimental psychologist

최초로 CNNs을 사용한 연구

- Gatys *et al.*, NIPS 2015

Texture Synthesis Using Convolutional Neural Networks

Leon A. Gatys
Centre for Integrative Neuroscience, University of Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
Graduate School of Neural Information Processing, University of Tübingen, Germany
leon.gatys@bethgelab.org

Alexander S. Ecker
Centre for Integrative Neuroscience, University of Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
Max Planck Institute for Biological Cybernetics, Tübingen, Germany
Baylor College of Medicine, Houston, TX, USA

Matthias Bethge
Centre for Integrative Neuroscience, University of Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
Max Planck Institute for Biological Cybernetics, Tübingen, Germany

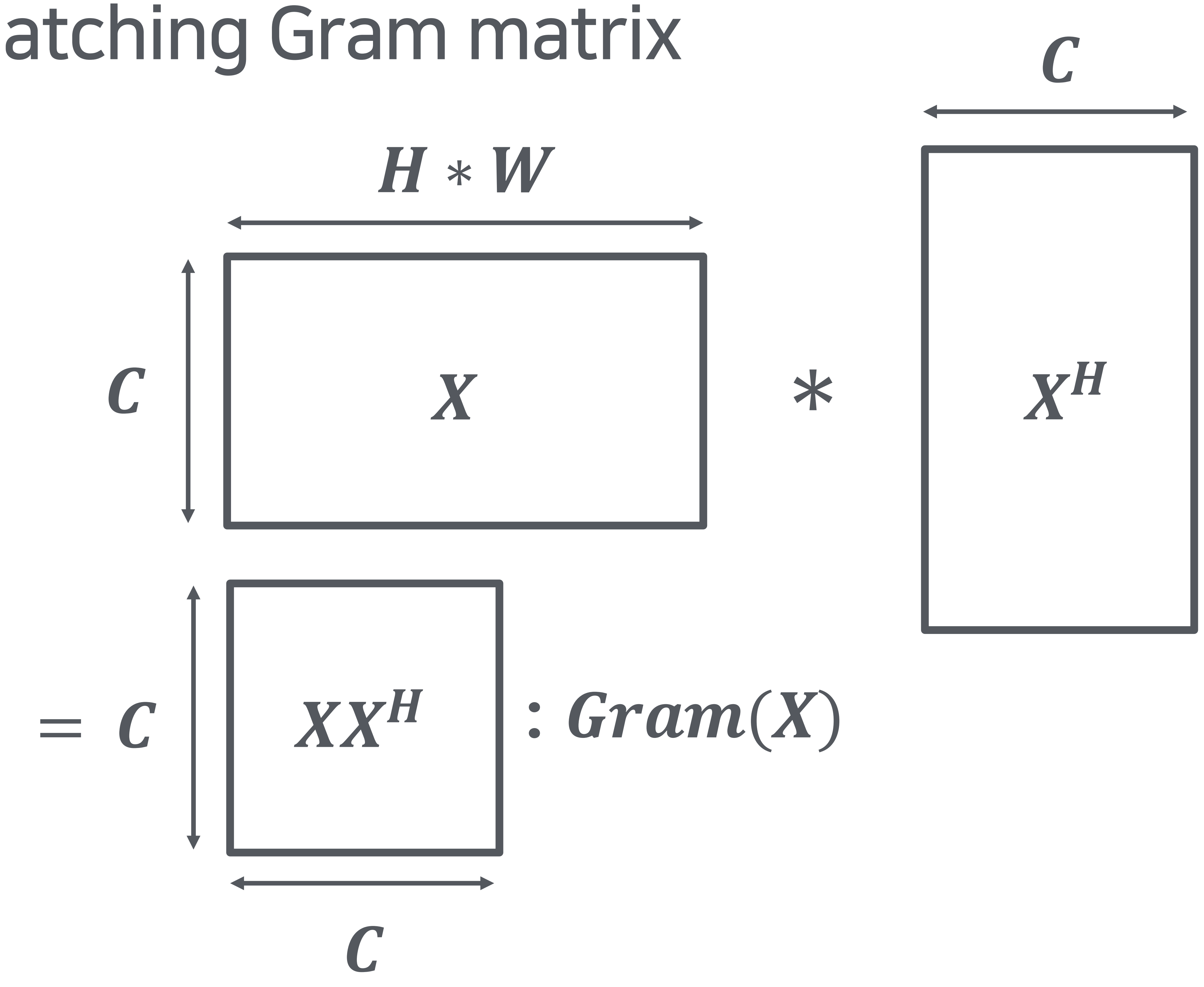
Abstract

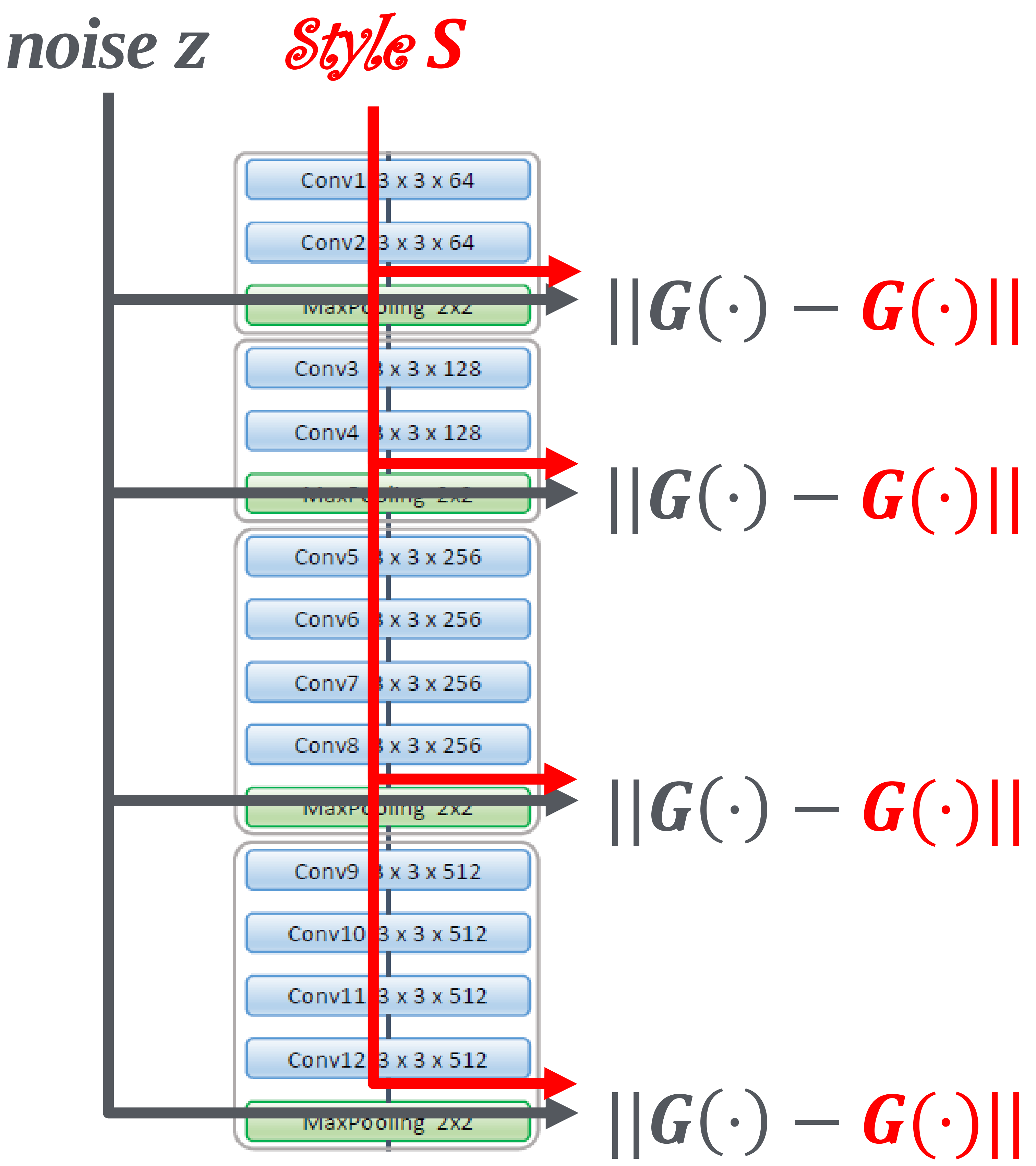
Here we introduce a new model of natural textures based on the feature spaces of convolutional neural networks optimised for object recognition. Samples from the model are of high perceptual quality demonstrating the generative power of neural networks trained in a purely discriminative fashion. Within the model, textures are represented by the correlations between feature maps in several layers of the network. We show that across layers the texture representations increasingly capture the statistical properties of natural images while making object information more and more explicit. The model provides a new tool to generate stimuli for neuroscience and might offer insights into the deep representations learned by convolutional neural networks.

Neural Style Algorithm

DEVIEW 2019

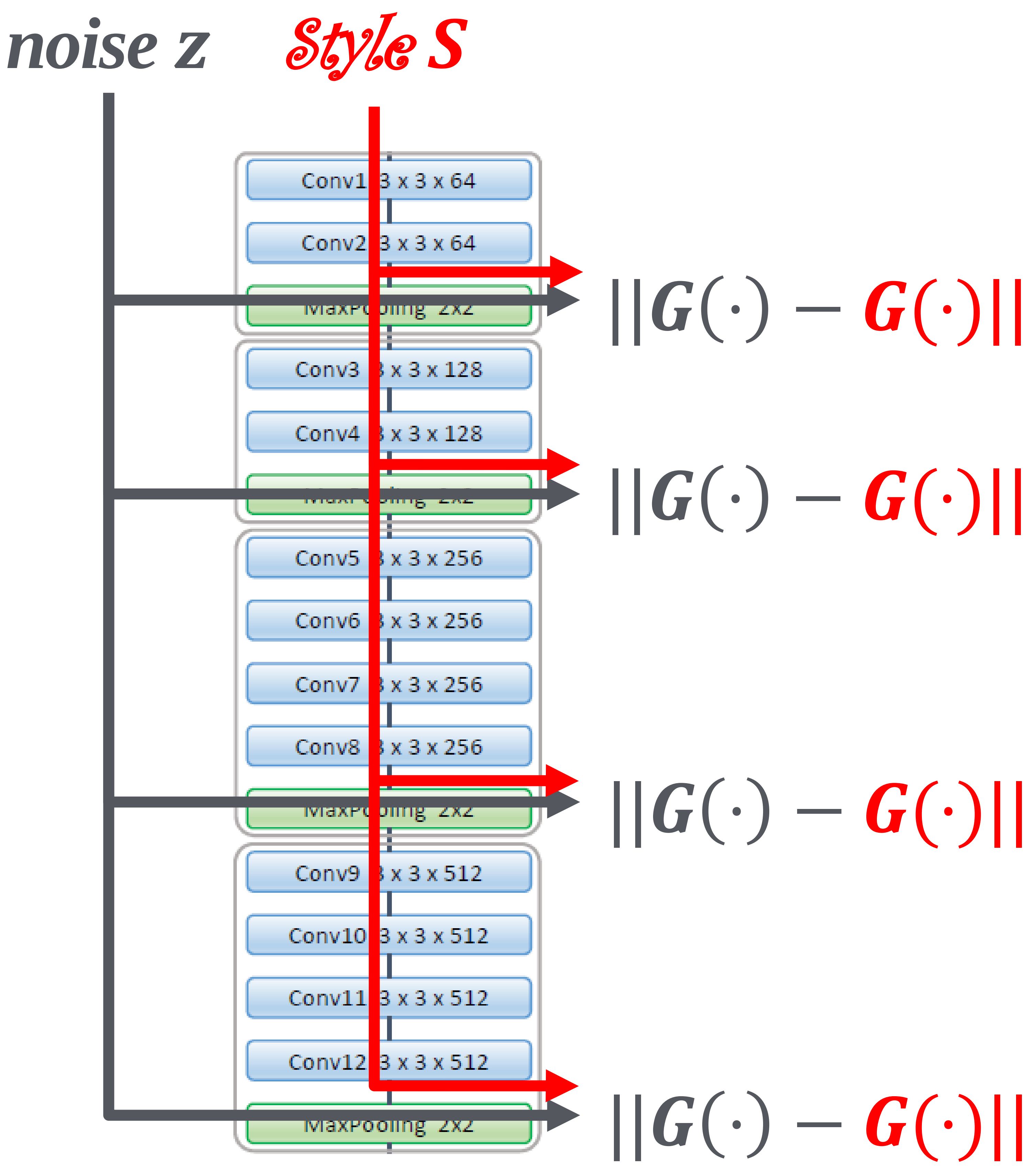
- Matching Gram matrix





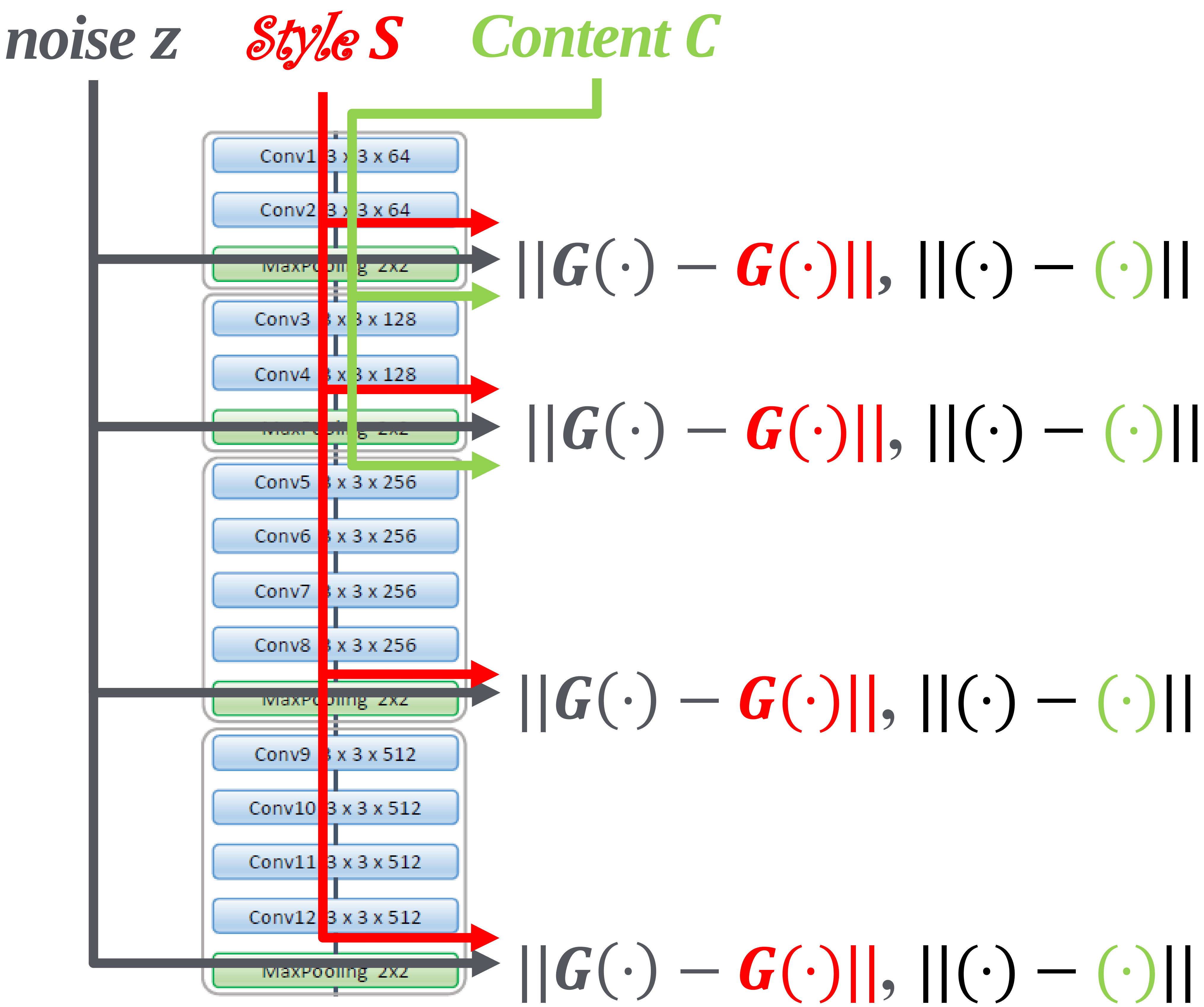
Neural Style Algorithm

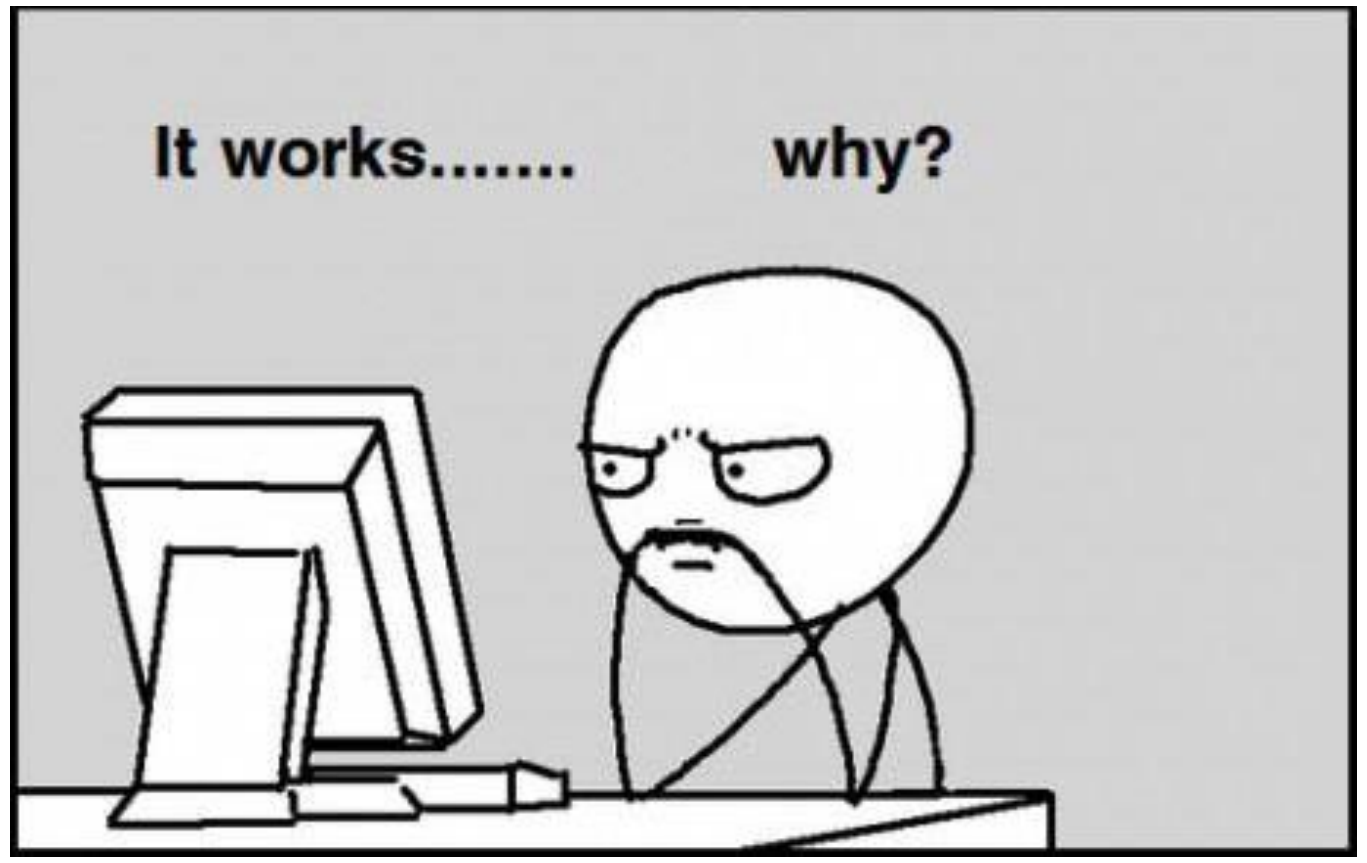
- Gatys *et al.*, NIPS 2015



Neural Style Algorithm

- Gatys *et al.*, CVPR 2016





이게 되네 ... 왜 때문...?

Theoretical analysis on Style Transfer

Minimizing Maximum Mean Discrepancy (MMD)

Demystifying Neural Style Transfer

Yanghao Li[†] **Naiyan Wang**[‡] **Jiaying Liu**[†*] **Xiaodi Hou**[‡]
[†] Institute of Computer Science and Technology, Peking University
[‡] TuSimple
lyttonhao@pku.edu.cn winsty@gmail.com liujiaying@pku.edu.cn xiaodi.hou@gmail.com

Abstract

Neural Style Transfer [Gatys *et al.*, 2016] has recently demonstrated very exciting results which catches eyes in both academia and industry. Despite the amazing results, the principle of neural style transfer, especially why the Gram matrices could represent style remains unclear. In this paper, we propose a novel interpretation of neural style transfer by treating it as a domain adaptation problem. Specifically, we theoretically show that matching the Gram matrices of feature maps is equivalent to minimize the Maximum Mean Discrepancy (MMD) with the second order polynomial kernel. Thus, we argue that the essence of neural style transfer is to match the feature distributions between the style images and the generated images. To further support our standpoint, we experiment with several other distribution alignment methods, and achieve appealing results. We believe this novel interpretation connects these two important research fields, and could enlighten future researches.

1 Introduction

Transferring the style from one image to another image is an interesting yet difficult problem. There have been many efforts to develop efficient methods for automatic style transfer [Hertzmann *et al.*, 2001; Efros and Freeman, 2001; Efros and Leung, 1999; Shih *et al.*, 2014; Kwatra *et al.*, 2005]. Recently, Gatys *et al.* proposed a seminal work [Gatys *et al.*, 2016]: It captures the style of artistic images and transfer it to other images using Convolutional Neural Networks (CNN). This work formulated the problem as finding an image that matching both the content and style statistics based on the neural activations of each layer in CNN. It achieved impressive results and several follow-up works improved upon this innovative approaches [Johnson *et al.*, 2016; Ulyanov *et al.*, 2016; Ruder *et al.*, 2016; Ledig *et al.*, 2016]. Despite the fact that this work has drawn lots of attention, the fundamental element of style representation: the Gram matrix in [Gatys *et al.*, 2016] is not fully explained. The reason why Gram matrix can represent artistic style still remains a mystery.

In this paper, we propose a novel interpretation of neural style transfer by casting it as a special domain adaptation [Beijbom, 2012; Patel *et al.*, 2015] problem. We theoretically prove that matching the Gram matrices of the neural activations can be seen as minimizing a specific Maximum Mean Discrepancy (MMD) [Gretton *et al.*, 2012a]. This reveals that neural style transfer is intrinsically a process of distribution alignment of the neural activations between images. Based on this illuminating analysis, we also experiment with other distribution alignment methods, including MMD with different kernels and a simplified moment matching method. These methods achieve diverse but all reasonable style transfer results. Specifically, a transfer method by MMD with linear kernel achieves comparable visual results yet with a lower complexity. Thus, the second order interaction in Gram matrix is not a must for style transfer. Our interpretation provides a promising direction to design style transfer methods with different visual results. To summarize, our contributions are shown as follows:

1. First, we demonstrate that matching Gram matrices in neural style transfer [Gatys *et al.*, 2016] can be reformulated as minimizing MMD with the second order polynomial kernel.
2. Second, we extend the original neural style transfer with different distribution alignment methods based on our novel interpretation.

2 Related Work

In this section, we briefly review some closely related works and the key concept MMD in our interpretation.

Style Transfer Style transfer is an active topic in both academia and industry. Traditional methods mainly focus on the non-parametric patch-based texture synthesis and transfer, which resamples pixels or patches from the original source texture images [Hertzmann *et al.*, 2001; Efros and Freeman, 2001; Efros and Leung, 1999; Liang *et al.*, 2001]. Different methods were proposed to improve the quality of the patch-based synthesis and constrain the structure of the target image. For example, the image quilting algorithm based on dynamic programming was proposed to find optimal texture

*Corresponding author

- Li *et al.*, IJCAI 2017

TL;DR

"Gram 행렬을 맞추는 것 = 2nd-order polynomial kernels를 사용해서 MMD 최소화하는 것"

By using the second order degree polynomial kernel $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y})^2$, Eq. 8 can be represented as:

$$\mathcal{L}^l_{style} = \frac{1}{4N_l^2 M_l^2} \sum_{k_1=1}^{M_l} \sum_{k_2=1}^{M_l} \Big(k(\mathbf{f}^l_{\cdot k_1}, \mathbf{f}^l_{\cdot k_2}) + k(\mathbf{s}^l_{\cdot k_1}, \mathbf{s}^l_{\cdot k_2}) - 2k(\mathbf{f}^l_{\cdot k_1}, \mathbf{s}^l_{\cdot k_2}) \Big)$$
$$= \frac{1}{4N_l^2} \mathrm{MMD}^2[\mathcal{F}^l, \mathcal{S}^l],$$

왜 하필 VGG만...?

DEVIEW
2019

[D] Eat Your VGGtables, or, Why Does Neural Style Transfer Work Best With Old VGG CNNs' Features?

reddit MACHINELEARNING 댓글

This is an archived post. You won't be able to vote or comment.

170

Discussion [D] Eat Your VGGtables, or, Why Does Neural Style Transfer Work Best With Old VGG CNNs' Features? (self.MachineLearning)

submitted 1 년 전에 * by gwern

> Previous: Twitter discussion.

An acquaintance a year or two ago was messing around with neural style transfer (Gatys et al 2016), experimenting with some different approaches, like a tile-based GPU implementation for making large poster-size transfers, or optimizing images to look different using a two-part loss: one to encourage being like the style of the style image, and a negative one to penalize having content like the source image; this is unstable and can diverge, but when it works, looks cool. (Example: "The Great Wave" + Golden Gate Bridge. I tried further Klimt-ising it but at that point too much has been lost.)

VGG worked best for style transfer

One thing they noticed was that using features from a pretrained ImageNet VGG-16/19 CNN from 2014 (4 years ago), like the original Gatys paper did, worked much better than anything else; indeed, almost any set of 4-5 layers in VGG would provide great features for the style transfer optimization to target (as long as they were spread out and weren't exclusively bottom or top layers), while using more modern resnets (resnet-50) or GoogLeNet Inception v1 didn't work - it was hard to find sets of layers that would work at all and when they did, the quality of the style transfer was not as good. Interestingly, this appeared to be true of VGG CNNs trained on the MIT Places scene recognition database too, suggesting there's something architectural going on which is not database specific or peculiar to those two trained models. And their attempt at an upscaling CNN modeled on Johnson et al 2016's VGG-16 for CIFAR-100 worked well too.

Everyone uses VGG

CNNs이 “texture”에 편향되어 있더라...

DEVIEW 2019

IMAGENET-TRAINED CNNS ARE BIASED TOWARDS TEXTURE; INCREASING SHAPE BIAS IMPROVES ACCURACY AND ROBUSTNESS

Robert Geirhos
University of Tübingen & IMPRS-IS
robert.geirhos@bethgelab.org

Patricia Rubisch
University of Tübingen & U. of Edinburgh
p.rubisch@sms.ed.ac.uk

Claudio Michaelis
University of Tübingen & IMPRS-IS
claudio.michaelis@bethgelab.org

Matthias Bethge*
University of Tübingen
matthias.bethge@bethgelab.org

Felix A. Wichmann*
University of Tübingen
felix.wichmann@uni-tuebingen.de

Wieland Brendel*
University of Tübingen
wieland.brendel@bethgelab.org

ABSTRACT

Convolutional Neural Networks (CNNs) are commonly thought to recognise objects by learning increasingly complex representations of object shapes. Some recent studies suggest a more important role of image textures. We here put these conflicting hypotheses to a quantitative test by evaluating CNNs and human observers on images with a texture-shape cue conflict. We show that ImageNet-trained CNNs are strongly biased towards recognising textures rather than shapes, which is in stark contrast to human behavioural evidence and reveals fundamentally different classification strategies. We then demonstrate that the same standard architecture (ResNet-50) that learns a texture-based representation on ImageNet is able to learn a shape-based representation instead when trained on 'Stylized-ImageNet', a stylized version of ImageNet. This provides a much better fit for human behavioural performance in our well-controlled psychophysical lab setting (nine experiments totalling 48,560 psychophysical trials across 97 observers) and comes with a number of unexpected emergent benefits such as improved object detection performance and previously unseen robustness towards a wide range of image distortions, highlighting advantages of a shape-based representation.

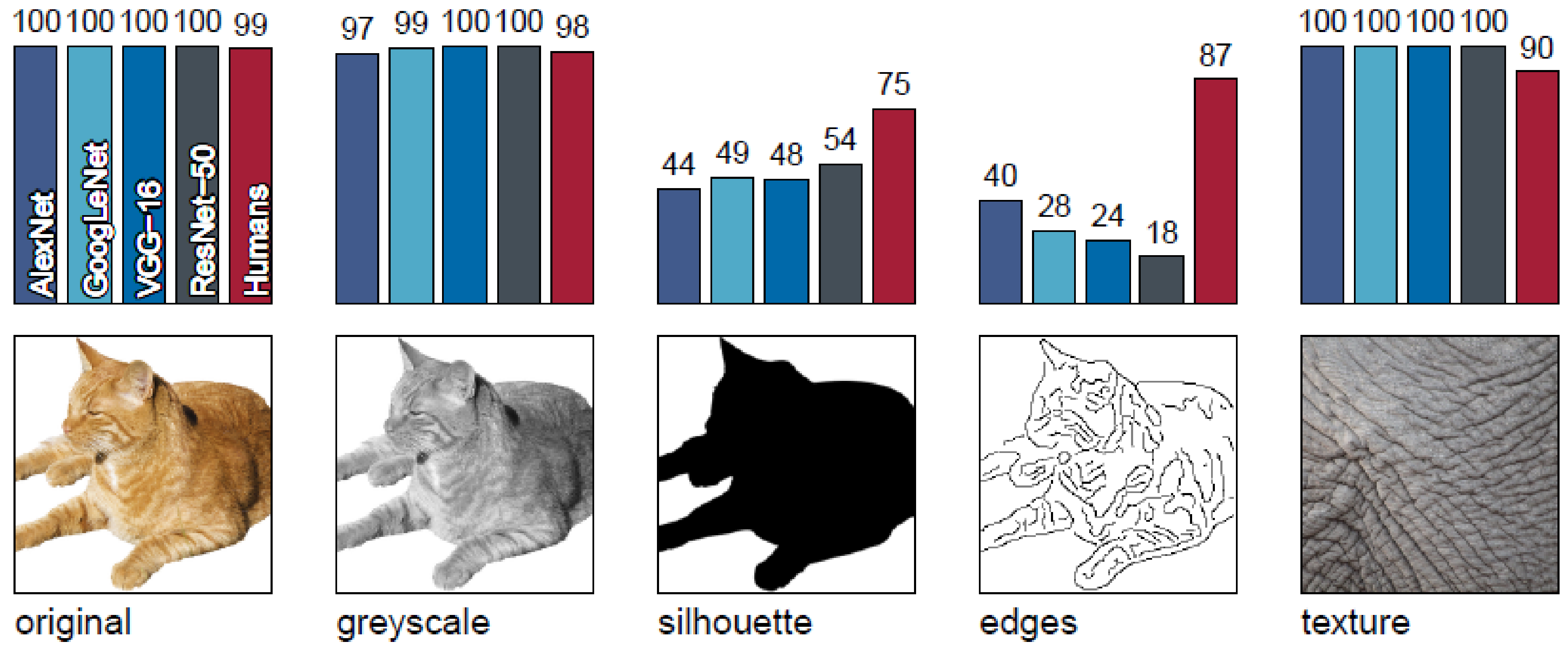
(a) Texture image
81.4% **Indian elephant**
10.3% indri
8.2% black swan

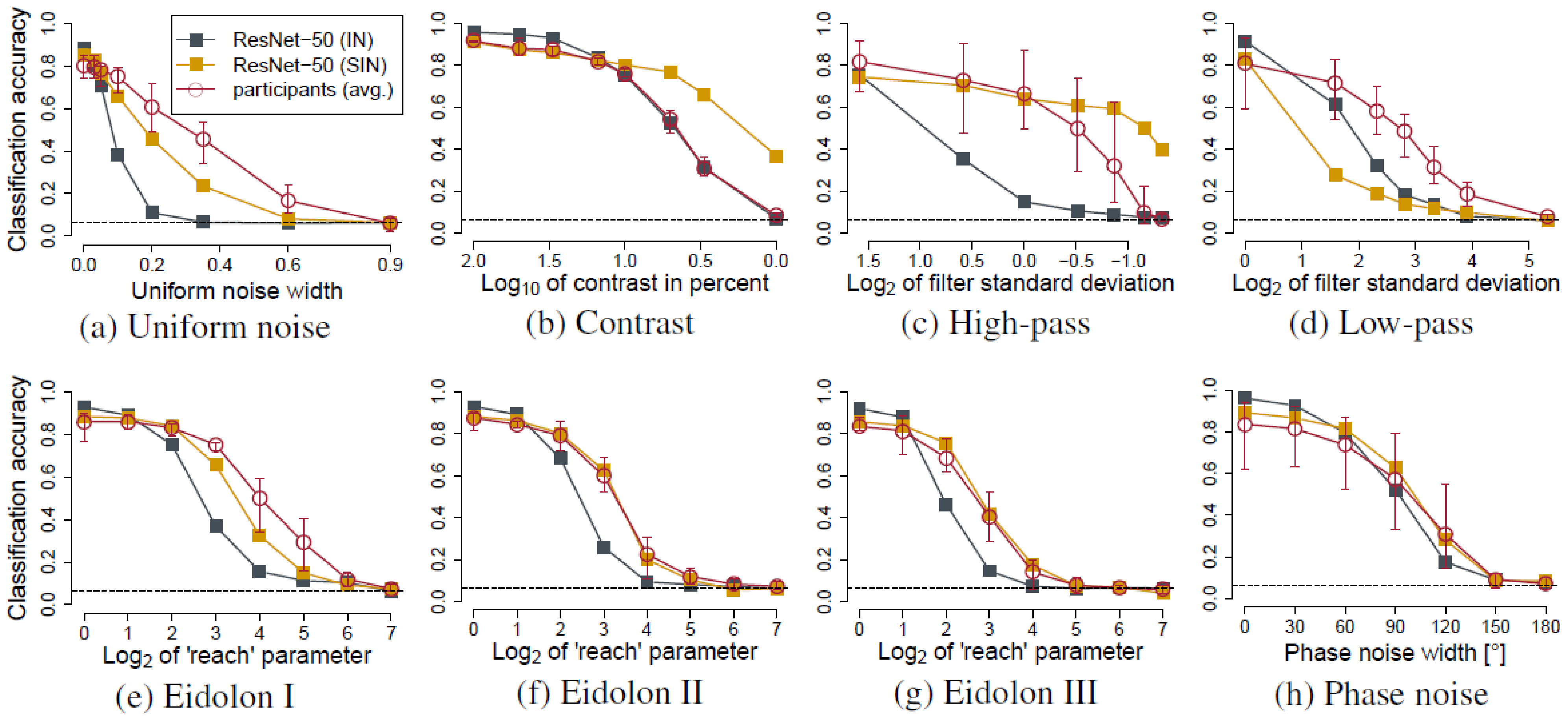
(b) Content image
71.1% **tabby cat**
17.3% gray fox
3.3% Siamese cat

(c) Texture-shape cue conflict
63.9% **Indian elephant**
26.4% indri
9.6% black swan

Figure 1: Classification of a standard ResNet-50 of **(a)** a texture image (elephant skin: only texture cues); **(b)** a normal image of a cat (with both shape and texture cues), and **(c)** an image with a texture-shape cue conflict, generated by style transfer between the first two images.

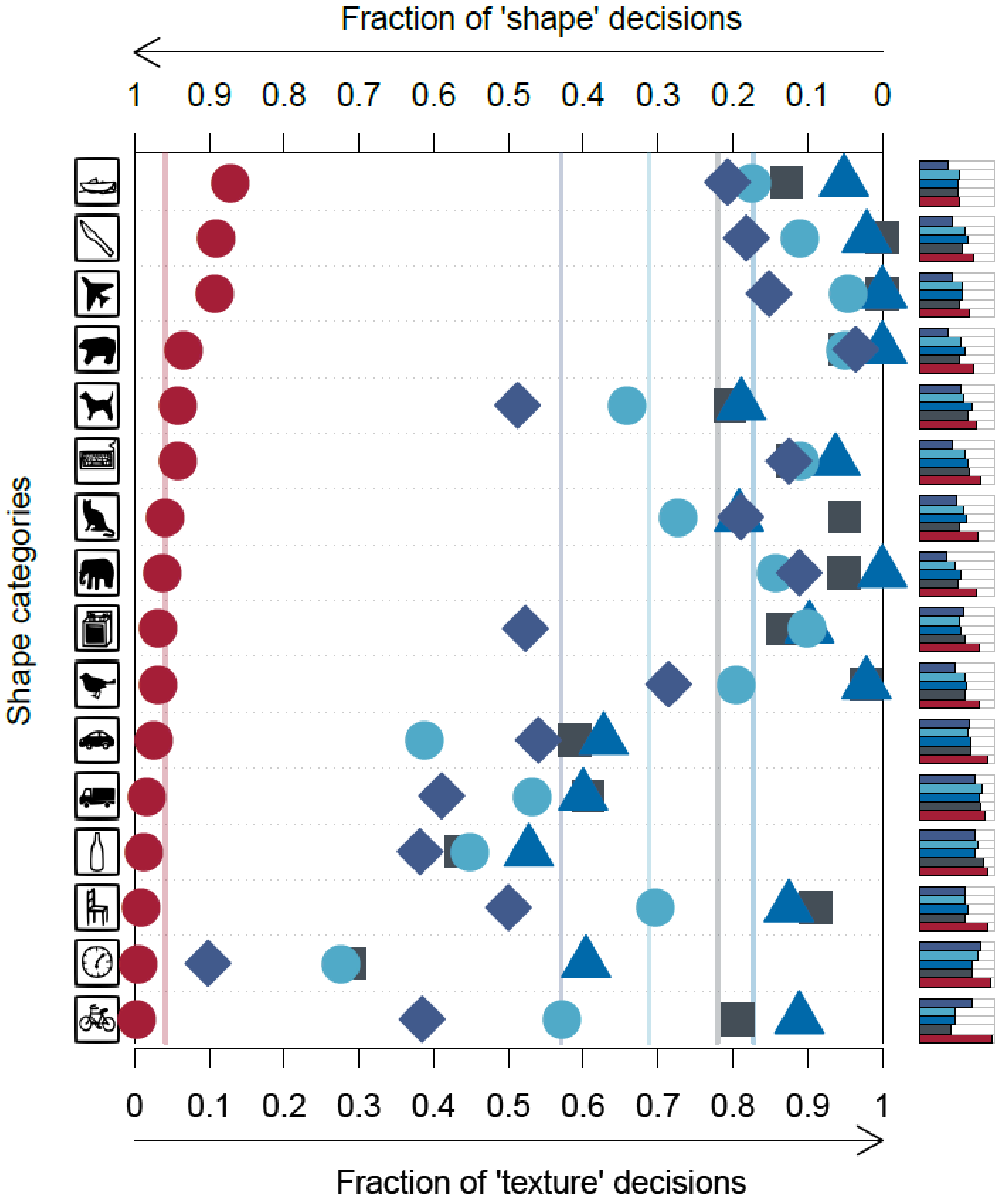
*Joint senior authors





Gerihos *et al.*, ICLR 2019 / ratings: **7, 8, 8**

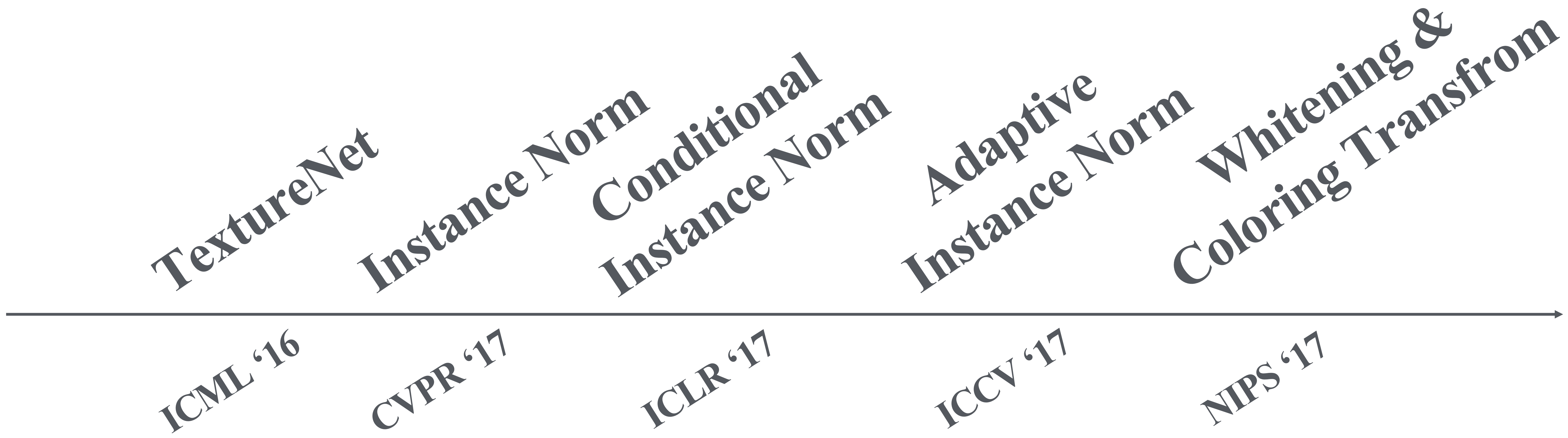
CNNs이 “texture”에 편향되어 있더라...



“Human observers show a striking bias towards responding with the shape category (95.9% of correct decisions).”

“This pattern is reversed for CNNs, which show a clear bias towards responding with the texture category (**VGG-16: 17.2% shape vs. 82.8% texture**; GoogLeNet: 31.2% vs. 68.8%; AlexNet: 42.9% vs. 57.1%; ResNet-50: 22.1% vs. 77.9%).”

*** ▲: VGG-16**

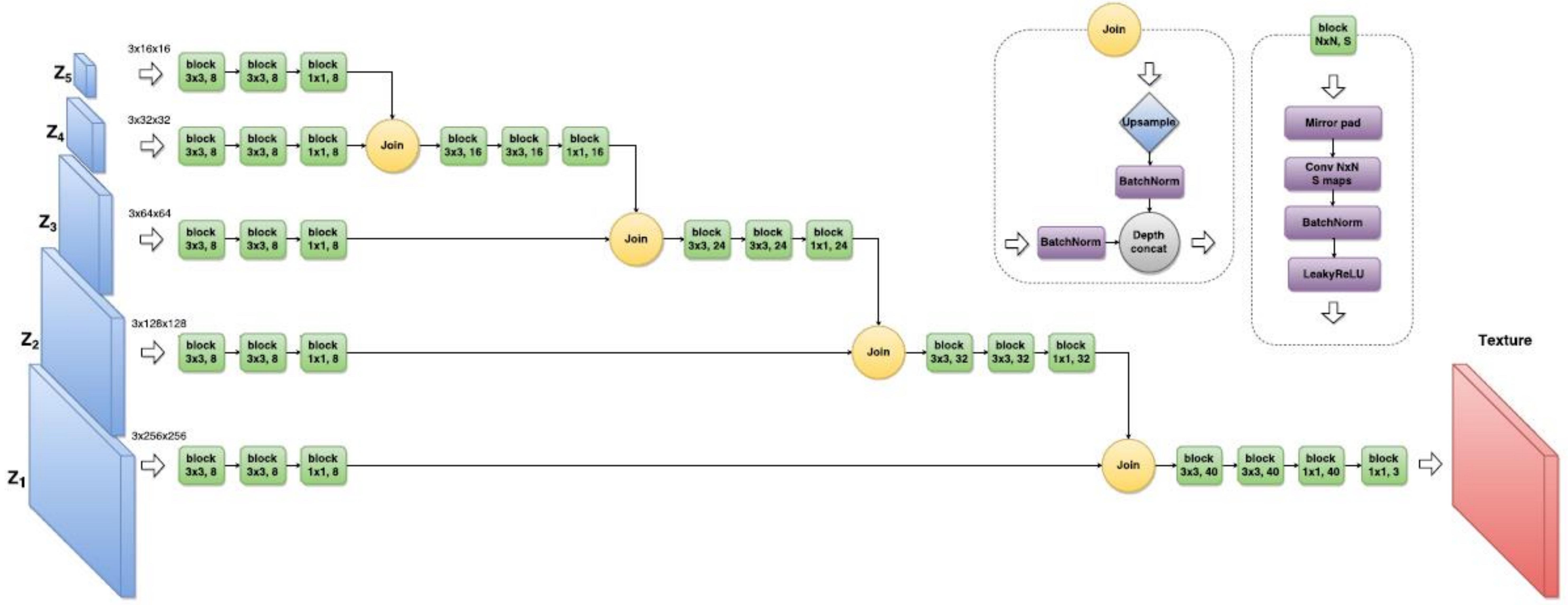


최신 연구 흐름 한눈에 살펴보기

1. Artistic Style Transfer
2. Photorealistic Style Transfer

Feed-forward network

"최적화 문제 푸는게 느리니까 **그 결과 자체를 학습**시켜버리자!"



$$\theta_{x_0} = \underset{\theta}{\operatorname{argmin}} \; E_{z\sim\mathcal{Z}} \left[\mathcal{L}_T \left(g(z;\theta), x_0 \right) \right].$$

$$\mathcal{L}_T(x; x_0) = \sum_{l \in L_T} \| G^l(x) - G^l(x_0) \|_2^2$$

Feed-forward network

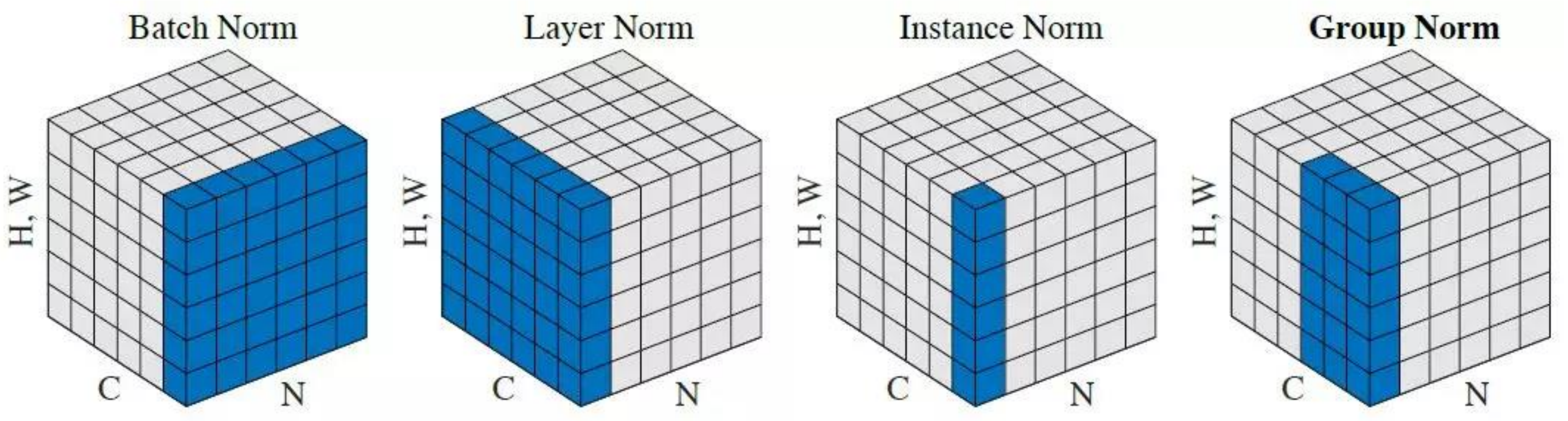
- 하나의 스타일마다 네트워크 하나씩 학습 필요

- from Unyanov *et al.*, ICML 2016

Content | Texture nets (ours) | Gatys et al. | Style

Instance Normalization (IN)

"스타일 변환 결과가 컨텐츠 이미지의 Contrast에 따라 바뀌지 않게 하자"



$$\mathrm{IN}(x) = \gamma \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \beta$$

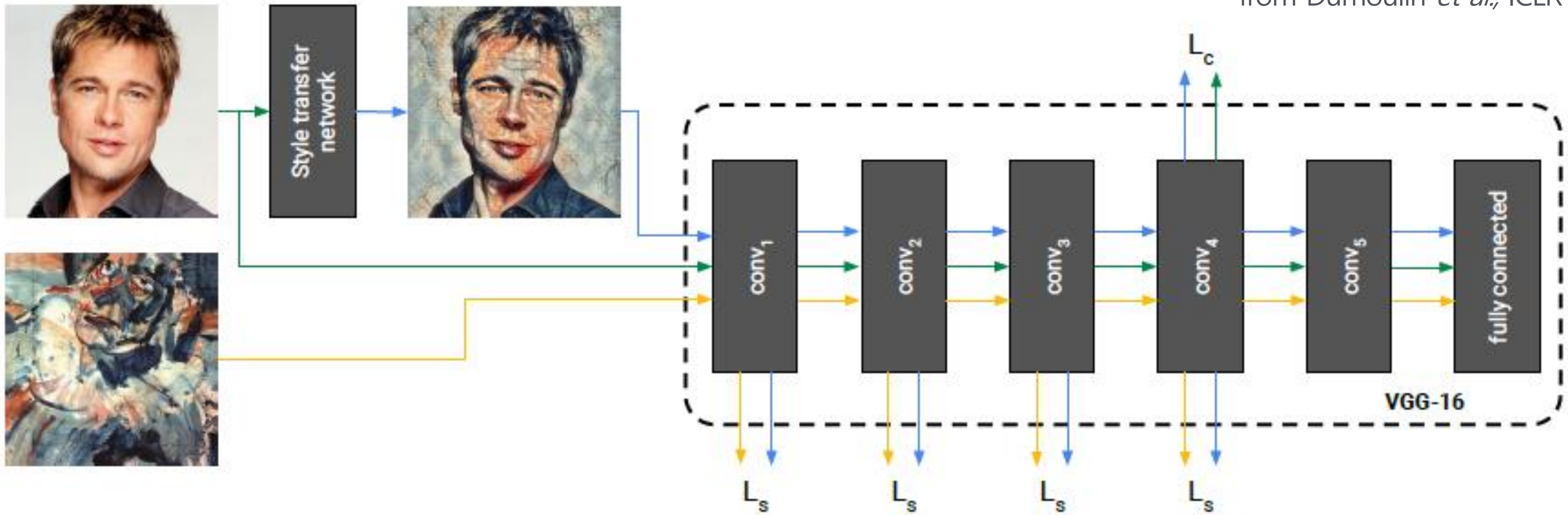
- from Unyanov *et al.*, CVPR 2017

Conditional Instance Normalization (CIN)

DEVIEW 2019

"정규화 된 이미지를 스타일 값으로 shifting & scaling만 해도 스타일이 먹히네?"

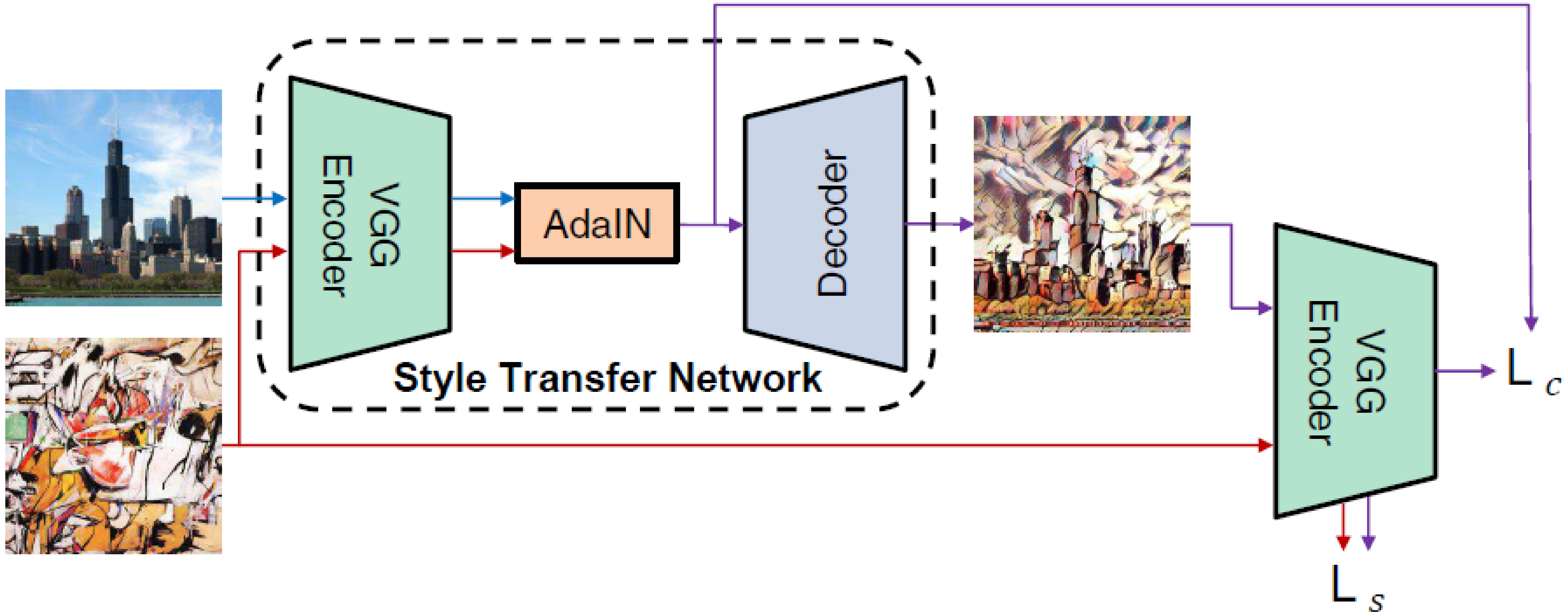
- from Dumoulin *et al.*, ICLR 2017



$$\mathrm{CIN}(x; s) = \gamma^s \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \beta^s$$

Adaptive Instance Normalization (AdaIN)

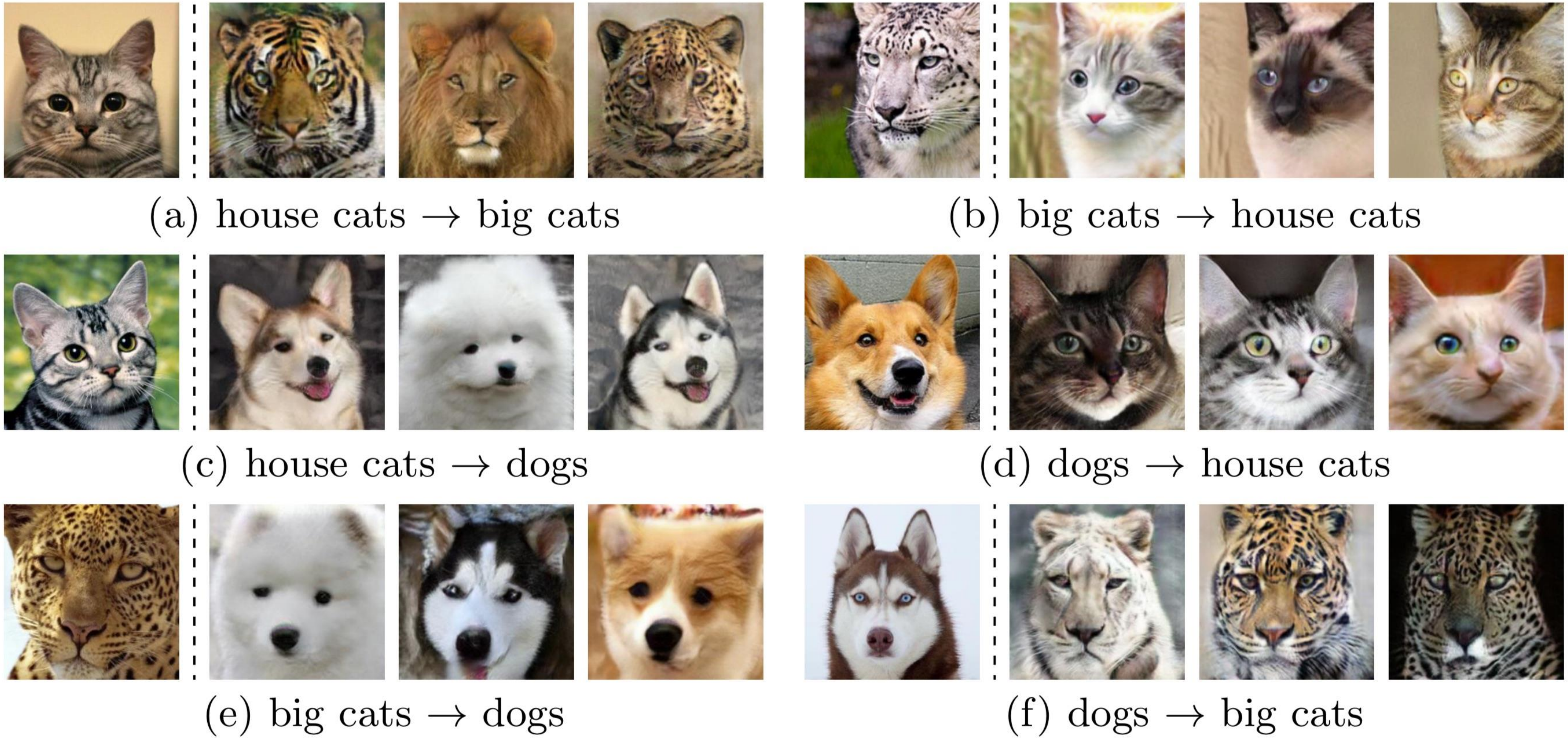
"Scaling이랑 Shifting 값을 구하는 과정을 아예 학습과 분리하면 어떨까?"



$$\mathrm{AdaIN}(x, y) = \sigma(y)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y)$$

- from Huang *et al.*, ICCV 2017

Adaptive Instance Normalization (AdaIN)



- from Huang *et al.*, ECCV 2018

Adaptive Instance Normalization (AdaIN)

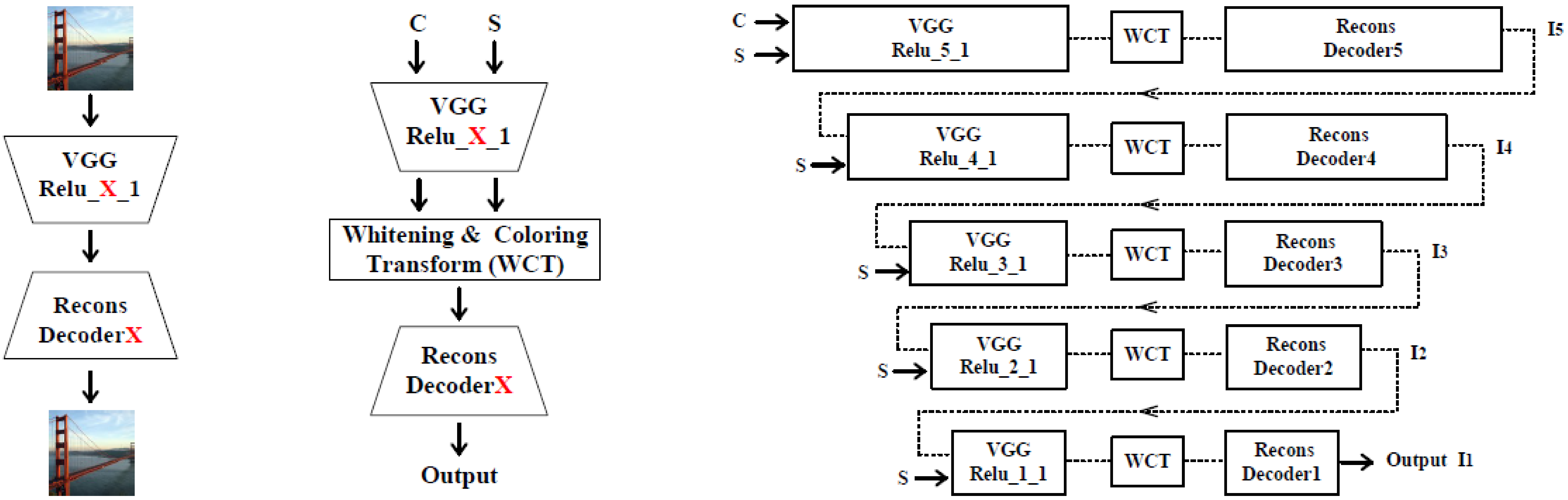
DEVIEW 2019

- from Kerras *et al.*, CVPR 2019

Whitening and Coloring Transforms (WCT)

DEVIEW 2019

"평균과 분산만 맞추지 말고 **공분산까지 맞춰서** 더 스타일을 잘 입혀보자"

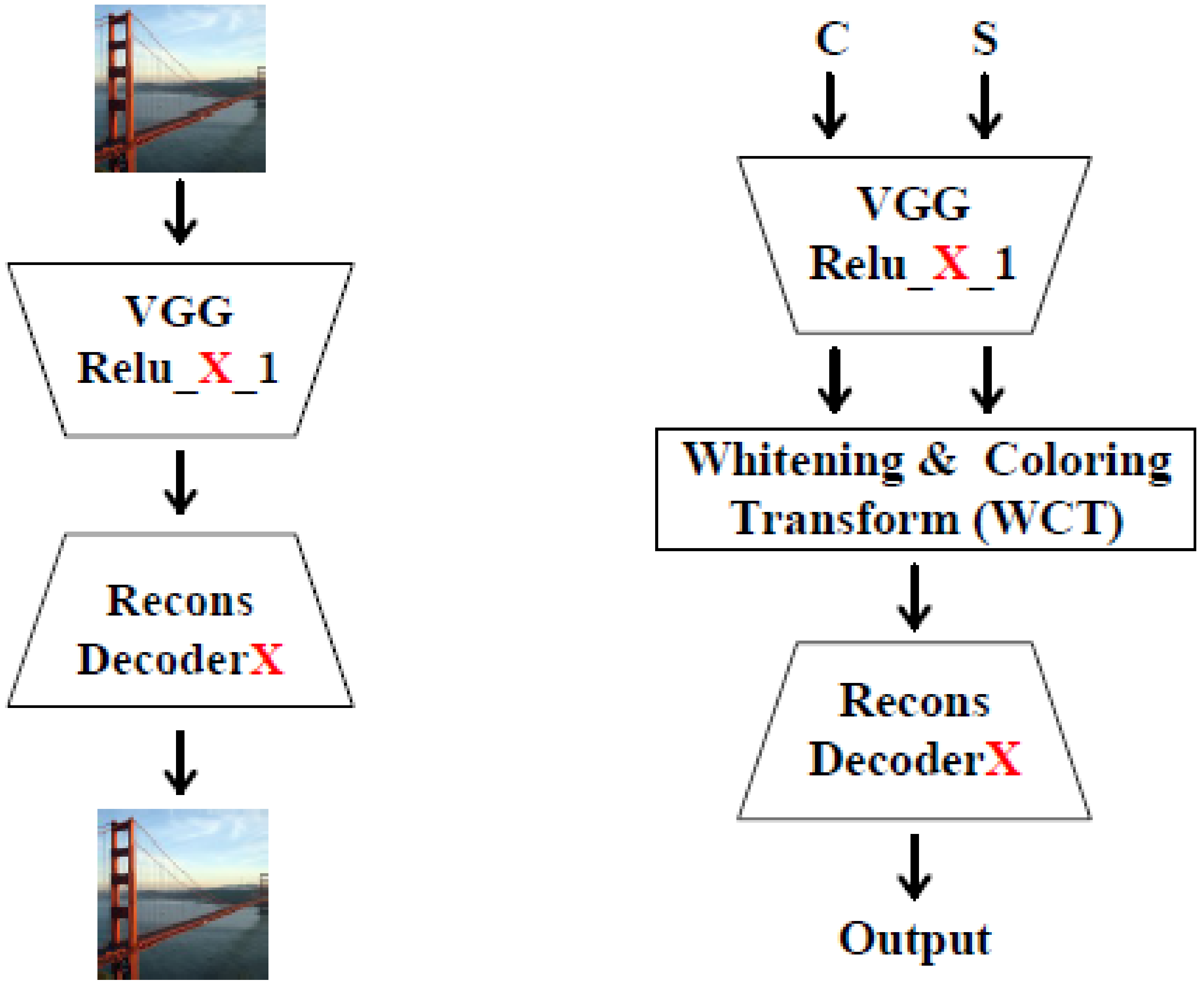


(a) Reconstruction (b) Single-level stylization (c) Multi-level stylization

- from Yi *et al.*, NIPS 2017

Whitening and Coloring Transforms (WCT)

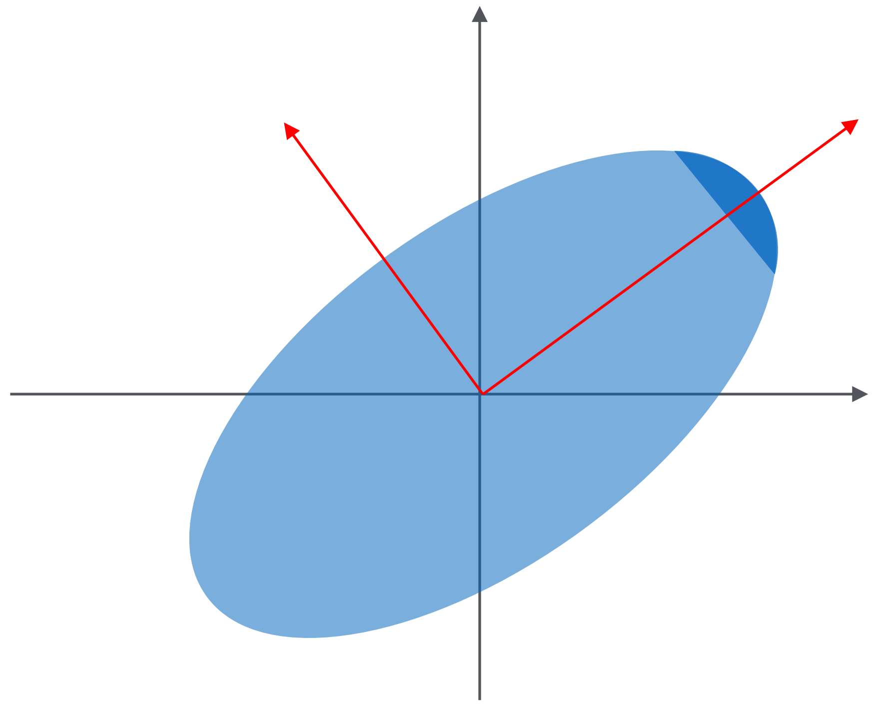
"평균과 분산만 맞추지 말고 **공분산까지 맞춰서** 더 스타일을 잘 입혀보자"



(a) Reconstruction (b) Single-level stylization

How?

*** ZCA: Zero-phase Component Analysis**

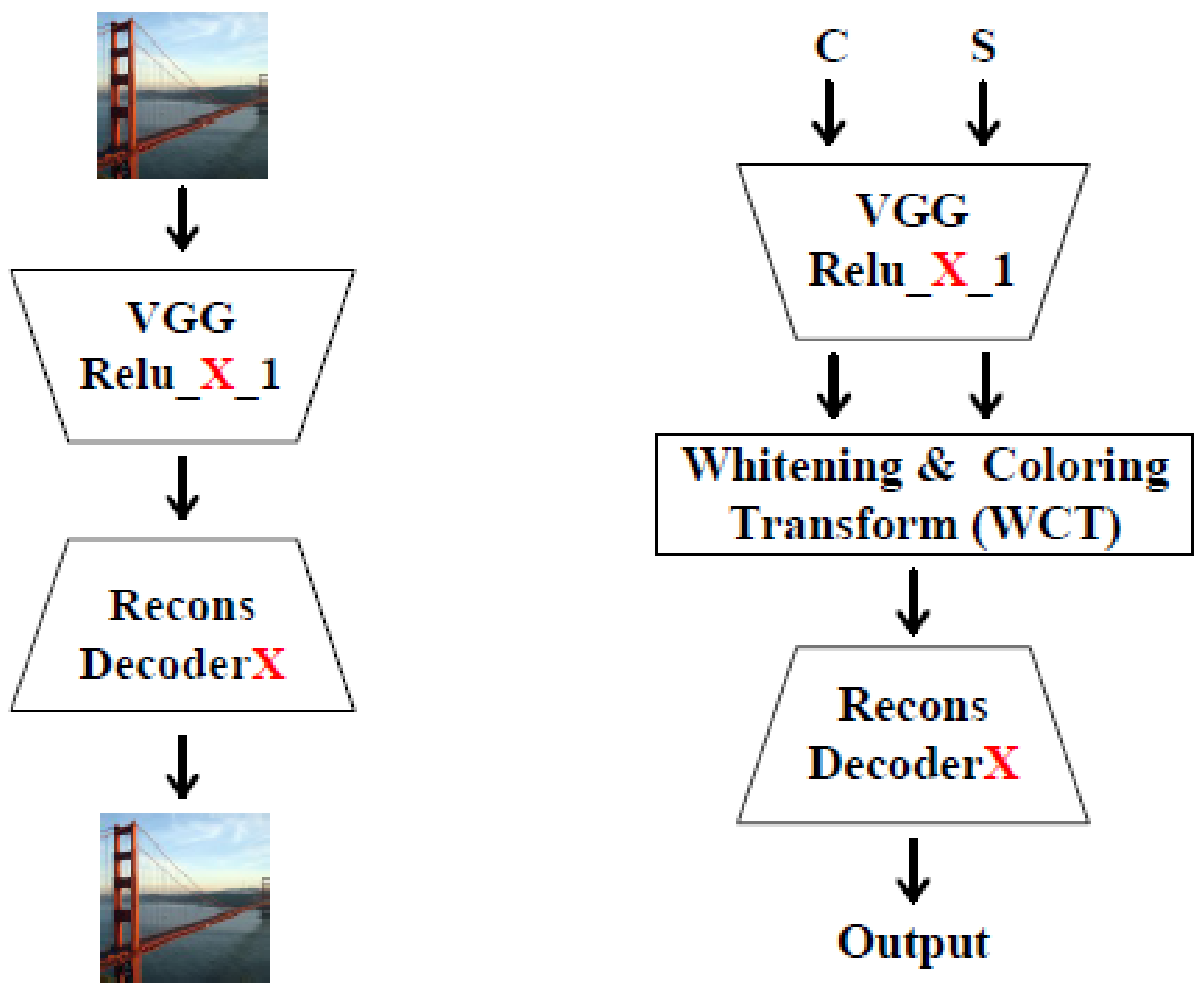


$\boldsymbol{f}$

where, $\boldsymbol{f}\boldsymbol{f}^H = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^H$ - from Yi *et al.*, NIPS 2017

Whitening and Coloring Transforms (WCT)

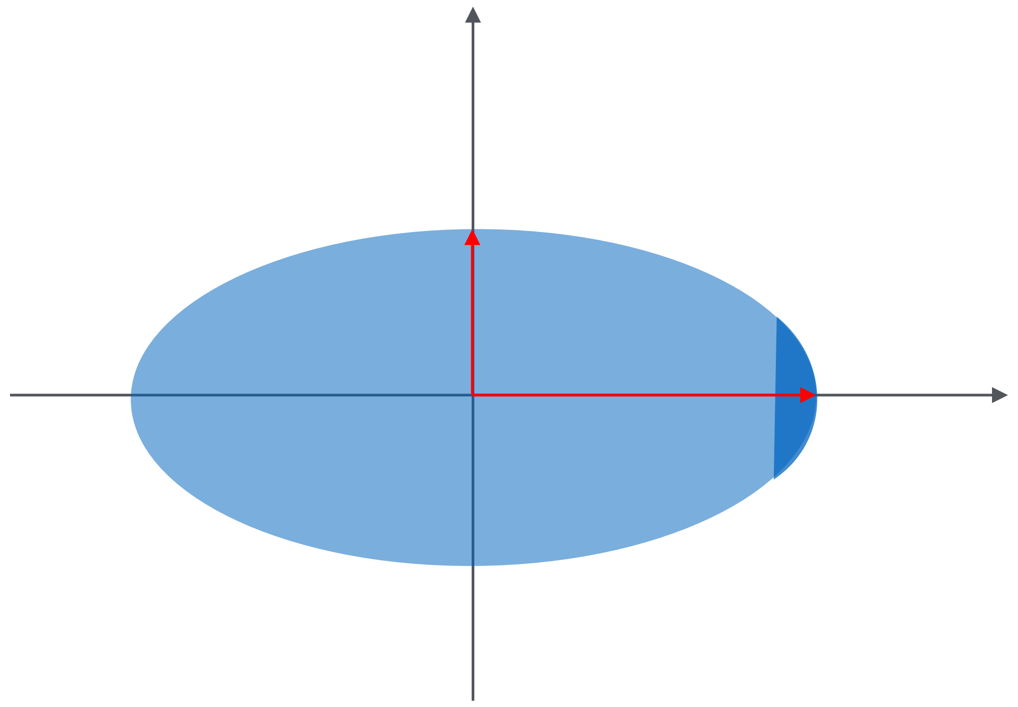
"평균과 분산만 맞추지 말고 공분산까지 맞춰서 더 스타일을 잘 입혀보자"



(a) Reconstruction (b) Single-level stylization

How?

* **ZCA: Zero-phase Component Analysis**

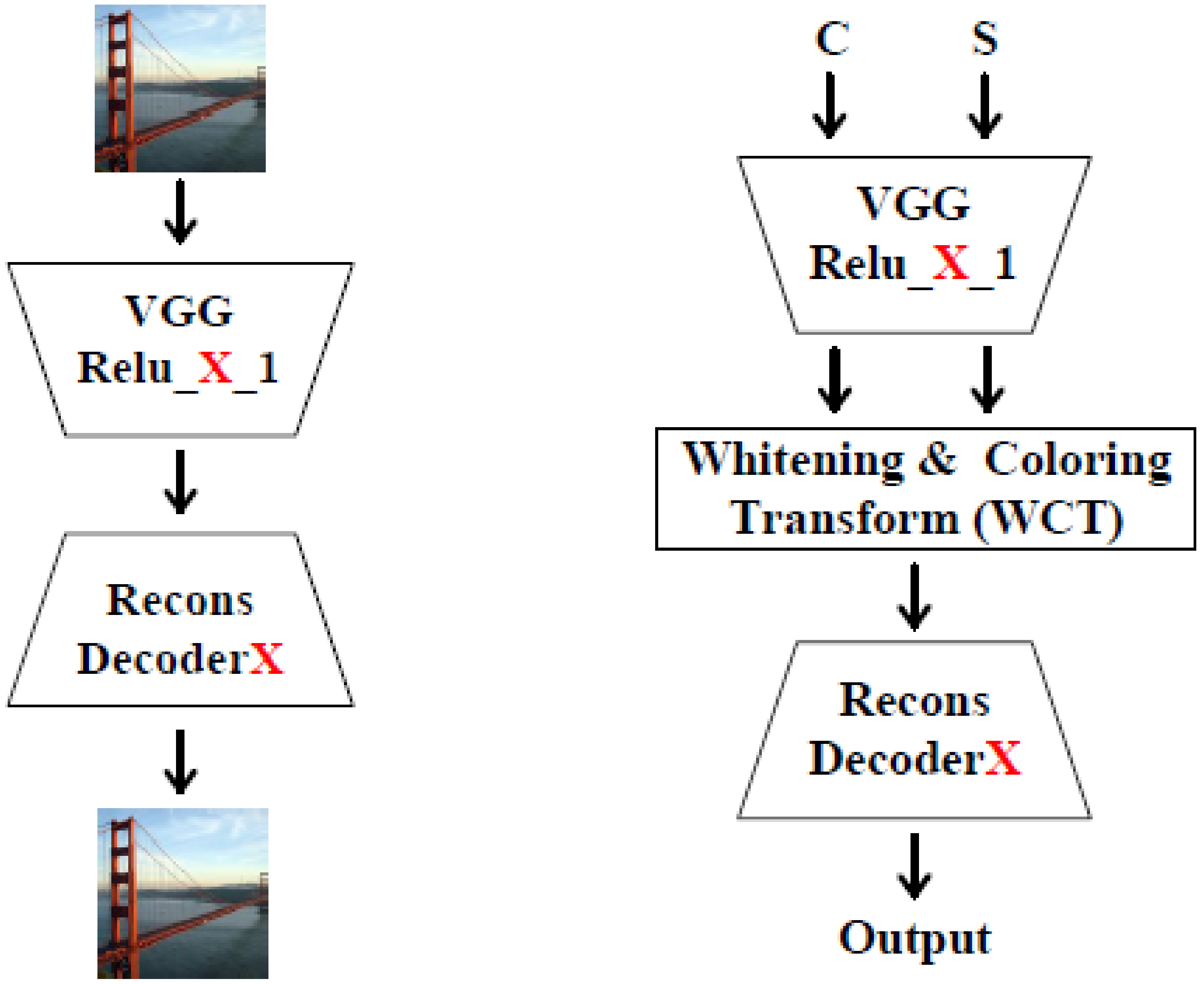


$$E^H f$$

where, $ff^H = E\Lambda E^H$ - from Yi *et al.,* NIPS 2017

Whitening and Coloring Transforms (WCT)

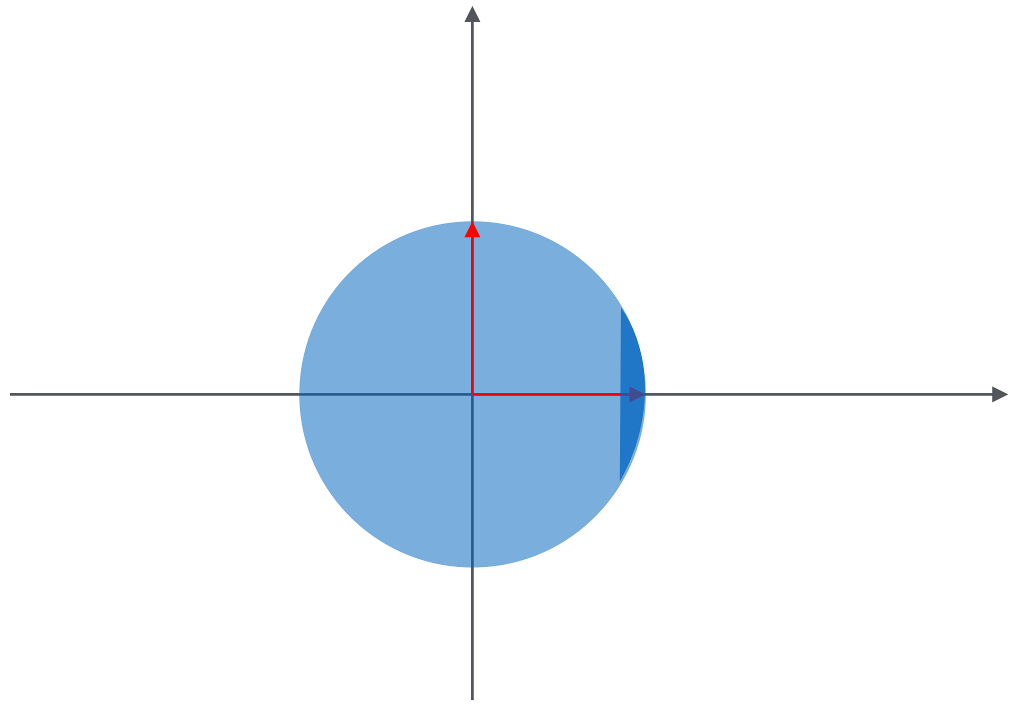
"평균과 분산만 맞추지 말고 공분산까지 맞춰서 더 스타일을 잘 입혀보자"



(a) Reconstruction (b) Single-level stylization

How?

*** ZCA: Zero-phase Component Analysis**

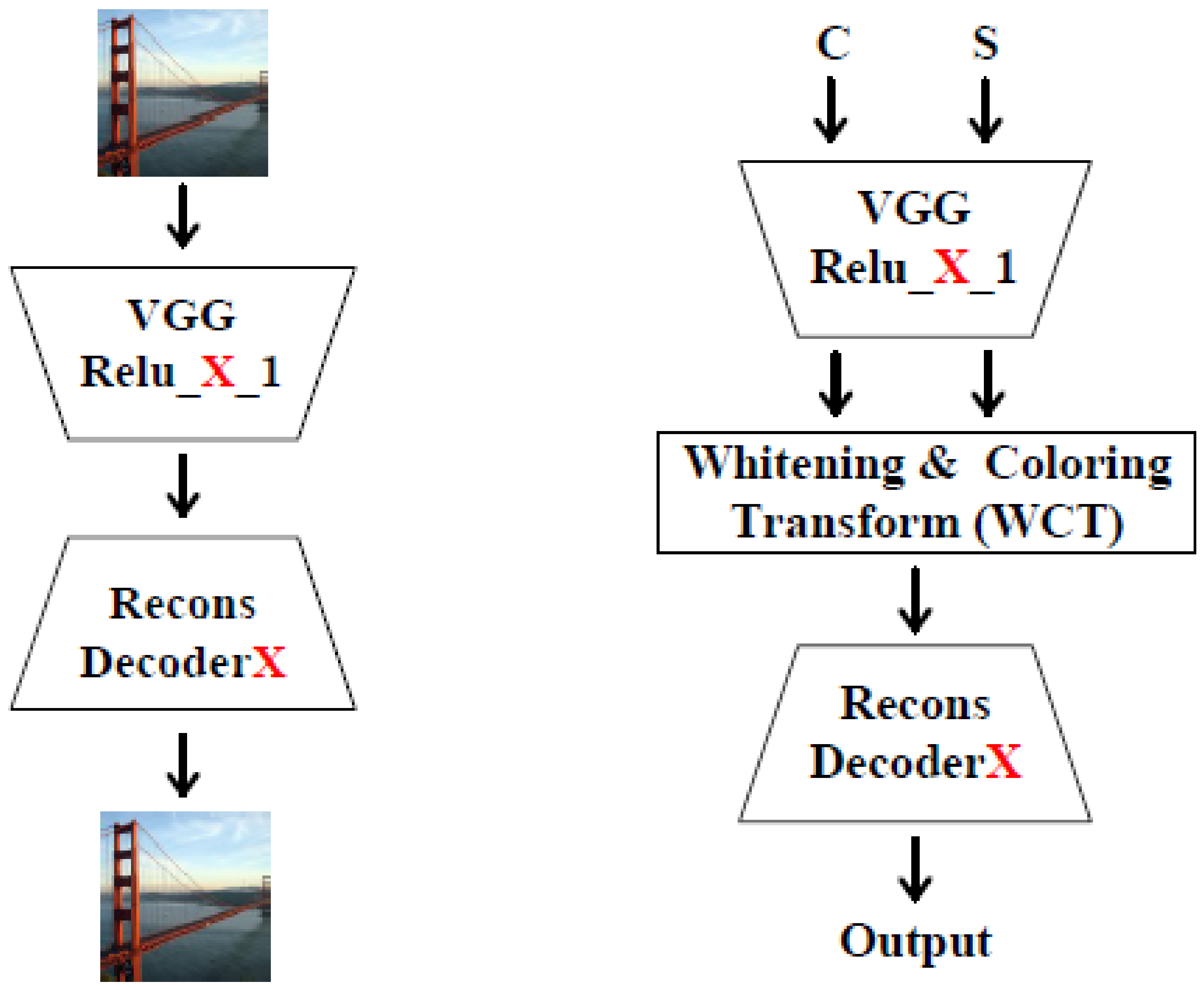


$$\Lambda^{-\frac{1}{2}} E^H f$$

where, $ff^H = E\Lambda E^H$ - from Yi *et al.,* NIPS 2017

Whitening and Coloring Transforms (WCT)

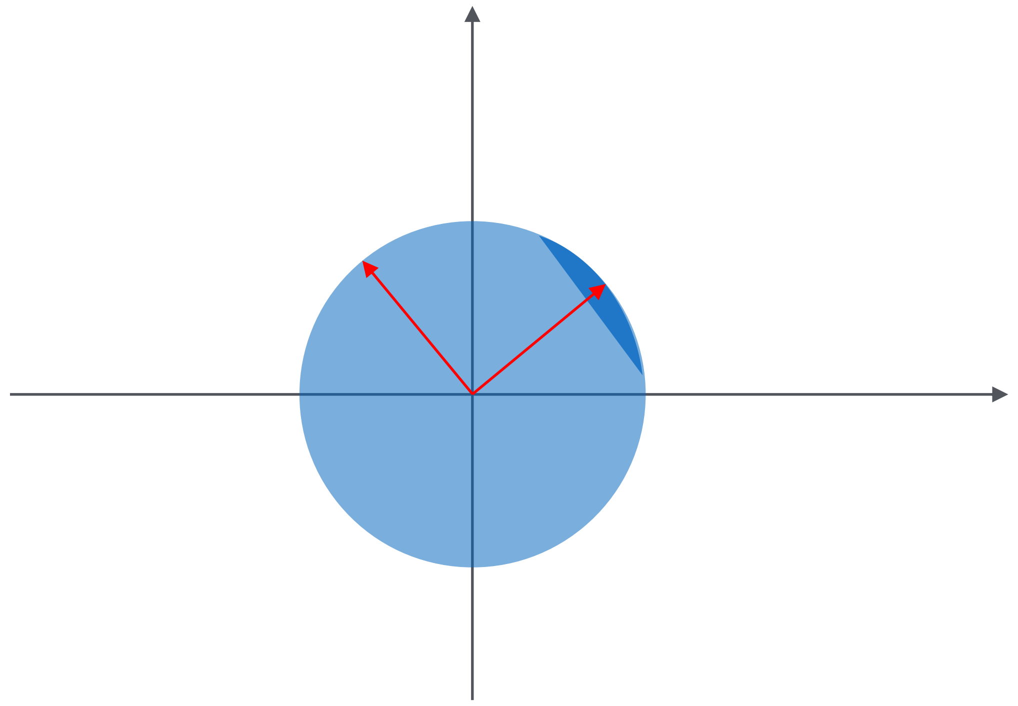
"평균과 분산만 맞추지 말고 **공분산까지 맞춰서** 더 스타일을 잘 입혀보자"



(a) Reconstruction (b) Single-level stylization

How?

* ZCA: Zero-phase Component Analysis



$$E\Lambda^{-\frac{1}{2}}E^{H}f$$

where, $ff^{H} = E\Lambda E^{H}$

\- from Yi *et al.*, NIPS 2017

Whitening and Coloring Transforms (WCT)

DEVIEW 2019

Whitening

$$\hat{f}_c = E_c \, D_c^{-\frac{1}{2}} \, E_c^{\top} \, f_c$$

f_c centered content feature

D_c diagonal matrix with the eigenvalues of the covariance matrix

E_c orthogonal matrix of eigenvectors

Coloring

$$\hat{f}_{cs} = E_s \, D_s^{\frac{1}{2}} \, E_s^{\top} \, \hat{f}_c$$

f_s centered style feature

D_s diagonal matrix with the eigenvalues of the covariance matrix

E_s orthogonal matrix of eigenvectors

Whitening and Coloring Transforms (WCT)

"하지만 한 번에는 잘 안 먹히니까 recursive하게 여러 번 스타일을 바꿔주는데…"

Multi-level Stylization

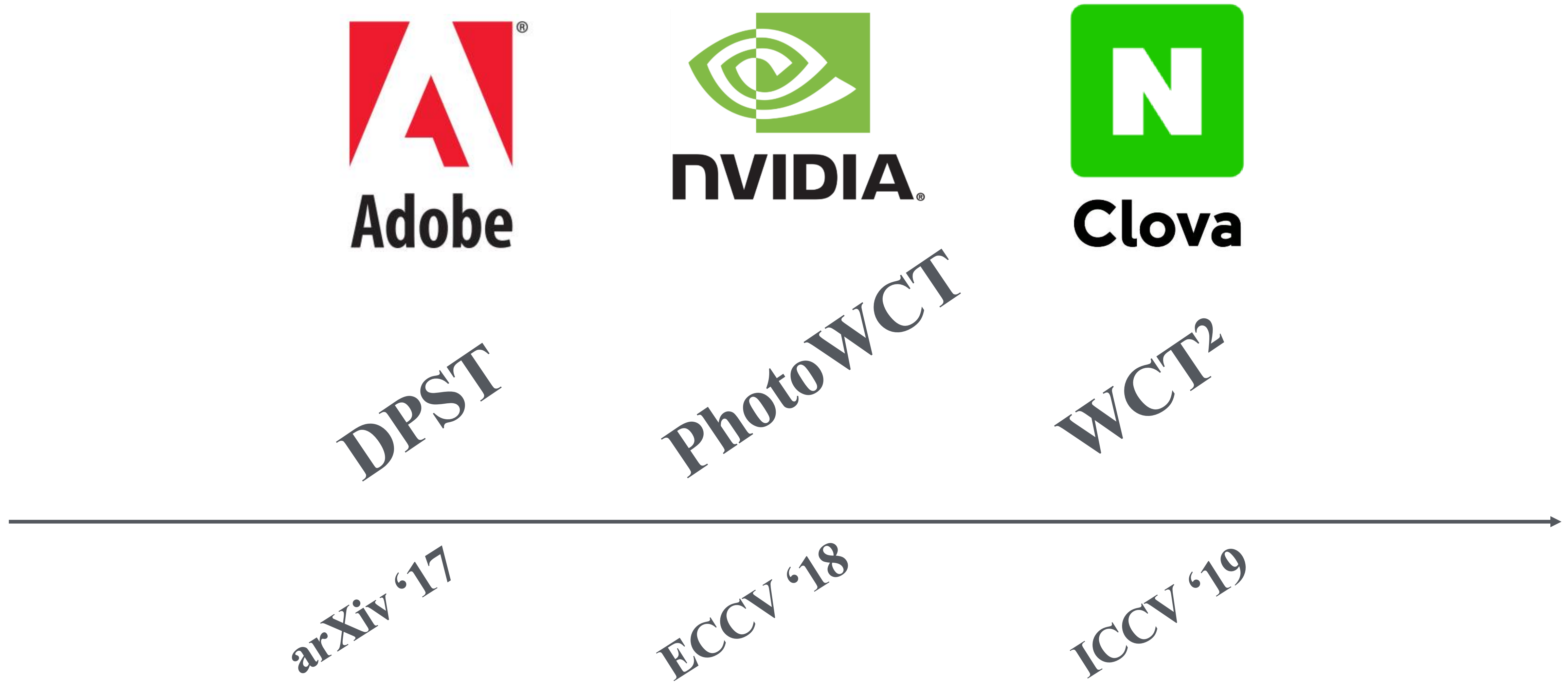
- 느리고
- Error는 증폭되고
- 모델 크기는 매우 크다.

Contents

Single level

Multi level

여러 모로 artistic style transfer 계열은 우리 목적인 photorealistic 결과를 얻는 것에 적합하지 않다.

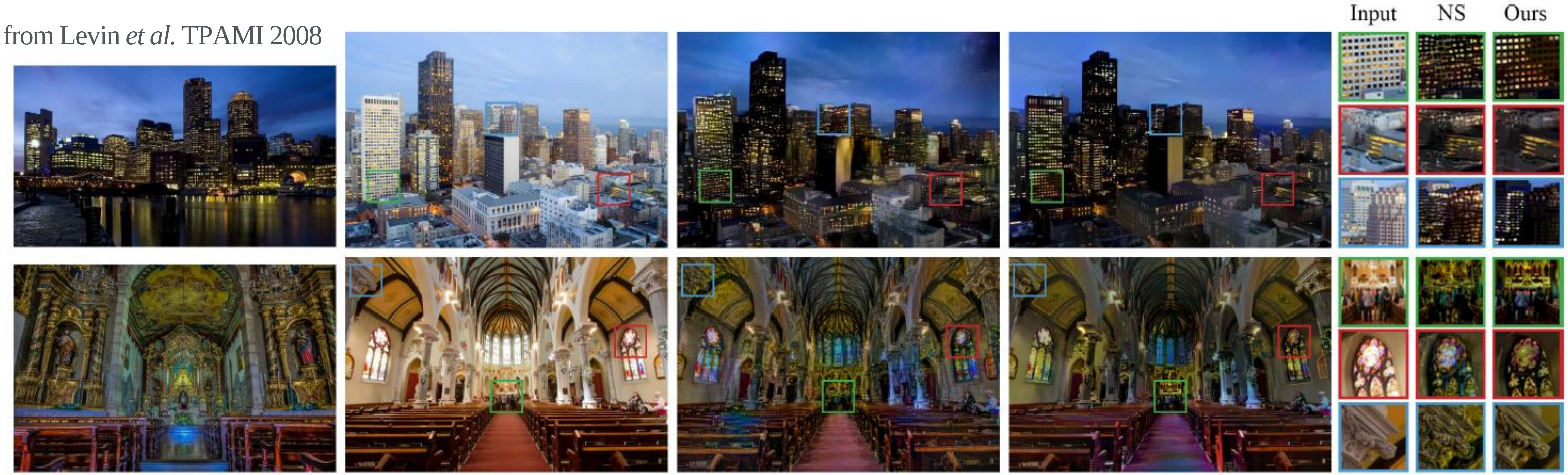


최신 연구 흐름 한눈에 살펴보기

1. Artistic Style Transfer
2. Photorealistic Style Transfer

Deep Photo Style Transfer

- from Levin *et al.* TPAMI 2008



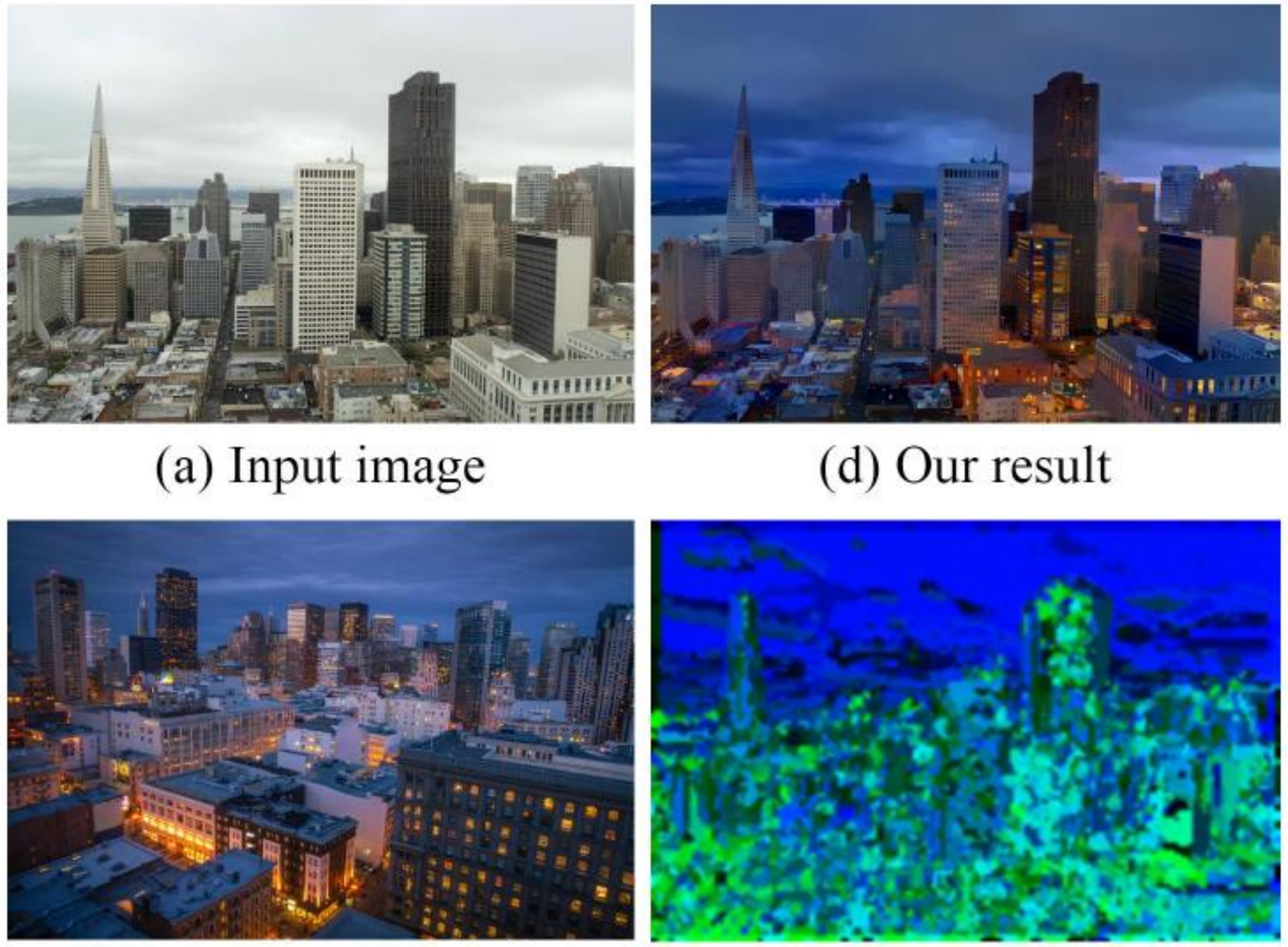
(a) Reference style image (b) Input image (c) Neural Style (distortions) (d) Our result (e) Insets

$$\mathcal{L}_{\text{total}} = \sum_{l=1}^{L} \alpha_\ell \mathcal{L}_c^\ell + \Gamma \sum_{\ell=1}^{L} \beta_\ell \mathcal{L}_{s+}^\ell + \lambda \mathcal{L}_m$$

given an input image I with N pixels, $\mathcal{M}_I$ is $N \times N$

$$\mathcal{L}_m = \sum_{c=1}^{3} V_c[O]^T \mathcal{M}_I V_c[O]$$

Deep Photo Style Transfer



(a) Input image (d) Our result

(e) Reference style image (h) Correspondence of (d) and (e)

"하늘은 하늘끼리 건물은 건물끼리
스타일이 맞도록 따로따로 관리하자"

$$\mathcal{L}_{\text{total}} = \sum_{l=1}^{L} \alpha_\ell \mathcal{L}_c^\ell + \boxed{\Gamma \sum_{\ell=1}^{L} \beta_\ell \mathcal{L}_{s+}^\ell} + \lambda \mathcal{L}_m$$

$$\mathcal{L}_{s+}^\ell = \sum_{c=1}^{C} \frac{1}{2N_{\ell,c}^2} \sum_{ij} (G_{\ell,c}[O] - G_{\ell,c}[S])_{ij}^2$$

$$F_{\ell,c}[O] = F_\ell[O] M_{\ell,c}[I] \quad F_{\ell,c}[S] = F_\ell[S] M_{\ell,c}[S]$$

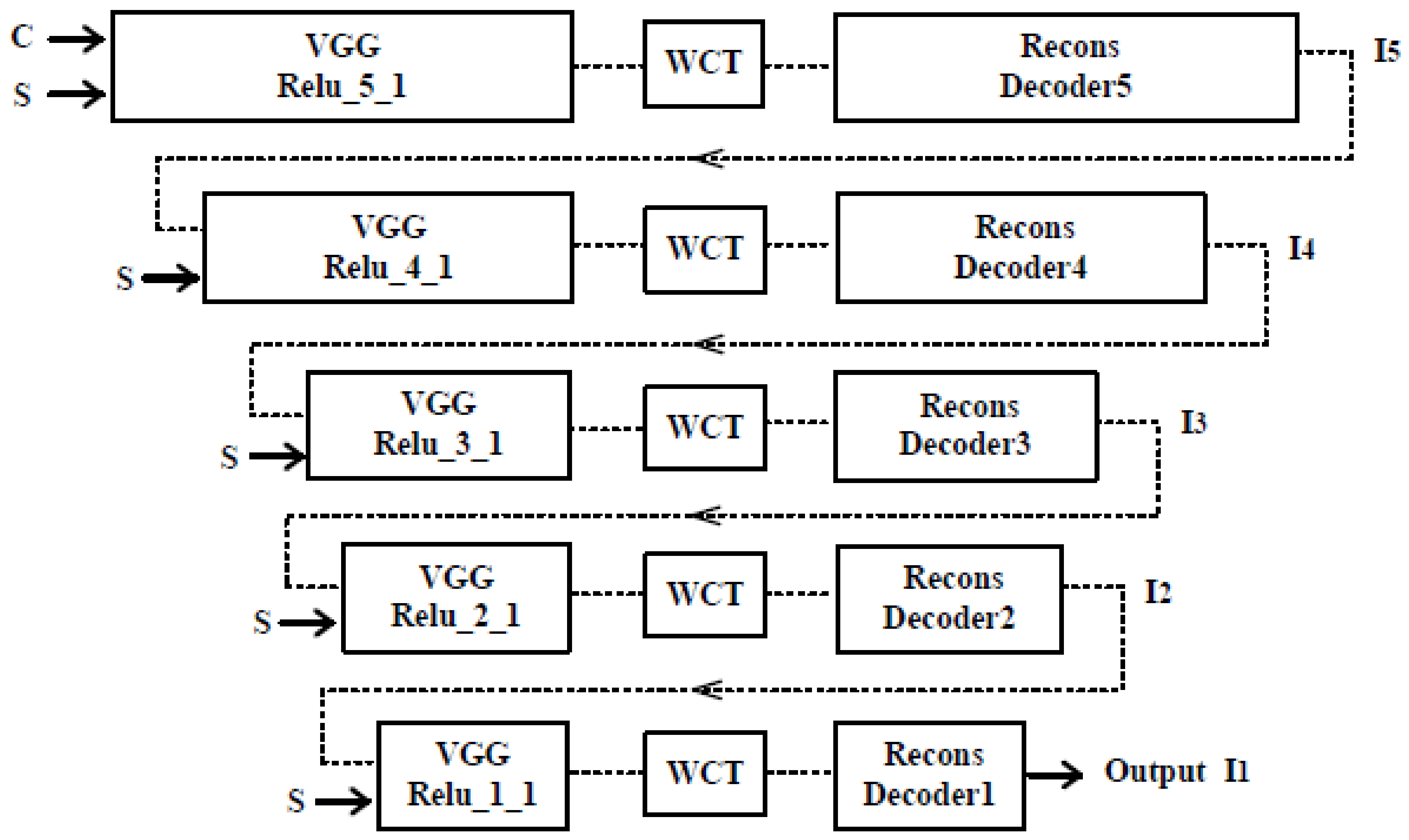
PhotoWCT

DEVIEW 2019

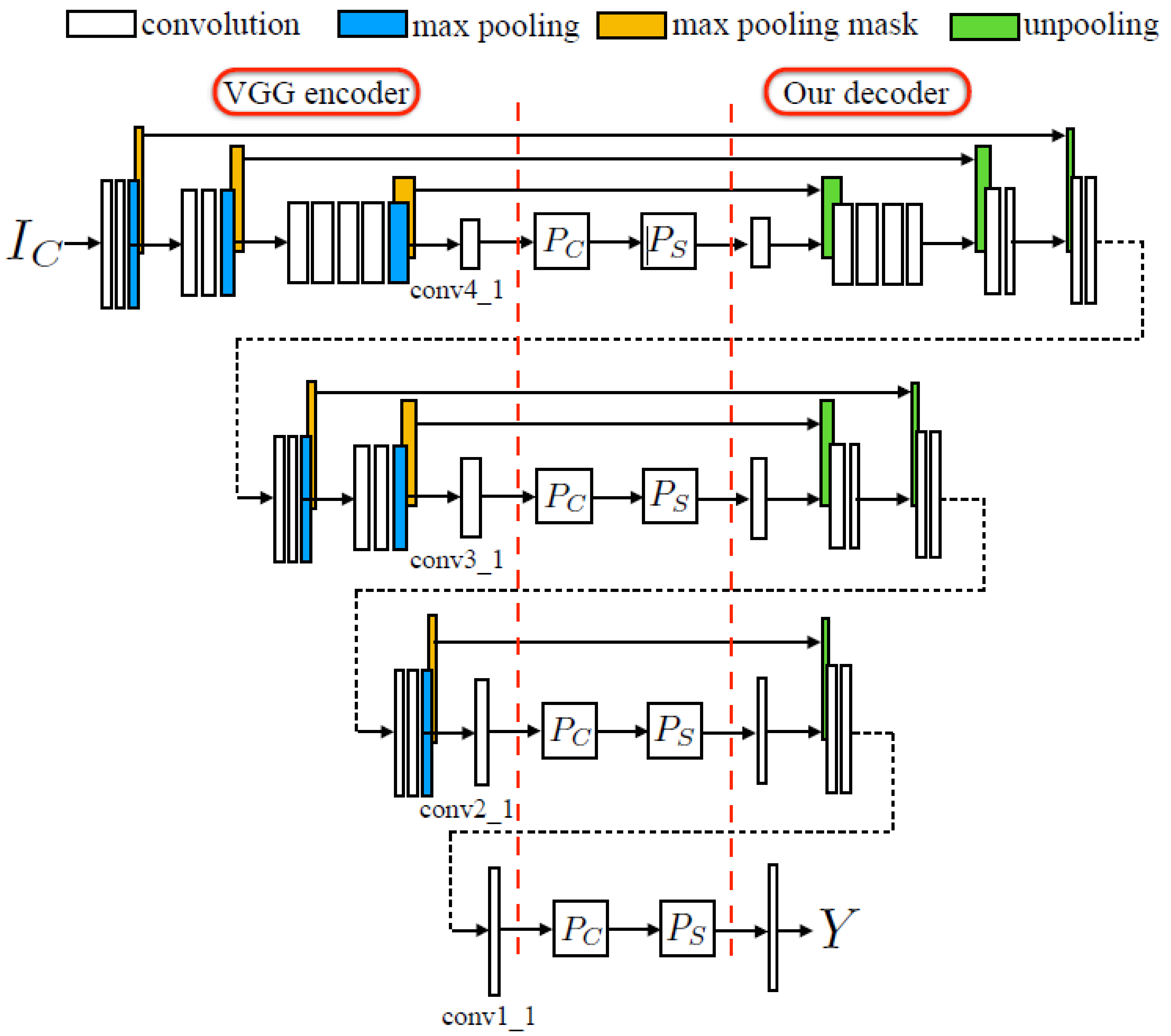
"Decoder에 최소한 max pooling 했던 위치라도 알려주자 (Unpooling) "

WCT (artistic model)

: "VGG decoder uses nearest-neighbor upsampling"

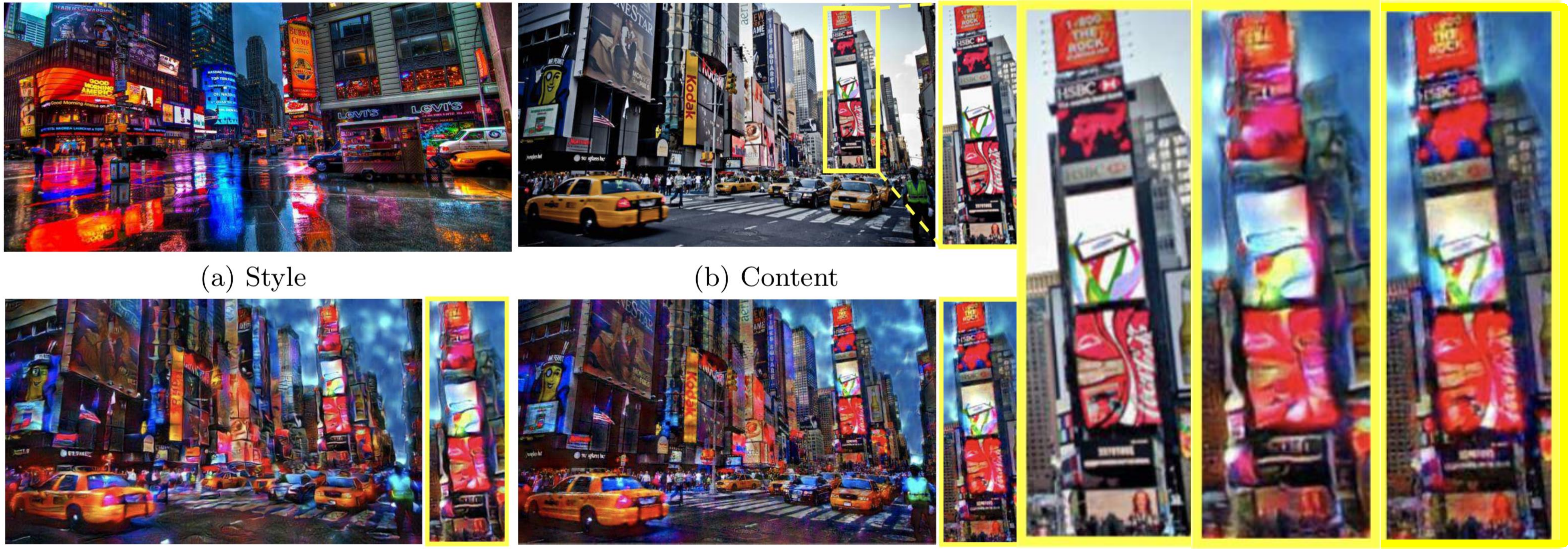


PhotoWCT (photorealistic model)



PhotoWCT

"Decoder에 max pool 위치를 알려주는 것(unpooling)은 생각보다 효과가 없다??"



(a) Style (b) Content

(c) WCT [10] (d) PhotoWCT

Content WCT PhotoWCT

PhotoWCT

"사실상 smoothing이라는 후처리 과정이 모든 일을 다하고 있었던 것"

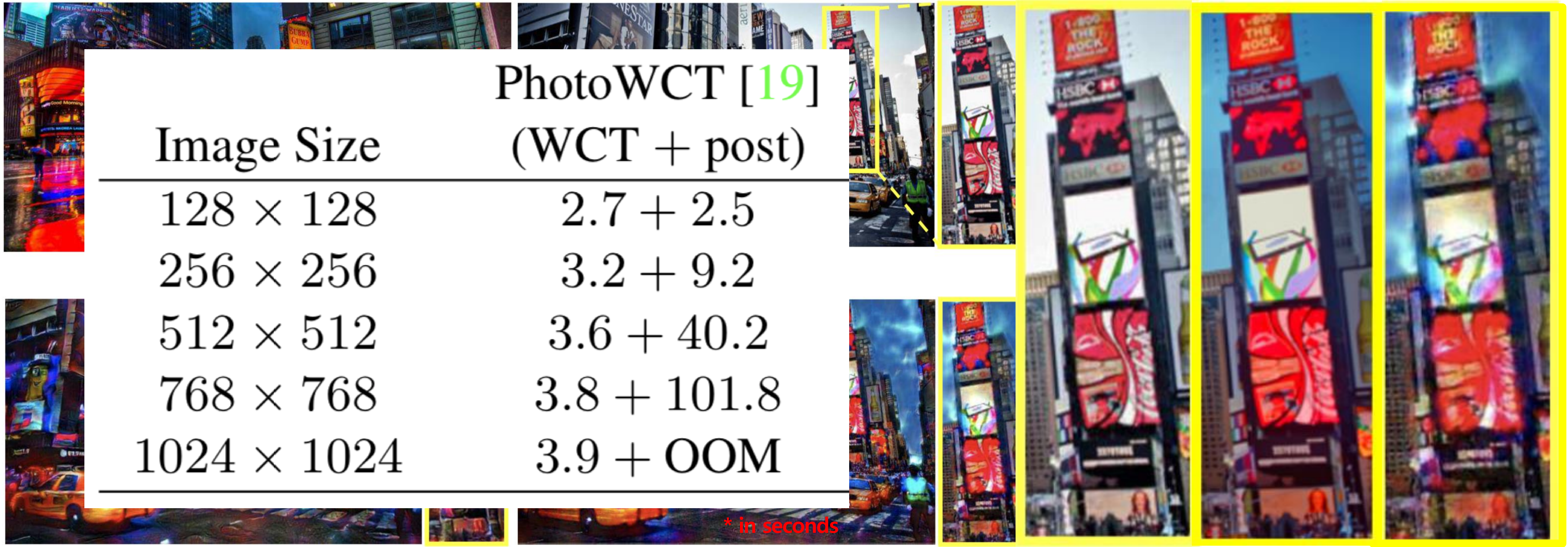


Image Size	PhotoWCT [19] (WCT + post)
128×128	$2.7 + 2.5$
256×256	$3.2 + 9.2$
512×512	$3.6 + 40.2$
768×768	$3.8 + 101.8$
1024×1024	$3.9 + \text{OOM}$

* in seconds

(c) WCT [10]

(d) PhotoWCT

Content

PhotoWCT + smoothing

PhotoWCT

PhotoWCT

"새로 개발한 방식은 **후처리 없이도** 기존 모델(+후처리)의 성능보다 나은 결과"

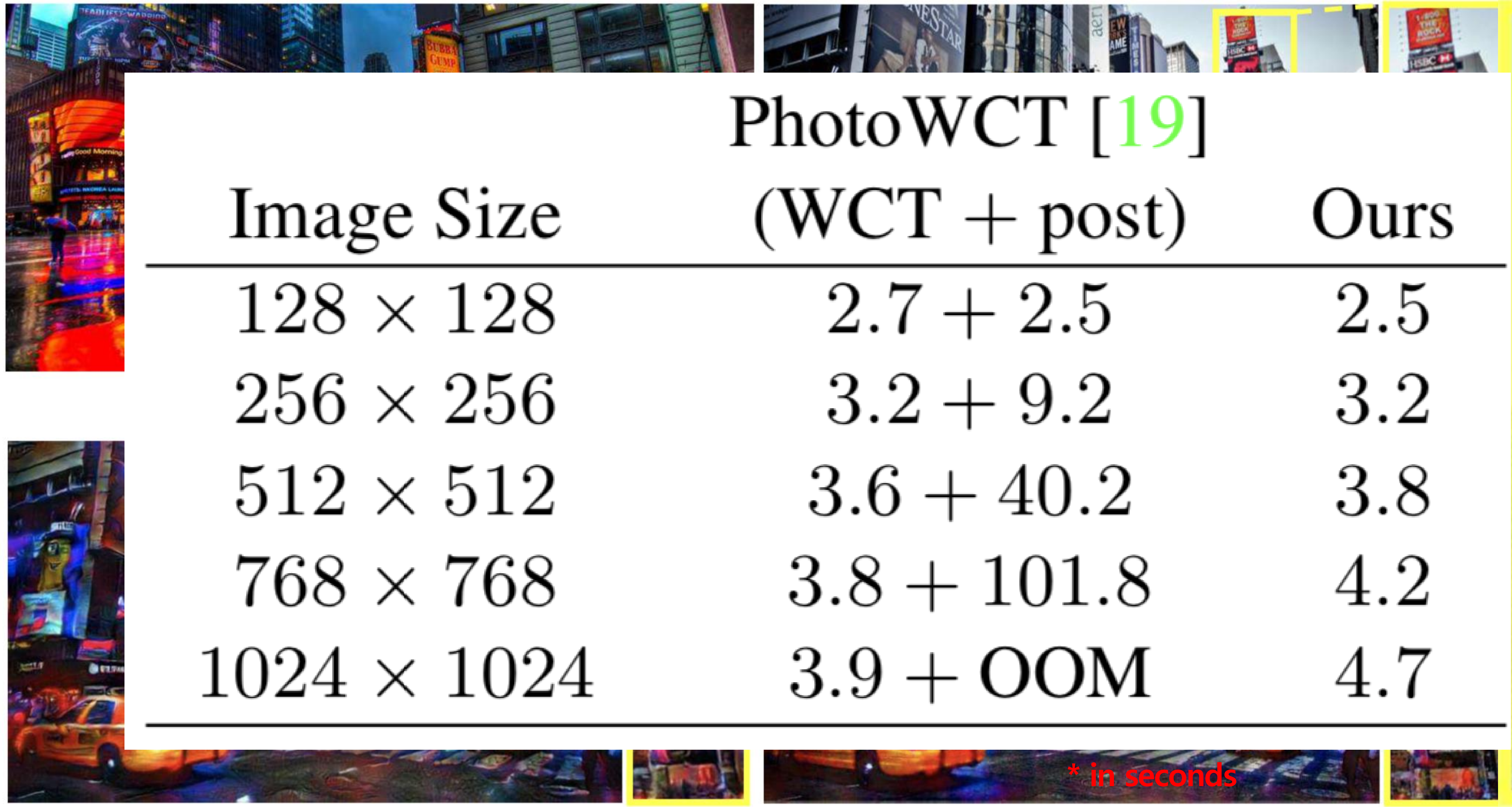


Image Size	PhotoWCT [19] (WCT + post)	Ours
128×128	$2.7 + 2.5$	2.5
256×256	$3.2 + 9.2$	3.2
512×512	$3.6 + 40.2$	3.8
768×768	$3.8 + 101.8$	4.2
1024×1024	$3.9 + \text{OOM}$	4.7

* in seconds

(c) WCT [10]

(d) PhotoWCT

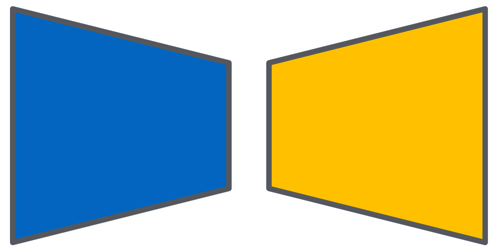
Content

PhotoWCT
+ smoothing

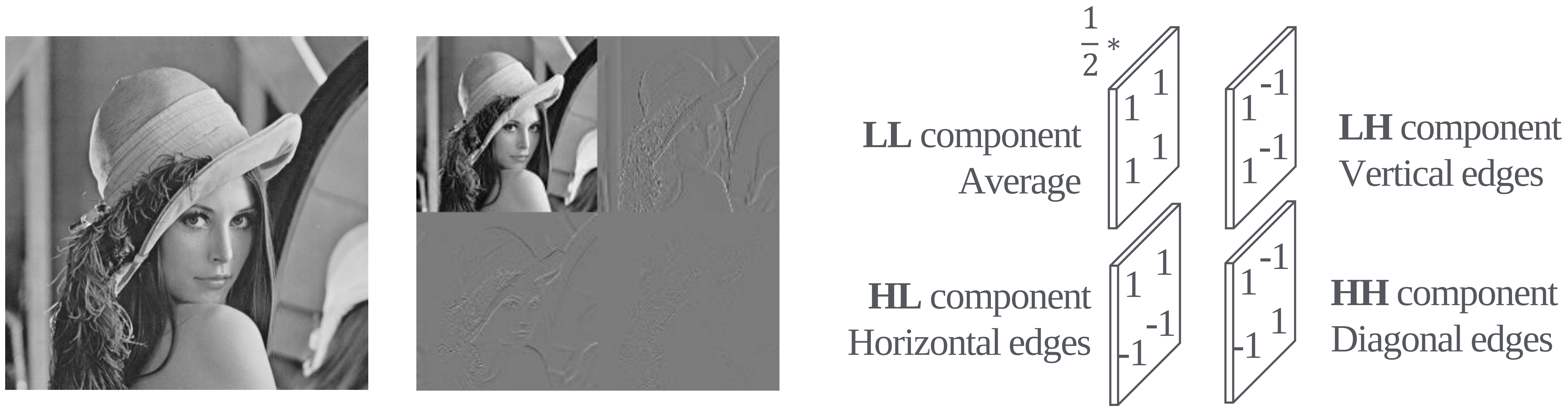
Ours

WCT via Wavelet Corrected Transforms (WCT2)

1. "Encoder-Decoder가 **좋은 성질**을 갖는 함수를 학습할 수 있도록 강제하는 구조"
   1) Pooling과 비슷한 역할을 하면서,
   2) Encode-decode 과정에서 정보를 잃지 않아야 하고,
   3) 입력 이미지의 특징을 충분히 잘 표현할 수 있는 모듈



$$\Phi\Phi^T = \sum_{k=1}^{L} T_k T_k^T = I.$$

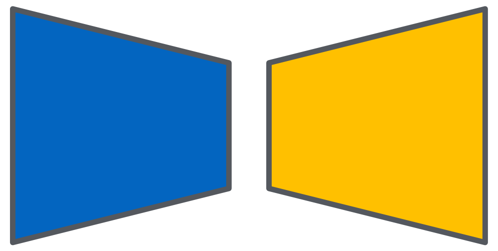


* Lena image decomposition using Haar wavelet transform

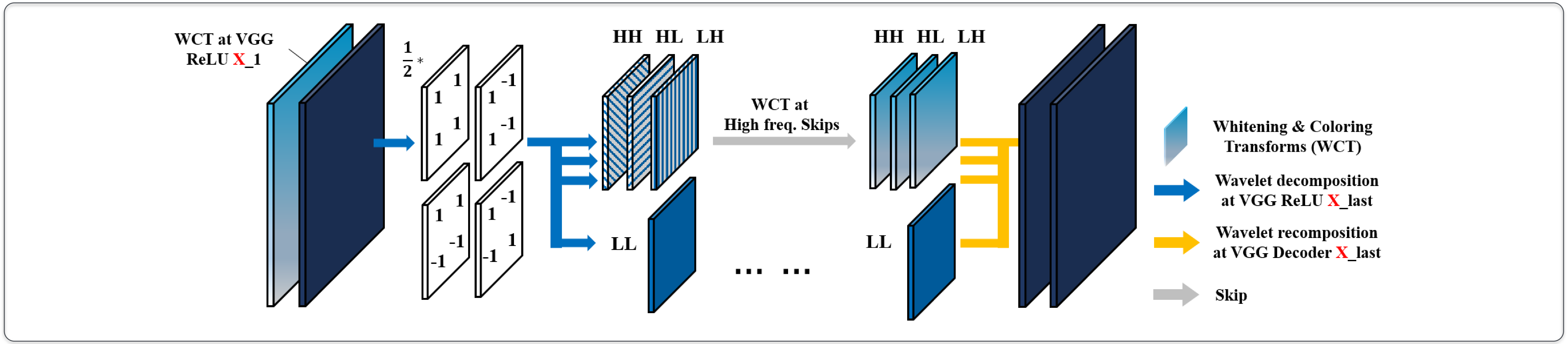
WCT via Wavelet Corrected Transforms (WCT2)

1. “Encoder-Decoder가 좋은 성질을 갖는 함수를 학습할 수 있도록 강제하는 구조”

   1) Pooling과 비슷한 역할을 하면서,
   2) Encode-decode 과정에서 정보를 잃지 않아야 하고,
   3) 입력 이미지의 특징을 충분히 잘 표현할 수 있는 모듈



$$\Phi\Phi^T = \sum_{k=1}^{L} T_k T_k^T = I.$$

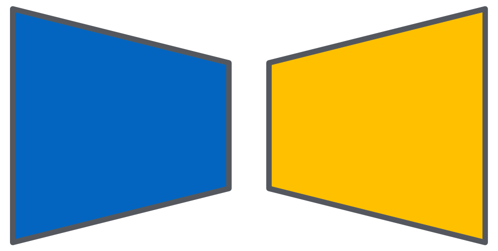


WCT (wavelet corrected transfer) 모듈

WCT via Wavelet Corrected Transforms (WCT2)

1. "Encoder-Decoder가 좋은 성질을 갖는 함수를 학습할 수 있도록 강제하는 구조"

   1) Pooling과 비슷한 역할을 하면서,
   2) Encode-decode 과정에서 정보를 잃지 않아야 하고,
   3) 입력 이미지의 특징을 충분히 잘 표현할 수 있는 모듈



$$\Phi\Phi^T = \sum_{k=1}^{L} T_k T_k^T = I.$$

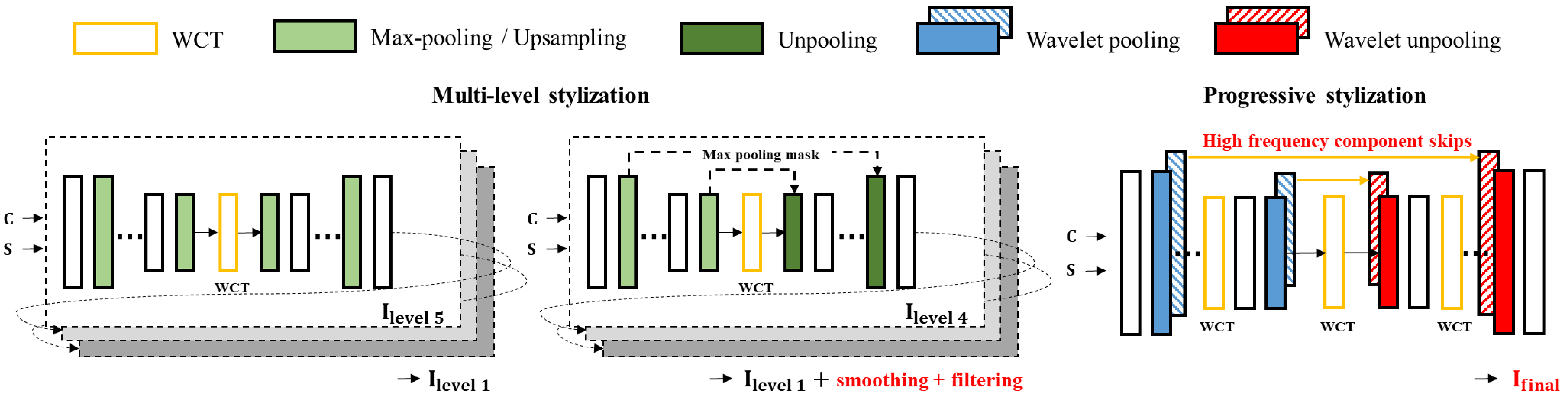
(a) Input (b) PhotoWCT [19] (c) Ours (d) Ours (LL only)

Low frequency만 스타일을 입히면 실제로 그렇게 동작한다!
(interpretable !)

WCT via Wavelet Corrected Transforms (WCT2)

DEVIEW 2019

2. "Multi-level 대신 Progressive stylization으로 한 번의 feed-forward만 수행"



(a) WCT (artistic) (b) PhotoWCT (c) Ours (WCT^2)

하나의 모델만 사용하기 때문에

1) error가 전파되는 것을 막아 깔끔하면서도 2) 더 가볍고 빠른 스타일 변환이 가능

WCT via Wavelet Corrected Transforms (WCT2)

DEVIEW 2019

Cherry pick **없는** 결과 이미지들

(a) Input (b) DPST [22] (c) PhotoWCT [19] (d) PhotoWCT (full) [19] (e) Ours (WCT^2)

WCT via Wavelet Corrected Transforms (WCT2)

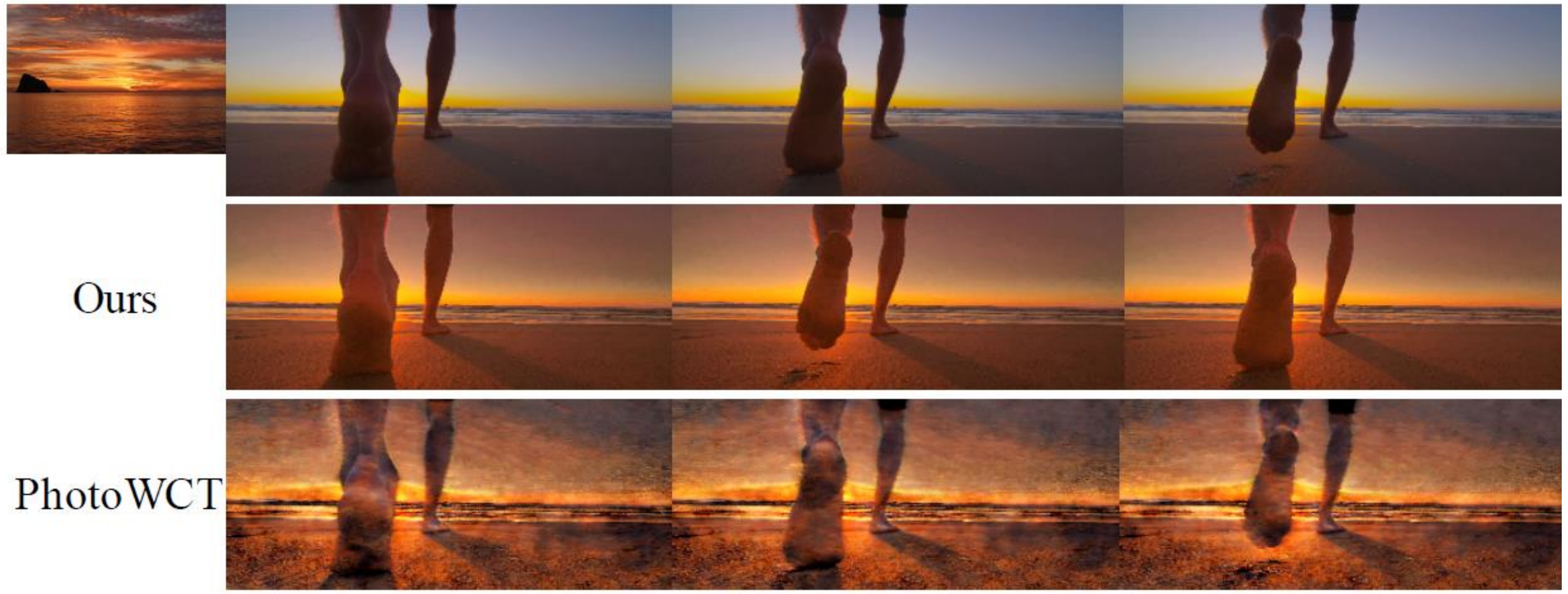
Cherry pick 없는 결과 이미지들

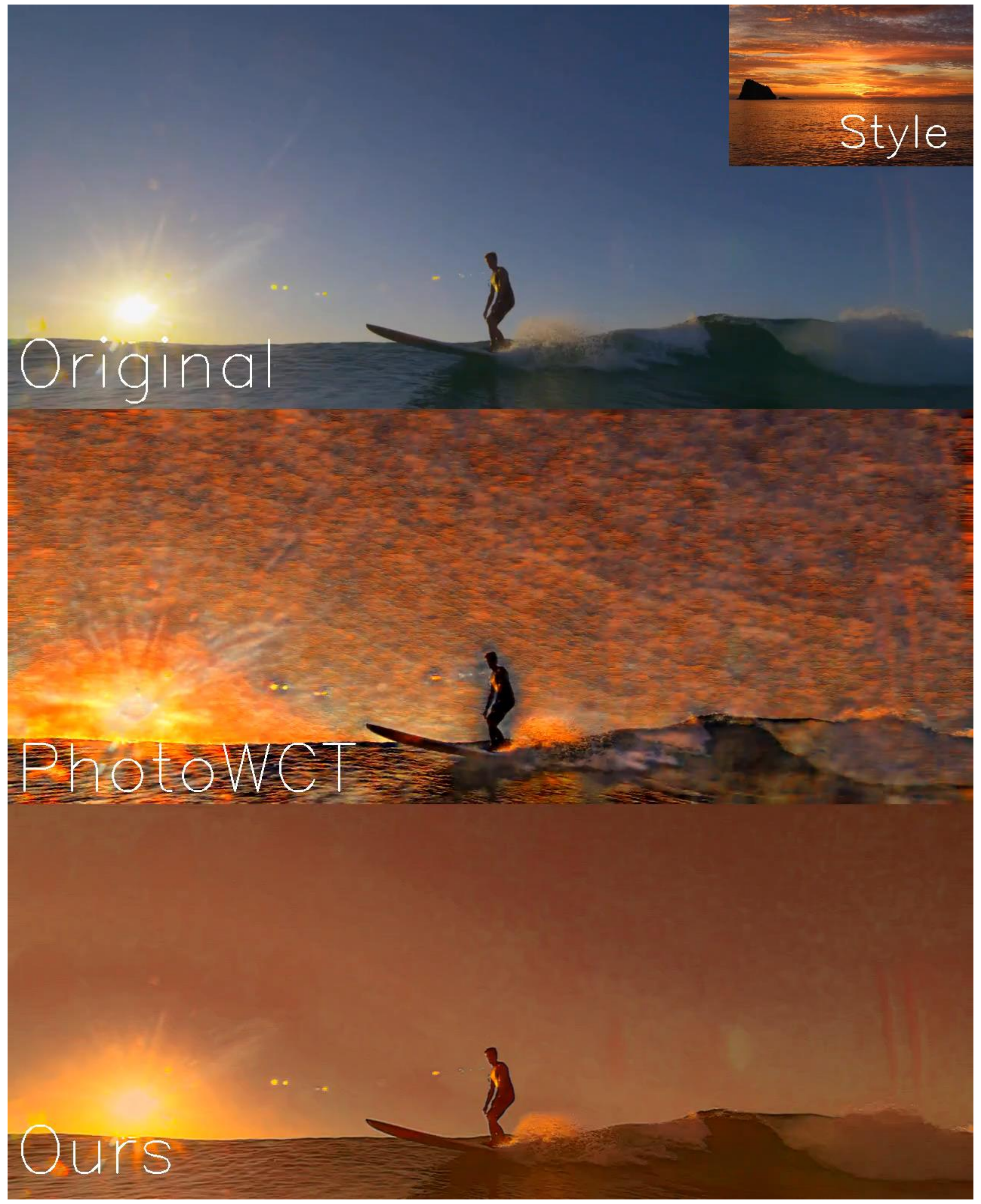
(a) Input (b) DPST [22] (c) PhotoWCT [19] (d) PhotoWCT (full) [19] (e) Ours (WCT^2)

WCT via Wavelet Corrected Transforms (WCT2)

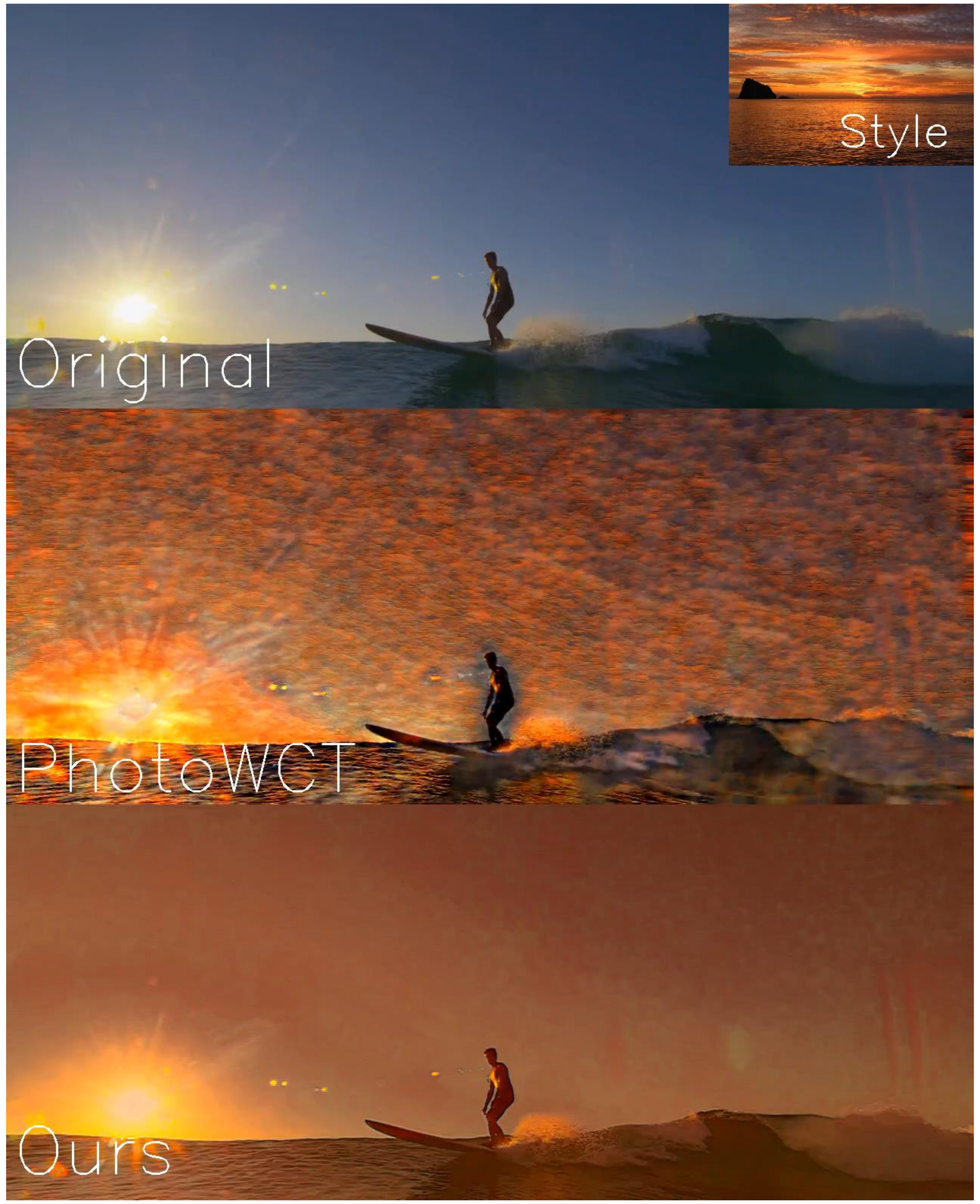
* 비디오 스타일 변환도 잘 된다!

"Wavelet의 안정적인 성질 덕분에 프레임 간의 시간적인 연속성을 고려한 보정이 없이도 깔끔한 결과를 볼 수 있다."





Photorealistic video stylization results (day-to-sunset).



THANK YOU ☺

jaejun.yoo@navercorp.com

Code, generated images, and pre-trained models are all available at github.com/clovaai/WCT2

WCT via Wavelet Corrected Transforms (WCT2)

DEVIEW 2019

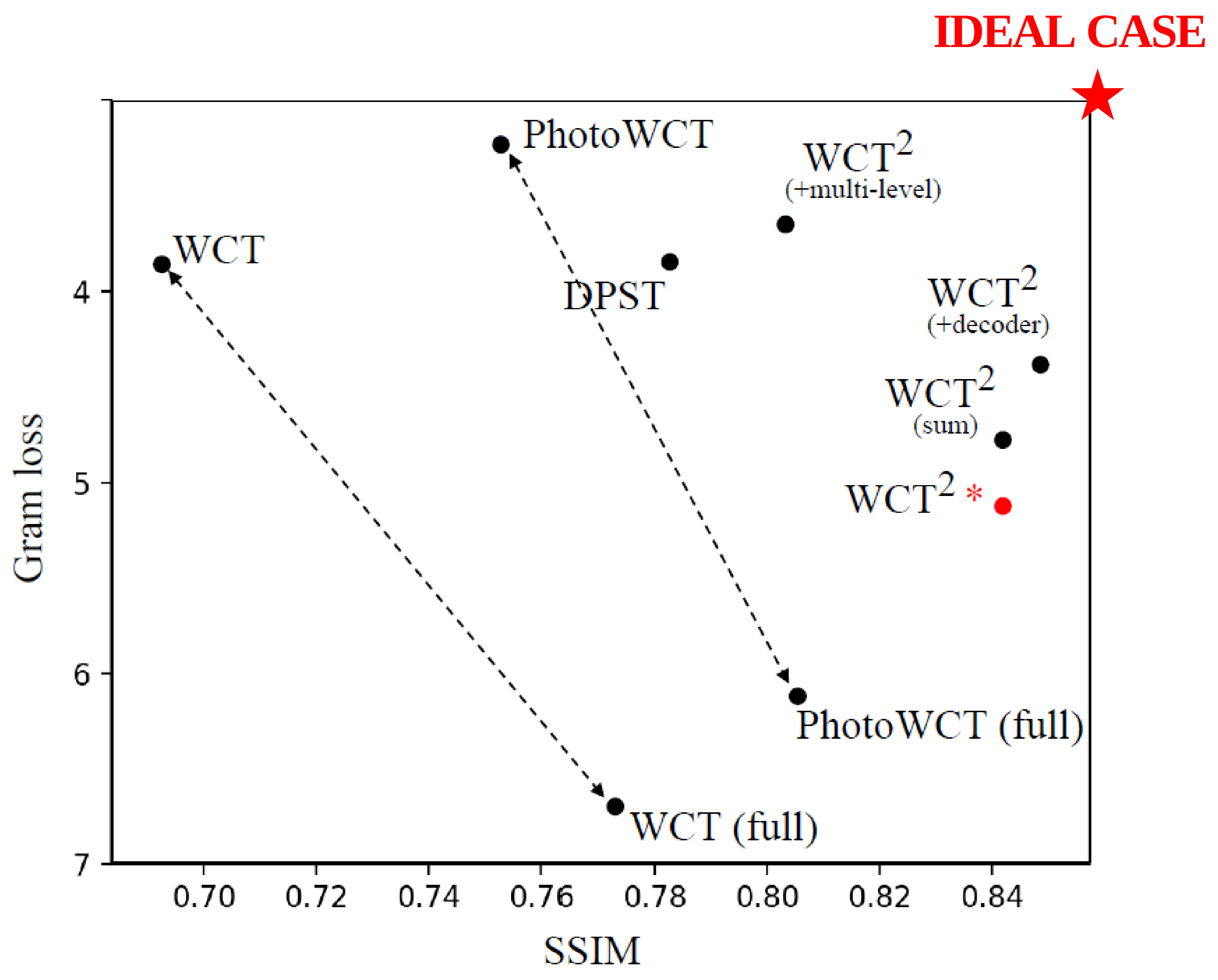
* User study results

	DPSP	PhotoWCT (full)	Ours
Fewest artifacts	21.34%	9.33%	**69.33%**
Best stylization	30.49%	12.74%	**56.77%**
Most preferred	24.63%	11.16%	**62.21%**

* Computational cost (seconds)

Image Size	DPST [22]	PhotoWCT [19] (WCT + post)	Ours
128×128	135.2	2.7 + 2.5	2.5
256×256	306.9	3.2 + 9.2	3.2
512×512	1020.7	3.6 + 40.2	3.8
768×768	2264.0	3.8 + 101.8	4.2
1024×1024	3887.8	3.9 + OOM	4.7

* SSIM index vs. Style loss

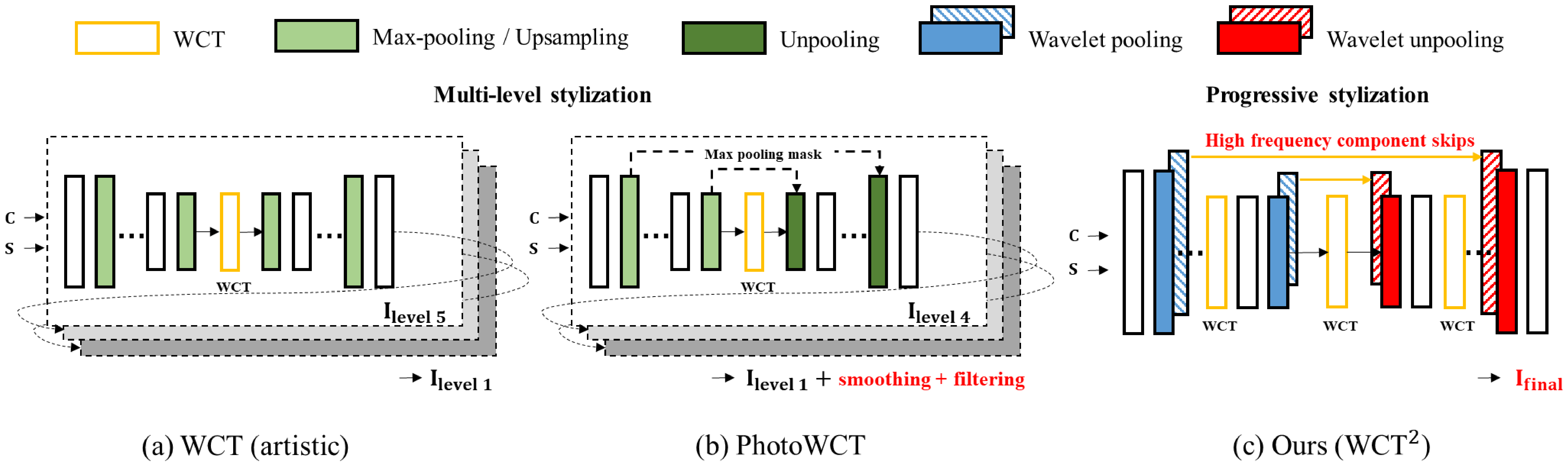


Content 이미지와 스타일 변환된 이미지 간의 structure similarity를 보는 SSIM은 값이 클 수록,

Style 이미지와 스타일 변환된 이미지 간의 스타일 정도의 차이를 보는 Gram loss는 작을 수록 좋다.

Summary

Wavelet pooing과 progressive 스타일 트랜스퍼를 사용하면서,



1. Better 스타일 변환 (less error propagation)
2. Faster model (기존 대비 840배 가속, no post-processing)
3. Lighter model (기존 대비 51% memory만 사용)
4. Stronger model (1k 고해상도 이미지 ~ 4 sec)