Photorealistic Style Transfer

Team 10

간주혁 강병규 김범준 이승한

Index

What is Style Transfer

Two images

- One image served as a content image, c
- Another image served as a style image, s

Change style of c to style of s

- having original c structure
- Just change style
  - Defining what is “style” is a hard problem
  - Here, style means just texture, color and so on

What is Style Transfer

[Content Image, c]

[Style Image, s]

What is Style Transfer

[Content Image, c]

[Style Image, s]

[Transferred Image]

What is Style Transfer

[Transferred Image]

- Here Transferred Image maintains original structure of the content image
- However the style changes to “Starry Night”

How To Transfer Style

Parametric Methods

- Optimization based
- Feed-forward based

Non-Parametric Methods

- Markov Random Field based

How To Transfer Style

Parametric Methods

- Optimization based
- Feed-forward based

Non-Parametric Methods

- Markov Random Field based

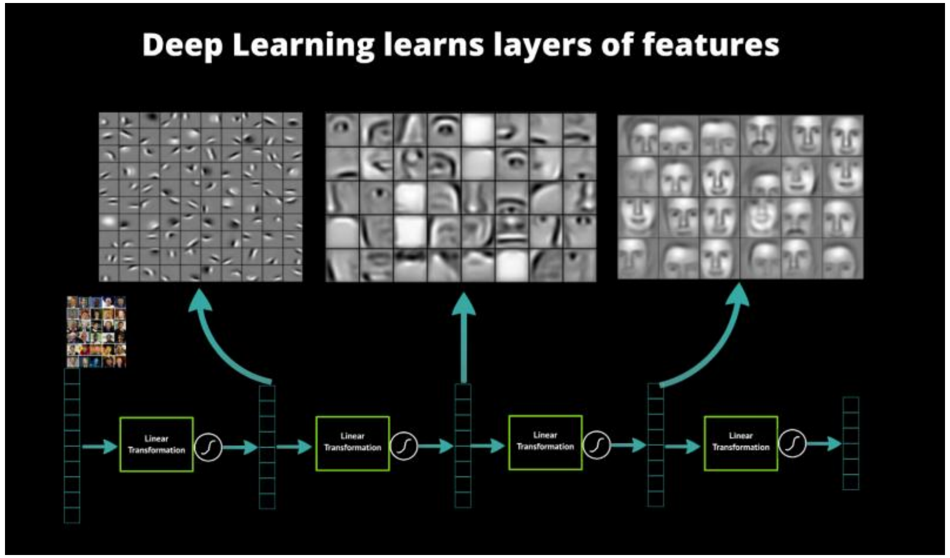
How To Transfer Style

Parametric Methods

- Optimization based → Gatys et al.
- Feed-forward based → AdaIN / WCT

Major Methods – Gatys et al.

- The very first paper defined the field
- Optimization based
- Uses pretrained CNN model (especially VGG)
  - Assume that the model captures some information about the input
    - If the models works well, it does extract edges and so on
    - Thus the feature map in the middle layer itself can be used as content feature



Leon A Gatys *et al.* Image style transfer using convolutional neural networks.

Major Methods – Gatys et al.

- What about style?
  - Defined by feature correlation, “Gram” Matrix
  - you can think them similar to covariance matrix

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}.$$

```
def gram_matrix(input):
    a, b, c, d = input.size()  # a=batch size(=1)
    # b=number of feature maps
    # (c,d)=dimensions of a f. map (N=c*d)

    features = input.view(a * b, c * d)  # resise F_XL into ₩hat F_XL

    G = torch.mm(features, features.t())  # compute the gram product

    # we 'normalize' the values of the gram matrix
    # by dividing by the number of element in each feature maps.
    return G.div(a * b * c * d)
```

Leon A Gatys *et al*. Image style transfer using convolutional neural networks.

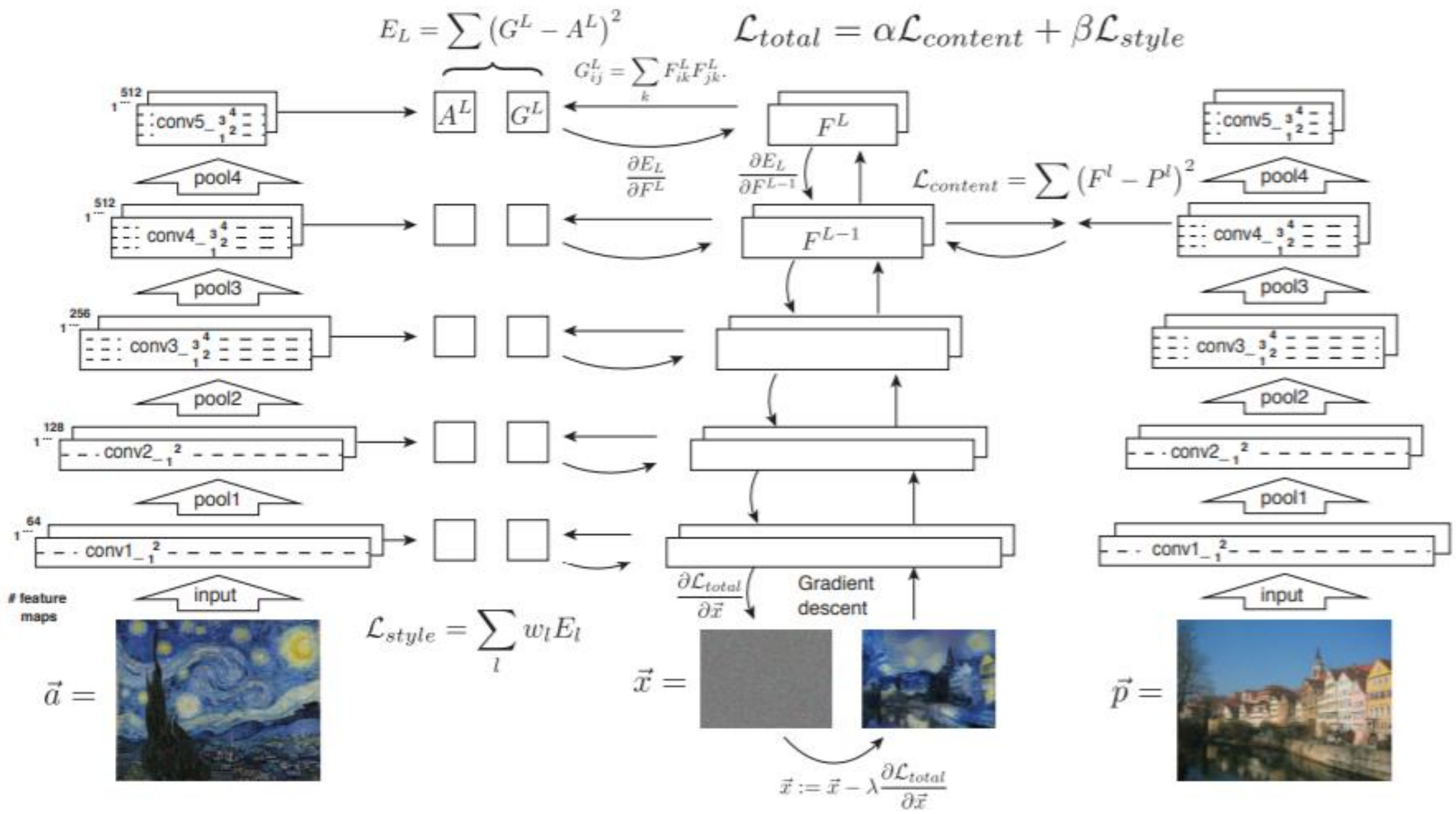
Major Methods – Gatys et al.

We can think style transfer as

- Make content feature's statistics closer to "style statistics"

Leon A Gatys *et al*. Image style transfer using convolutional neural networks.

Major Methods – Gatys et al.



Leon A Gatys *et al.* Image style transfer using convolutional neural networks.

Major Methods – Gatys et al.

Definition of loss

- Content loss
  - Difference of feature between transferred image and content image
- Style loss
  - Difference of gram matrix between transferred image and style image

Initialize transferred image with random noises

Consecutively “optimize” the transferred image to minimize above losses

Leon A Gatys *et al*. Image style transfer using convolutional neural networks.

Major Methods – Gatys et al.

- The quality is much better than other methods

- However the optimization based methods have some flaws
  - It takes much time for the optimization
  - Only one image can be transferred at once

Leon A Gatys *et al*. Image style transfer using convolutional neural networks.

Major Methods - Feed-Forward

- After Gatys et al. there are many researches developing style transfer methods. Their objectives can be summarized.
  - Make more faster methods
  - With no degradation of transferred images' quality

- Researchers proposed models using feed-forward.
  - By single feed-forward path, we can get transferred images
  - By using batch, we can transfer more than one image at once
  - However we need to train decoder
    - Once the decoder is trained, freeze it and transfer arbitrary pairs
    - Here, the methods to train decoders are omitted

Major Methods - Feed-Forward

- Content features are same as before
  - Just feature maps in the middle layer of VGG

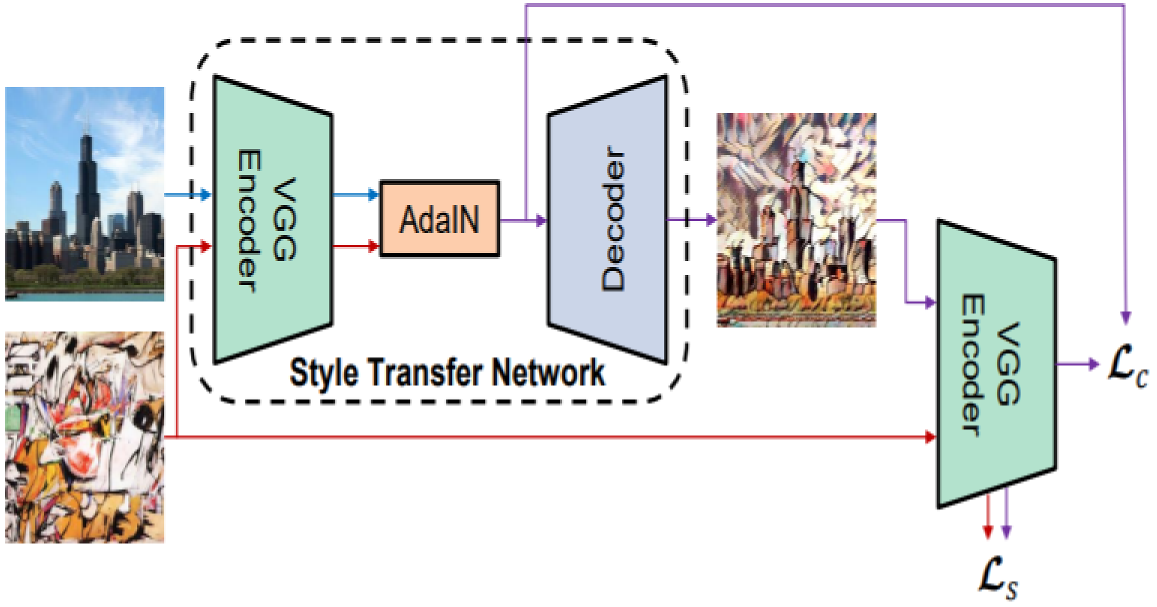
- The point is how to define “style”
  - What statistics we can use for “style”?

Major Methods – AdaIN

- One might think that

"Why don't we use mean and variance?"

- AdaIN implements that idea

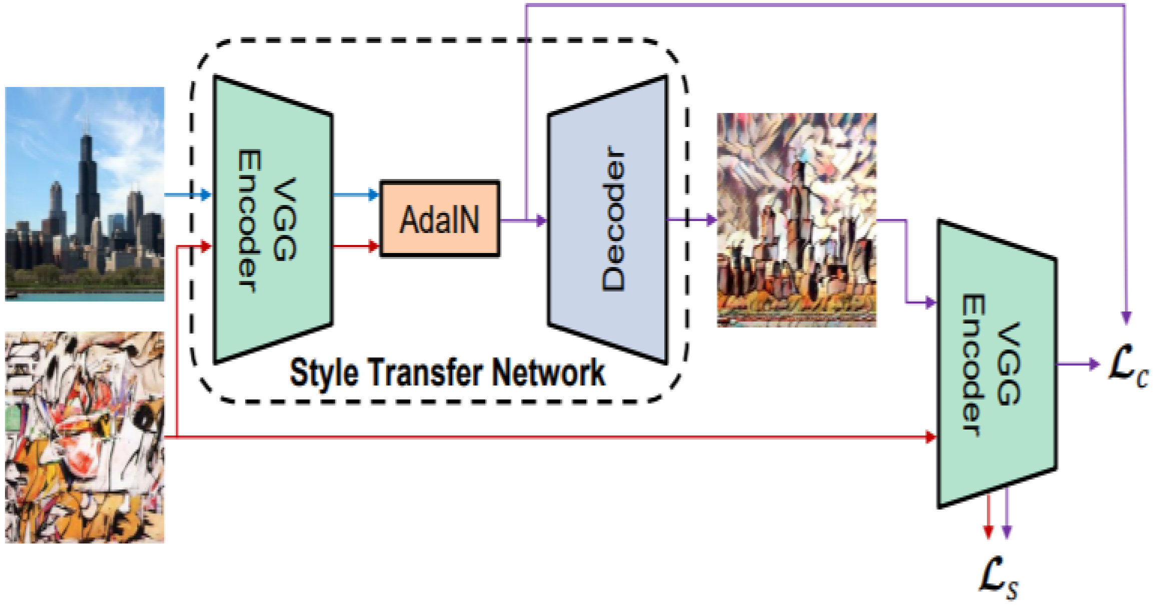


Xun Huang et al. Arbitrary style transfer in real-time with adaptive instance normalization.

Major Methods – AdaIN

- AdaIN module
  - Standardize the content feature
  - And then transfer it with style feature

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



Xun Huang et al. Arbitrary style transfer in real-time with adaptive instance normalization.

Major Methods – WCT

- WCT uses more complicated statistics
  1. Center content vector by subtracting its mean

  2. Then by using eigen decomposition, "whitening" the feature

     $$f_c f_c^\top = E_c D_c E_c^\top$$

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

     - If you just decode the whitened feature,
       it lose some color and texture information
     - Thus in eigen decomposition, there exists some style information

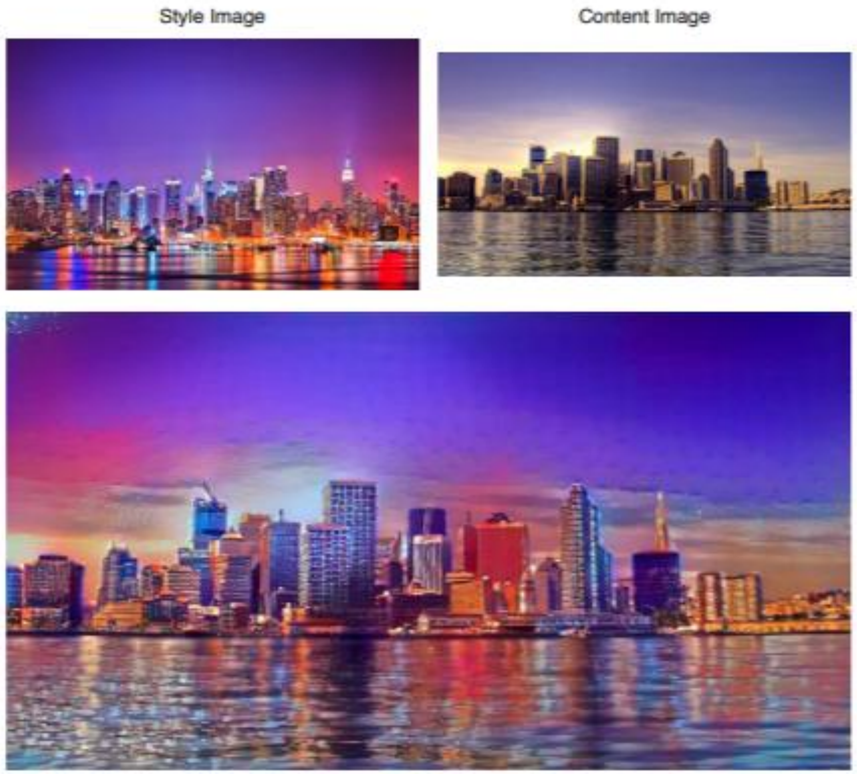
Yijun Li et al. Universal style transfer via feature transforms

Major Methods - WCT

- WCT uses more complicated statistics
  1. Center Style vector by subtracting its mean
  2. Then by using eigen decomposition, “Coloring” the feature
     $$\hat{f_{cs}} = E_s \, D_s^{\frac{1}{2}} \, E_s^{\top} \, \hat{f}_c$$
  3. Re-center it with style mean
  4. By decoding this feature, you can get the stylized image

Yijun Li et al. Universal style transfer via feature transforms

Photorealistic



- What if both content and style images are real photo?

−Day to night, summer to winter and so on

−Result must be look like realistic

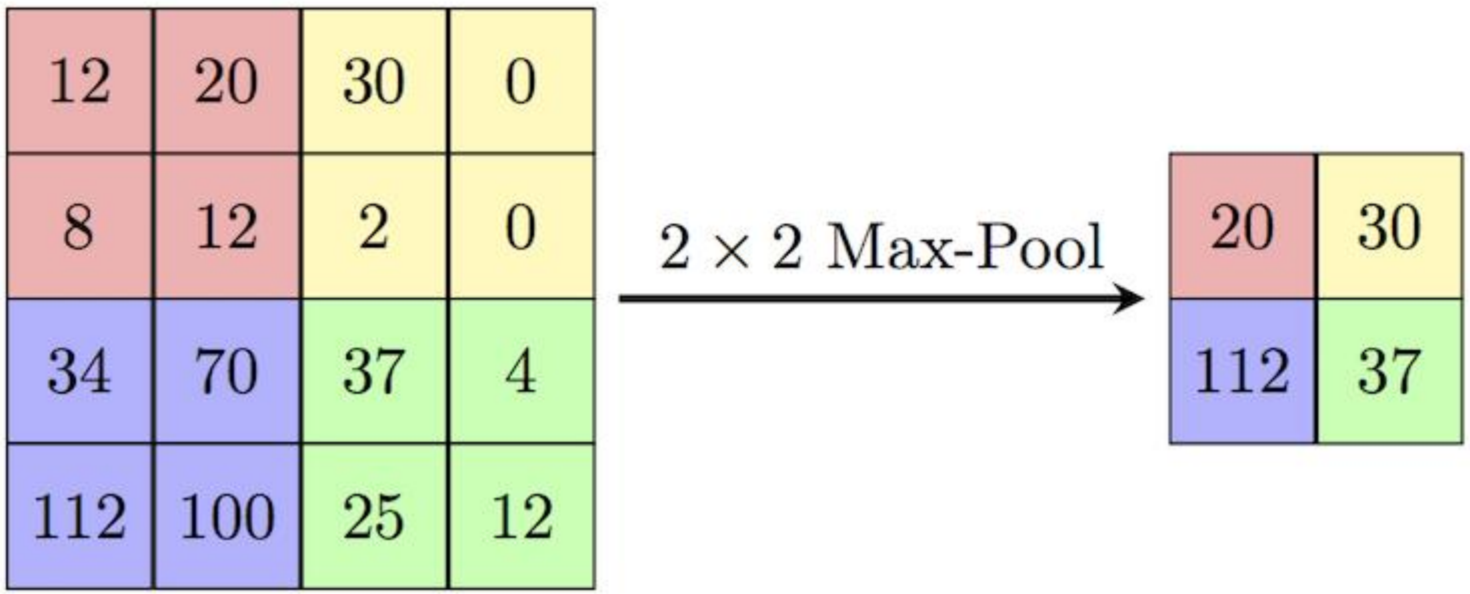
−We have to maintain the original content structure

- Ordinary style transfer methods make some distortion in transferred images
- Why?

Photorealistic

- Max pooling layer “compresses” the information

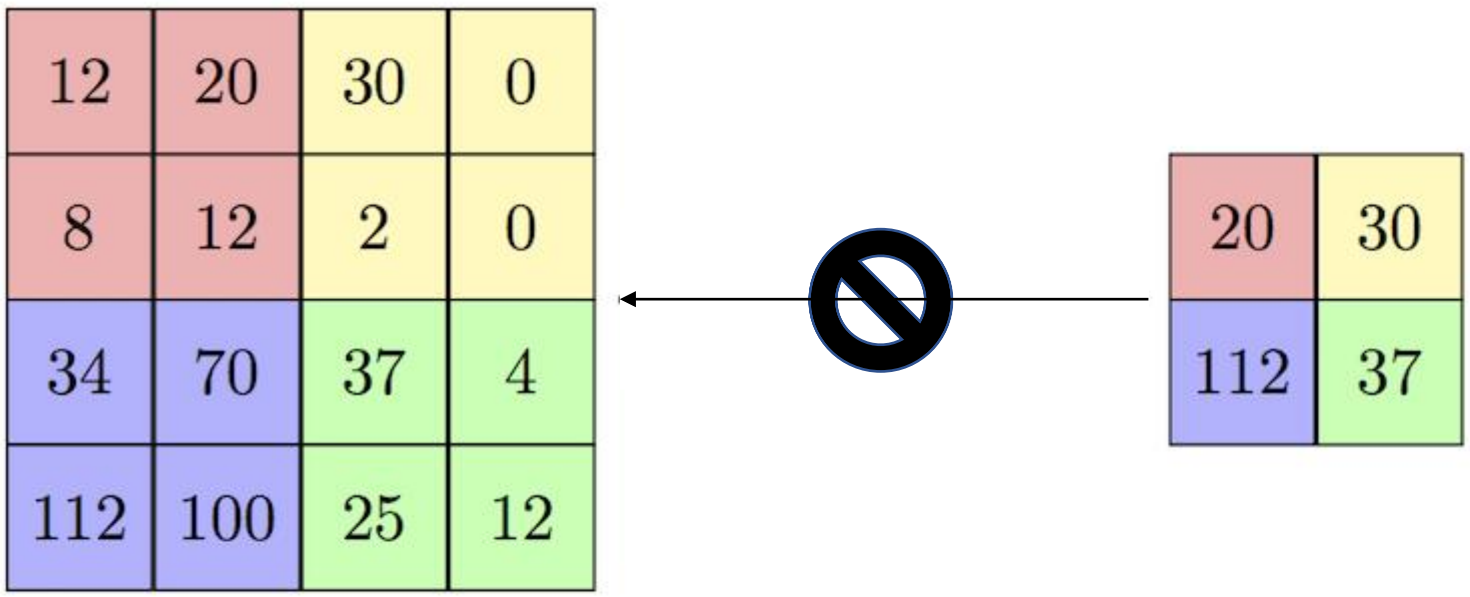
➡ the feature after pooling loses some information



Photorealistic

- Max pooling layer “compresses” the information

➡ the feature after pooling loses some information



Photorealistic

- Max pooling layer “compresses” the information

➡ the feature after pooling loses some information

- If you use CNN, you cannot restore original information

- In photorealistic style transfer, there exists some distortion in the results
  - The original content information is gone
  - Cannot remain original content structure

Photorealistic

- Max pooling layers are the problem

- What if we change them with other methods?

Wavelet Pooling

- Wavelet is used for signal processing
- It satisfies "perfect reconstruction"
  - It means you can reconstruct original image even after wavelet pooling
  - After the wavelet pooling, the feature map is decomposed to four smaller features
  - By using "wavelet unpooling", you can get the original feature map

Jaejun Yoo et al. Photorealistic Style Transfer via Wavelet Transforms.

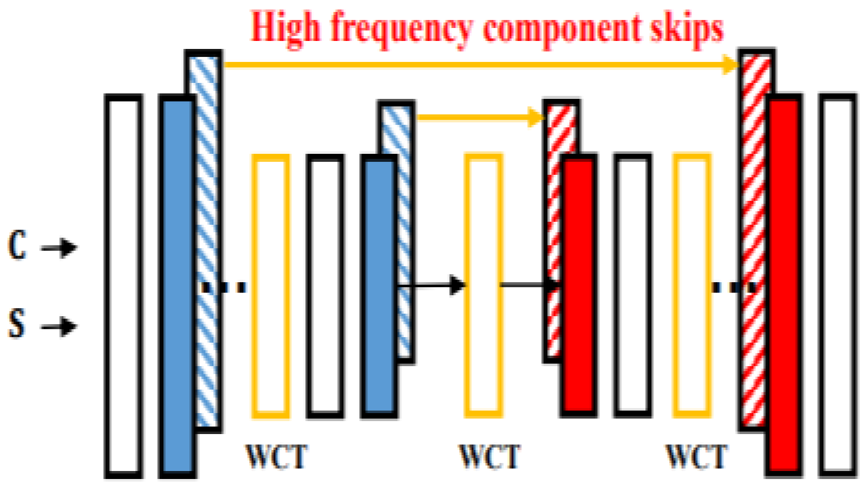
Wavelet Pooling

- The four components after wavelet pooling
- Top left has almost every texture information
- And other three have structural information

Jaejun Yoo et al. Photorealistic Style Transfer via Wavelet Transforms.

Wavelet Pooling

- Thus by stylizing only texture information and maintain other three information, we can restore original content structure with style transferred



Jaejun Yoo et al. Photorealistic Style Transfer via Wavelet Transforms.

Result

* All results are stylized with segmentation map for better stylization

Result

* All results are stylized with segmentation map for better stylization

Result

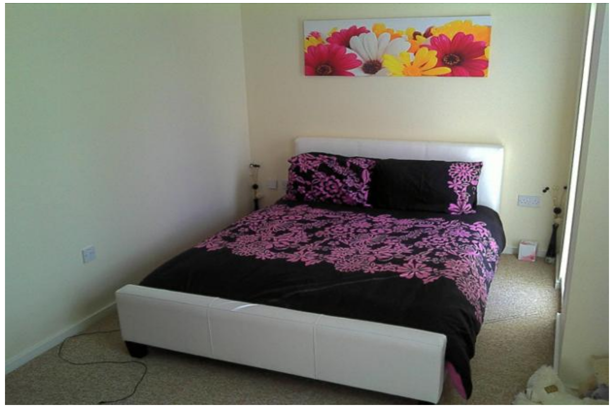
* All results are stylized with segmentation map for better stylization

Result

* All results are stylized with segmentation map for better stylization

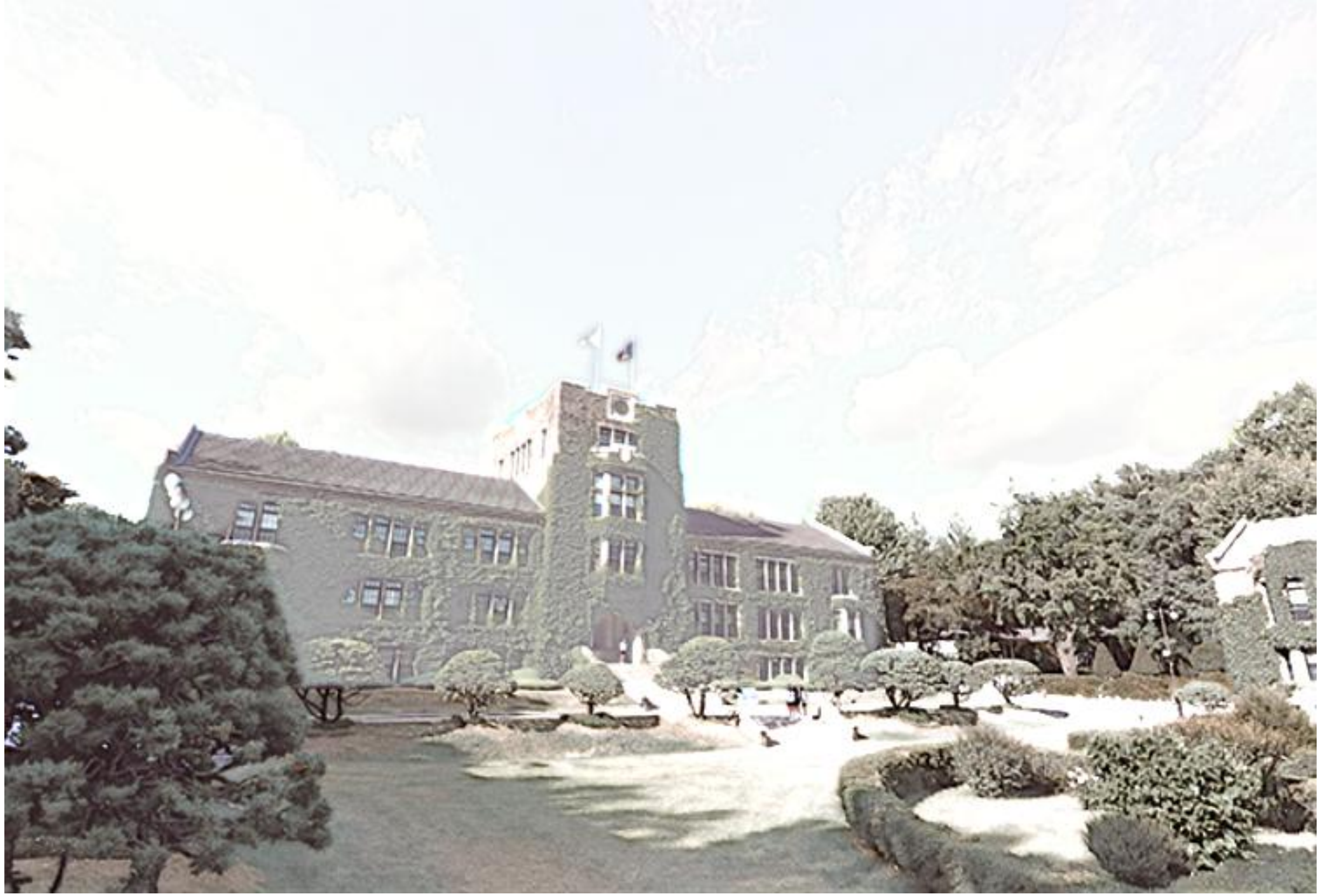
Failure Case

* All results are stylized with segmentation map for better stylization



Failure Case

* All results are stylized with segmentation map for better stylization



Further Improvement

- For qualitative stylization, you need segmentation map
  - Maybe we can use semantic mapping for automated guidance

- The network saves original structure information “too well”
  - If there is a mistake when you get a segmentation map,
    it will highlight that part
  - Need to soothe it

References

- Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2414 - 2423. IEEE, 2016

- Xun Huang and Serge J Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In ICCV, pages 1510 - 1519, 2017

- Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In Advances in Neural Information Processing Systems, pages 386 - 396, 2017.

- Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, Jung-Woo Ha. Photorealistic Style Transfer via Wavelet Transforms. In ICCV 2019