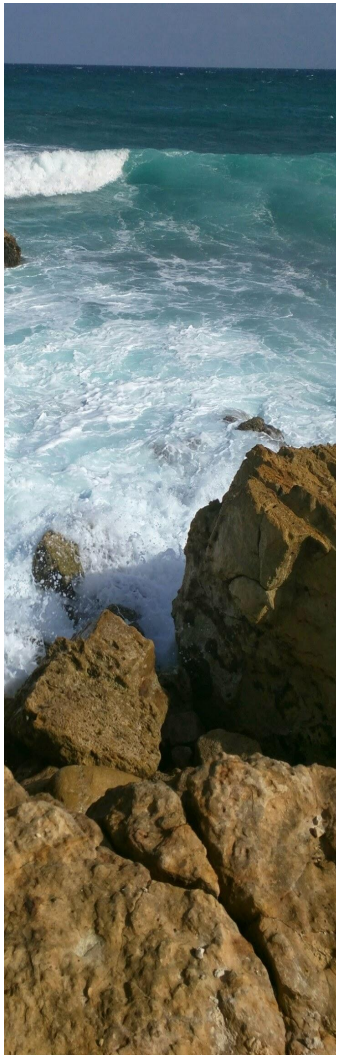
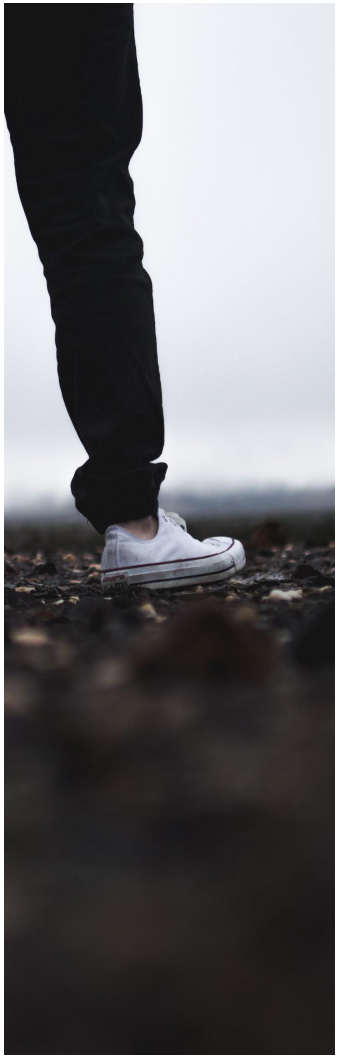
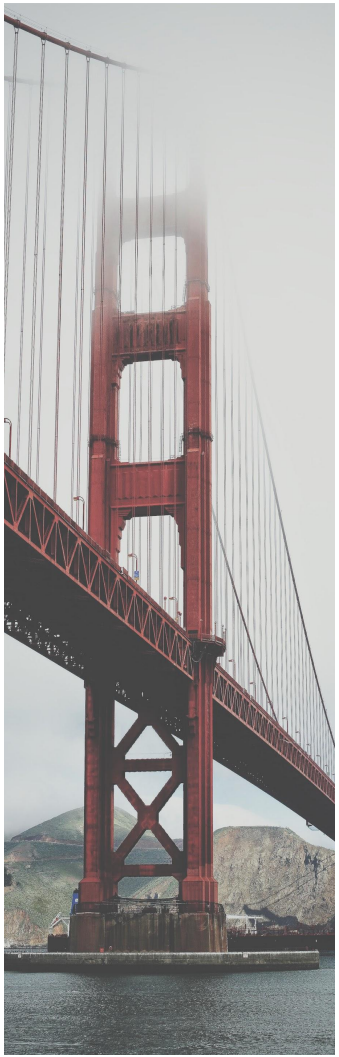




2D & 3D Style Transfer

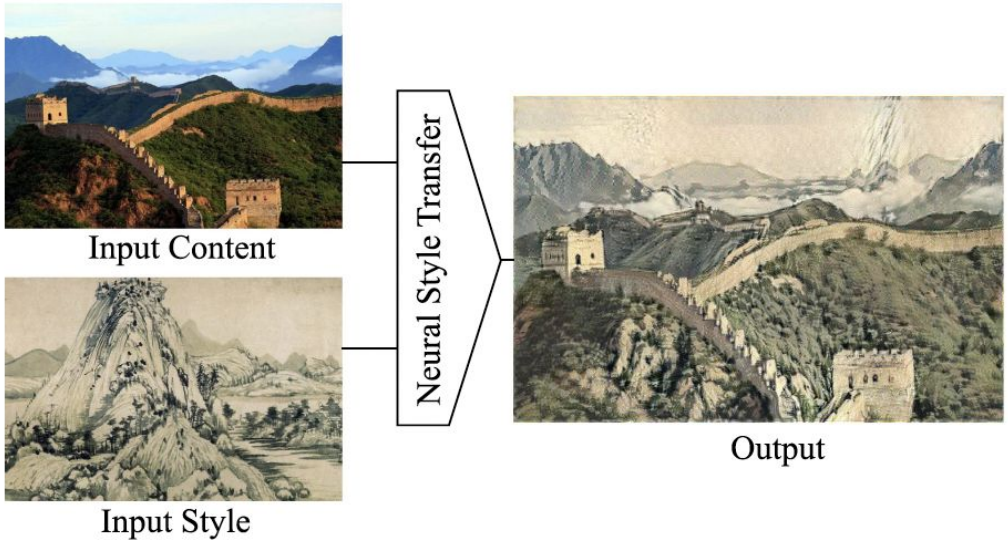
Yingnan Ma

Outline

- 2D Style Transfer
  - Artistic Style Transfer
  - Photo-realistic Style Transfer
- 3D Style Transfer
  - Mesh Style Transfer

2D Style Transfer

Arbitrary style transfer is a long-standing image processing topic, which aims to render an image with referenced arbitrary styles. The styles can be either artistic or photo-realistic.



Artistic Style Transfer

Gatys et al. CVPR ‘16

using CNN representation (VGG)

Artistic Style Transfer

Photo-realistic Style Transfer

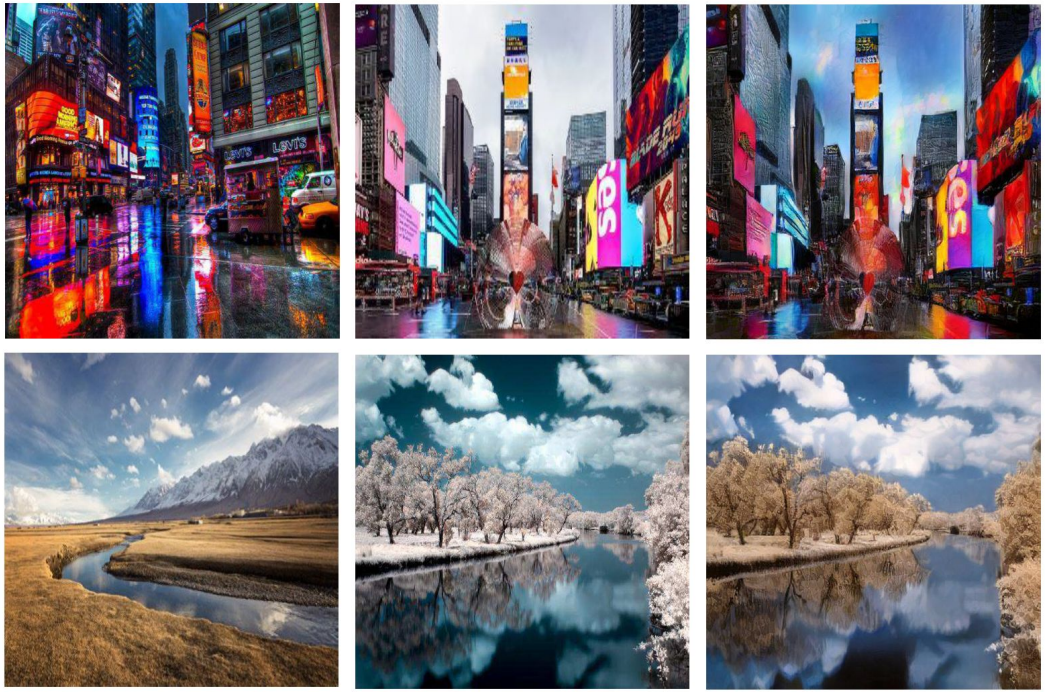


Photo-realistic Style Transfer

Applications:

- Day to night transfer
- Day to sunset transfer

What is Style Transfer?

Style Transfer

- Domain Adaptation
- Domain Translation
- Data Augmentation

Style Transfer

- Representation Learning

Traditional Texture Synthesis

Vision & Nature Image

"Vision is the process of extracting information from the images that enter the eye. The set of all possible images is vast, and yet only a small fraction of these are likely to be encountered in a natural setting."

"It has proven difficult to characterize this set of "natural" images, using either deterministic or statistical models."

Visual Texture?

"Loosely speaking, texture images are specially homogeneous and consist of repeated elements, often subject to some randomization in their location, size, color, orientation, etc."

"The class of images that we commonly call "visual texture" seems most amenable to statistical modeling."

Traditional Texture Synthesis

Two important developments

(enabling a new generation of more powerful statistical texture models)

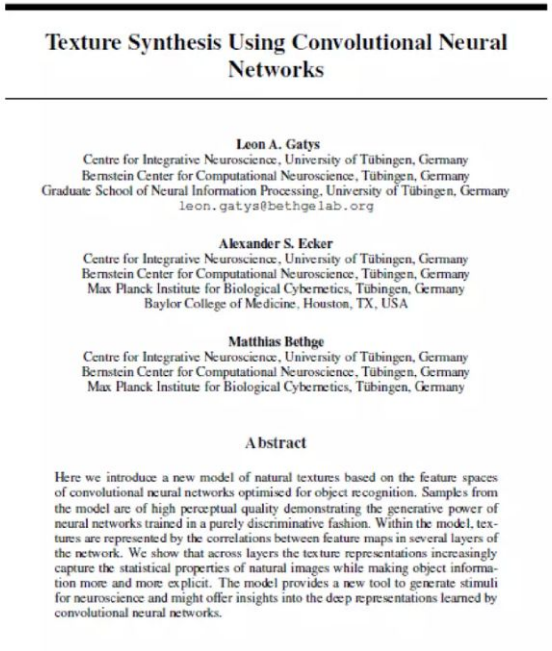
1. **The theory of Markov random fields**, in which the full model is characterized by statistical interactions within local neighborhoods
2. **The use of oriented linear kernels** at multiple spatial scales for image analysis and representation, e.g., wavelets

Limitations

1. Not capable of simulating an arbitrary style, which is not flexible.
   e.g., Stroke-Based Rendering, Region-Based Techniques
2. Only exploiting low-level features
   e.g., Example-Based Rendering (Image Analogies)
3. Very limited in style diversity
   e.g., Image Processing and Filtering

Texture Synthesis Using CNNs

- 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

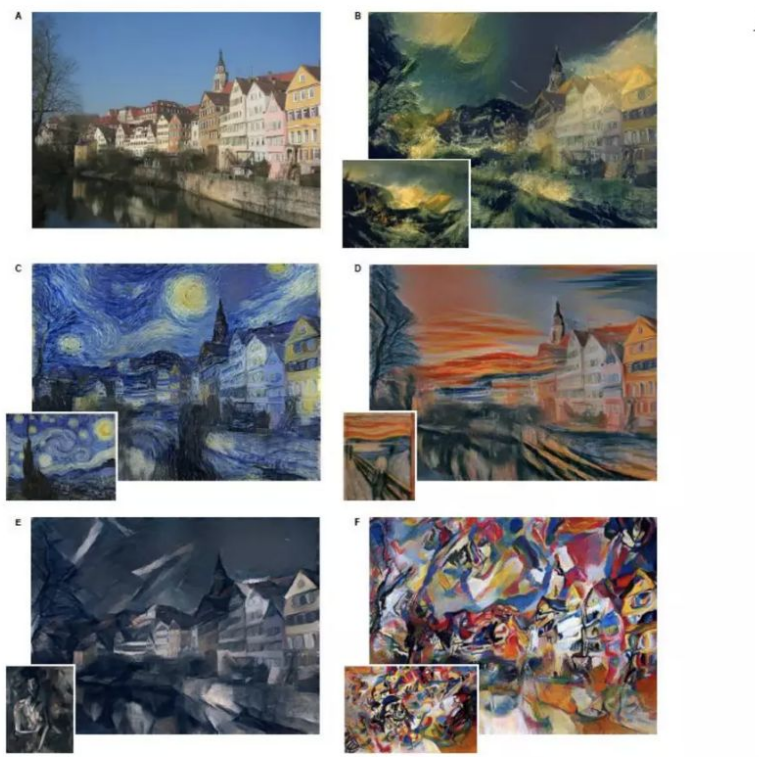
Alexander S. Ecker
Centre for Integrative Neuroscience, University of Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
Max Planck Institute for Biological Cybernetics, Tübingen, Germany
Baylor College of Medicine, Houston, TX, USA

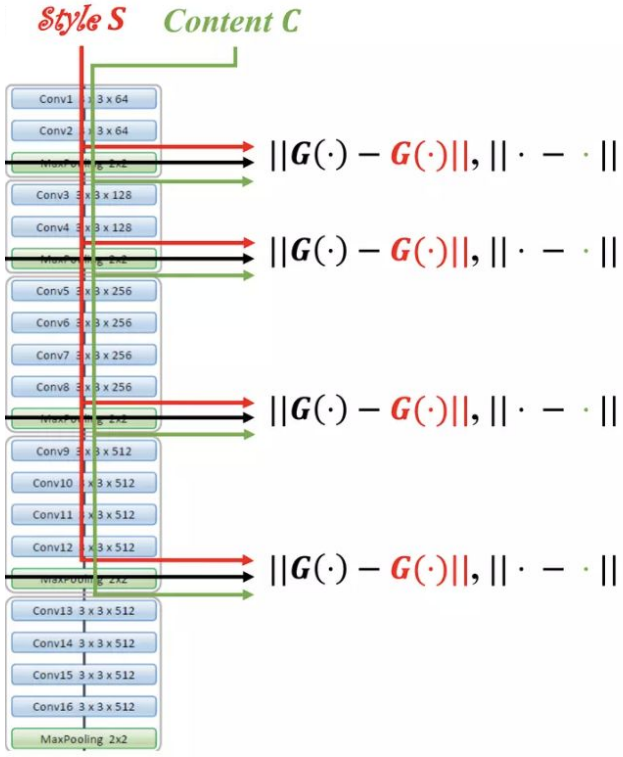
Matthias Bethge
Centre for Integrative Neuroscience, University of Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
Max Planck Institute for Biological Cybernetics, Tübingen, Germany

Abstract

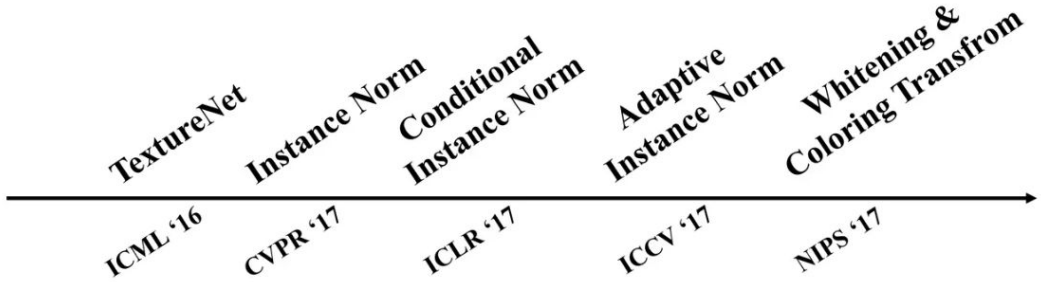
Here we introduce a new model of natural textures based on the feature spaces of convolutional neural networks optimised for object recognition. Samples from the model are of high perceptual quality demonstrating the generative power of neural networks trained in a purely discriminative fashion. Within 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





- from Gatys *et al.*, CVPR 2016

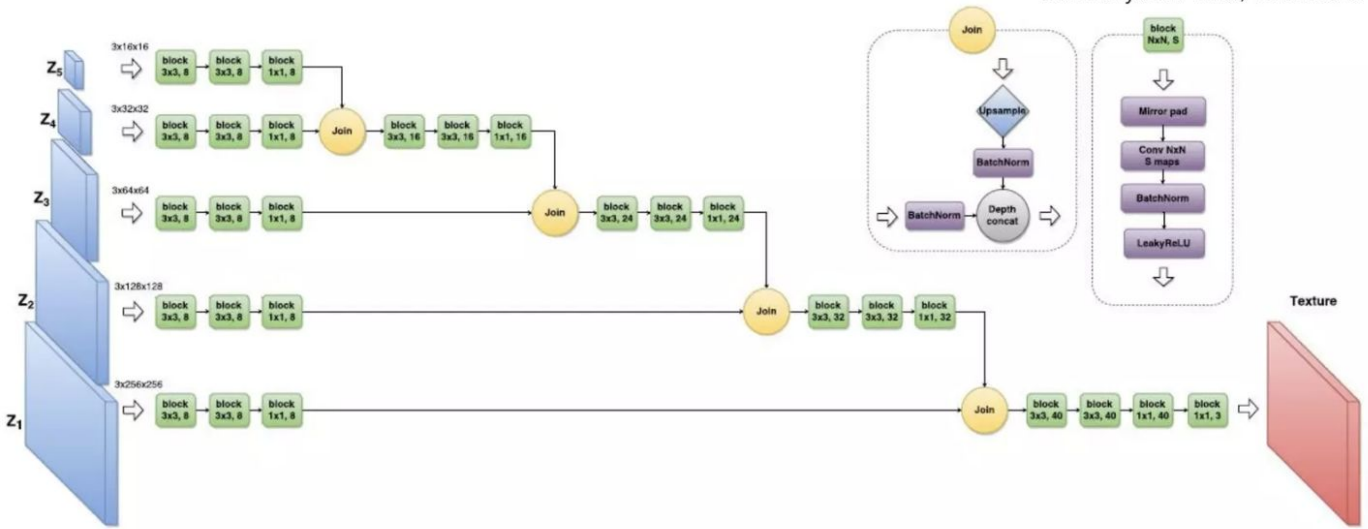


Recent Developments of Style Transfer

Feed-forward network

- **Iterative optimization is slow.**

- from Unyanov *et al.,* ICML 2016

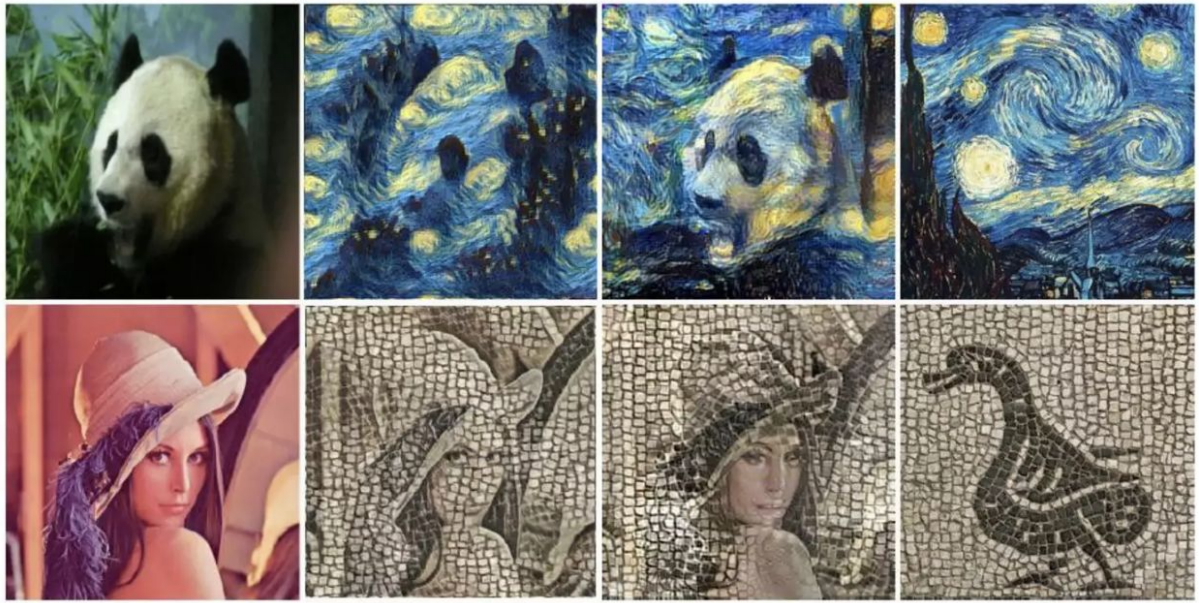


"Let's move the optimization cost into the training phase."

$$\theta_{\mathbf{x_0}} = \underset{\theta}{\operatorname{argmin}}\ E_{\mathbf{z}\sim\mathcal{Z}}\left[\mathcal{L}_T\left(\mathbf{g}(\mathbf{z};\theta), \mathbf{x_0}\right)\right]. \qquad \mathcal{L}_T(\mathbf{x};\mathbf{x_0}) = \sum_{l\in L_T} \|G^l(\mathbf{x}) - G^l(\mathbf{x_0})\|_2^2$$

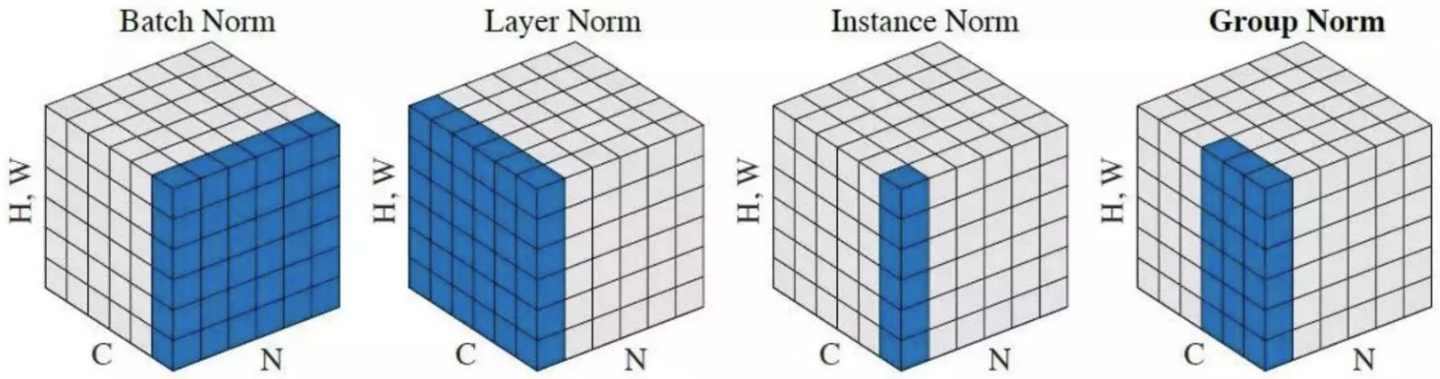
Feed-forward network

A network per a style



- from Unyanov *et al.,* ICML 2016

Instance Normalization (IN)

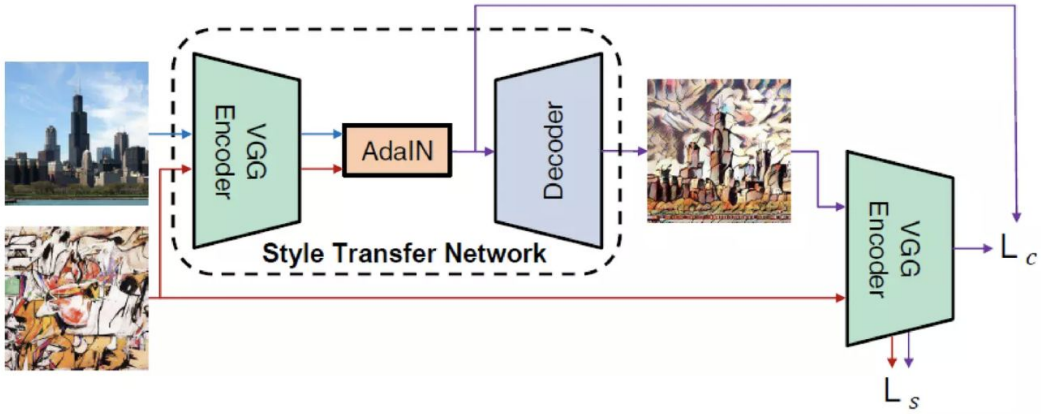


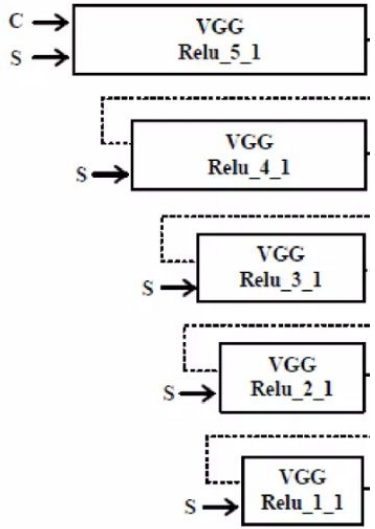
$$\mathrm{IN}(x) = \gamma\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \beta$$

For style transfer,

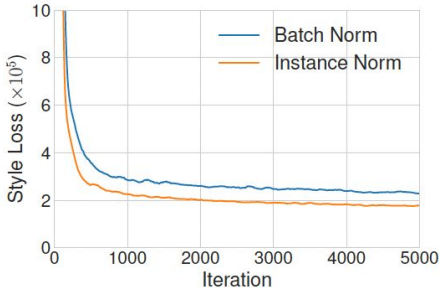
instance normalization is preferable as it can normalize each individual content image x0.

- from Unyanov *et al.,* CVPR 2017

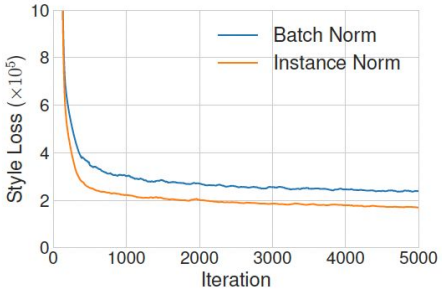




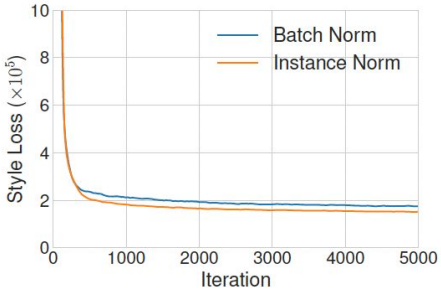
$$\text{AdaIN}(x, y) = \sigma(y)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y)$$



(a) Trained with original images.

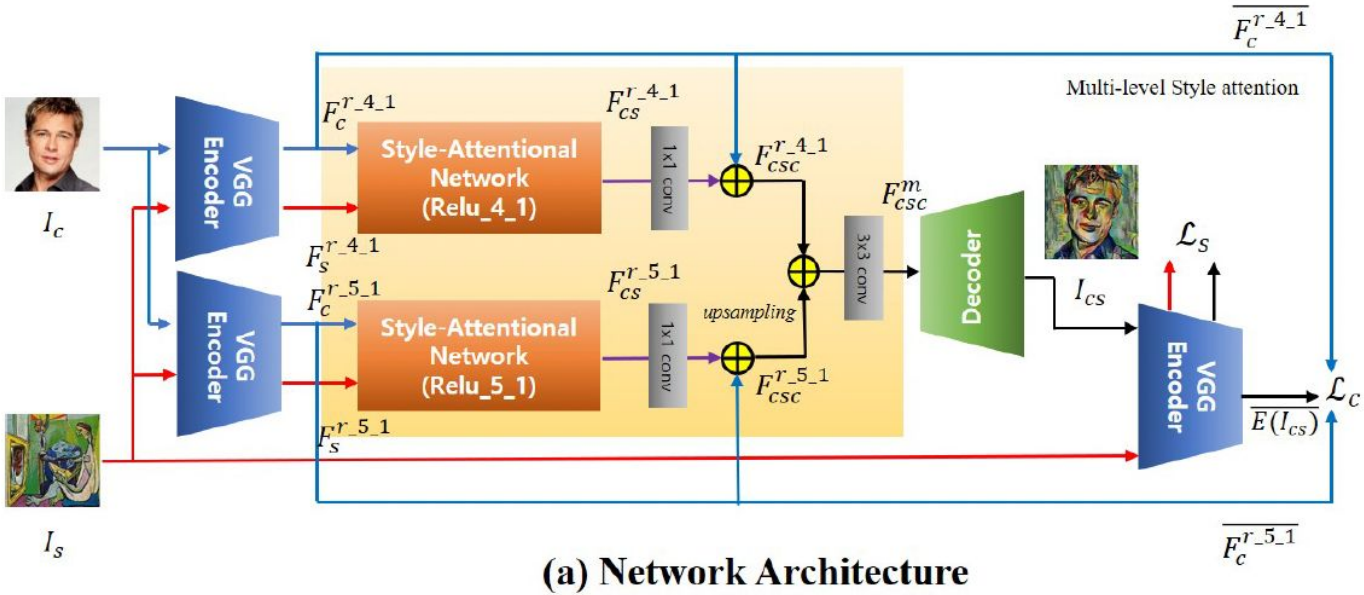


(b) Trained with contrast normalized images.



(c) Trained with style normalized images.

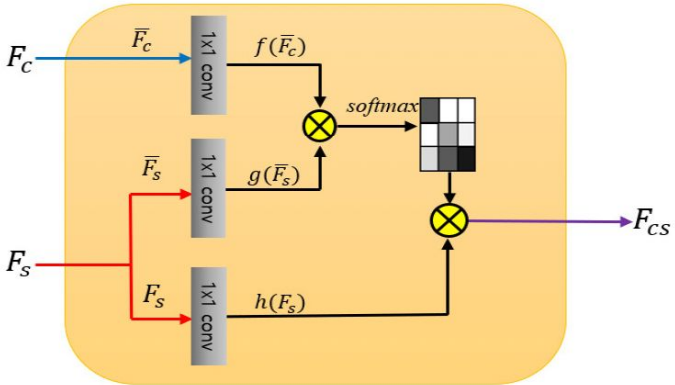
SANet



(a) Network Architecture

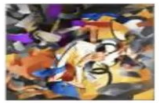
SANet

- Self-Attention Mechanism. The style-attentional module can calculate the response at a position in a sequence or an image by attending to all positions. The proposed SANet learns the mapping between the content features and the style features



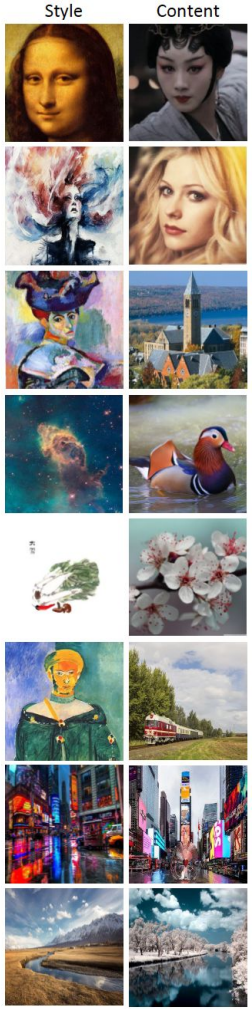
Loss Functions

$\mathcal{L}_{\text{content}}$ ( , ) ≈ 0

$\mathcal{L}_{\text{style}}$ ( , ) ≈ 0

$$f_s(\mathbf{I}_1, \mathbf{I}_2) = \sum_{i=1}^{L} \|\mathbb{E}(\phi_i(\mathbf{I}_1)) - \mathbb{E}(\phi_i(\mathbf{I}_2))\|_2 + \|\sigma(\phi_i(\mathbf{I}_1)) - \sigma(\phi_i(\mathbf{I}_2))\|_2$$

$$f_c(\mathbf{I}_1, \mathbf{I}_2) = \|\phi_{4\_1}(\mathbf{I}_1) - \phi_{4\_1}(\mathbf{I}_2)\|_2 + \|\phi_{5\_1}(\mathbf{I}_1) - \phi_{5\_1}(\mathbf{I}_2)\|_2 .$$

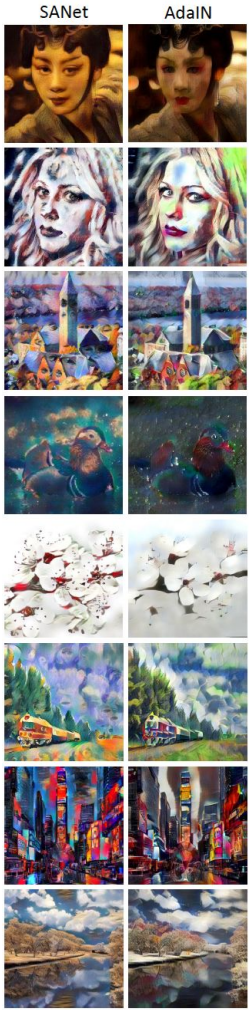


Limitations

Although existing algorithms can achieve the delivery of styles, the 2D style transfer task still has challenges maintaining a balance between content preservation and style embedding.

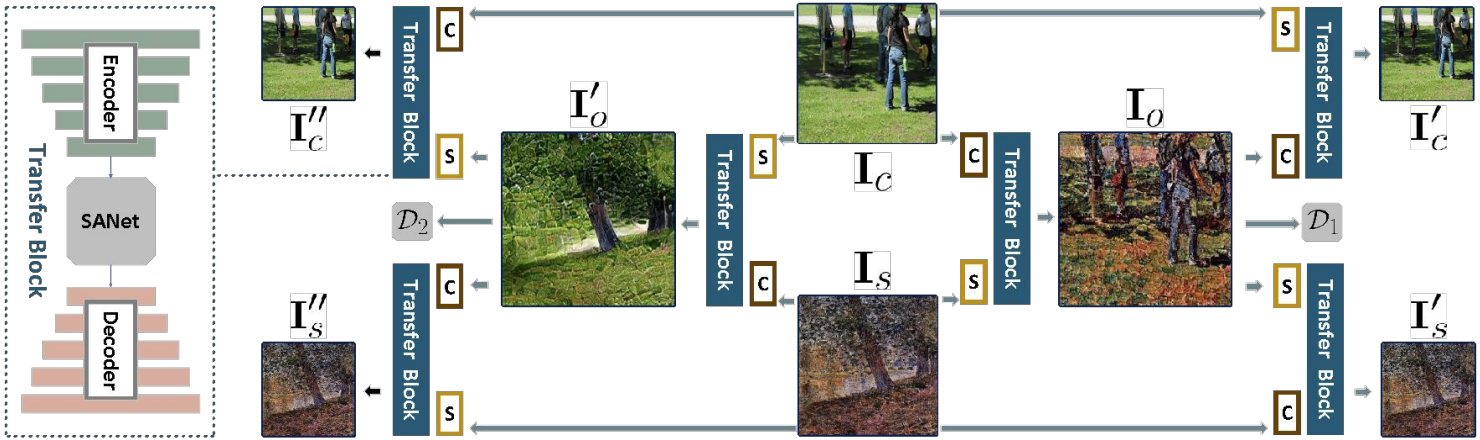
- In artistic style transfer, it is challenging to preserve the content details during artistic style embedding due to style differences. Excessive style embedding can result in changes in content information.

- For photo-realistic style transfer, the architecture pays more attention to content preservation, which can result in a lack of delicate patterns.



RAST

- We employ SANet as the backbone, which can map the correspondence between content feature map and style feature map semantically by calculating style attention.
- The pre-trained VGG-19 network is utilized as the encoder to obtain feature maps. The decoder is a symmetric VGG-19 network.
- Apart from this, multi-scale discriminators are applied as external discriminators to learn human-aware style information.



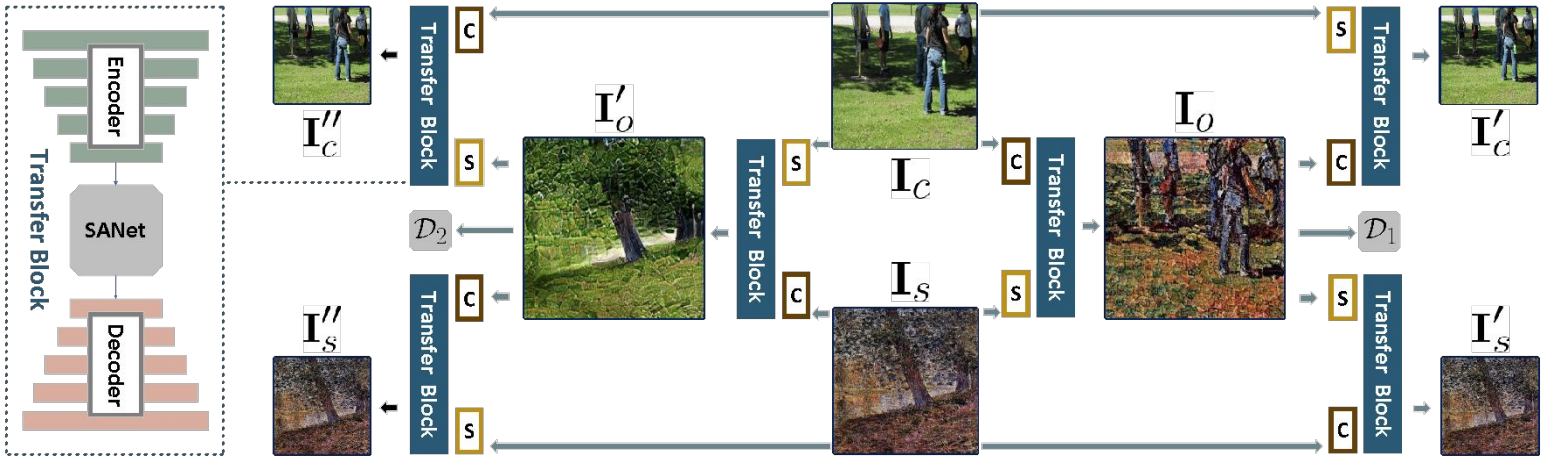
$$\mathbf{I}_o = \mathcal{T}(\mathbf{I}_c, \mathbf{I}_s) \equiv \mathbf{I}_c \xrightarrow{\mathbf{I}_s} \mathbf{I}_o$$

$$\mathbf{I}_c \xrightarrow{\mathbf{I}_s} \mathcal{T}(\mathbf{I}_c, \mathbf{I}_s) = \mathbf{I}_o \xrightarrow{\mathbf{I}_c} \mathcal{T}(\mathbf{I}_o, \mathbf{I}_c) = \mathbf{I}'_c \approx \mathbf{I}_c$$

$$\mathbf{I}_c \xrightarrow{\mathbf{I}_s} \mathcal{T}(\mathbf{I}_c, \mathbf{I}_s) = \mathbf{I}_o \xrightarrow{\mathbf{I}_c|\mathbf{I}_s} \begin{cases} \mathcal{T}(\mathbf{I}_o, \mathbf{I}_c) = \mathbf{I}'_c \approx \mathbf{I}_c \\ \mathcal{T}(\mathbf{I}_s, \mathbf{I}_o) = \mathbf{I}'_s \approx \mathbf{I}_s \end{cases}$$

$$\mathbf{I}_s \xrightarrow{\mathbf{I}_c} \mathcal{T}(\mathbf{I}_s, \mathbf{I}_c) = \mathbf{I}'_o \xrightarrow{\mathbf{I}_s|\mathbf{I}_c} \begin{cases} \mathcal{T}(\mathbf{I}'_o, \mathbf{I}_s) = \mathbf{I}''_s \approx \mathbf{I}_s \\ \mathcal{T}(\mathbf{I}_c, \mathbf{I}'_o) = \mathbf{I}''_c \approx \mathbf{I}_c \end{cases}$$

In the proposed approach, we re-input the stylized image of the proposed approach and the style of the content image, with the new output expected to approach the original content image



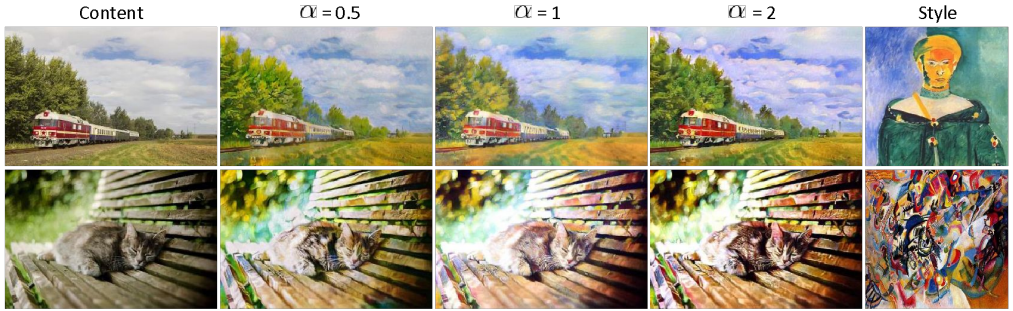
- To ensure the accuracy of the restoration, we design the feature-level multi-restoration loss and style difference loss to ensure feature consistency and style embedding, respectively.
- We utilize perceptual loss as the base functions to calculate the feature difference.

$$f_s(\mathbf{I}_1, \mathbf{I}_2) = \sum_{i=1}^{L} \|\mathbb{E}(\phi_i(\mathbf{I}_1)) - \mathbb{E}(\phi_i(\mathbf{I}_2))\|_2 + \|\sigma(\phi_i(\mathbf{I}_1)) - \sigma(\phi_i(\mathbf{I}_2))\|_2$$

$$f_c(\mathbf{I}_1, \mathbf{I}_2) = \|\phi_{4\_1}(\mathbf{I}_1) - \phi_{4\_1}(\mathbf{I}_2)\|_2 + \|\phi_{5\_1}(\mathbf{I}_1) - \phi_{5\_1}(\mathbf{I}_2)\|_2 .$$

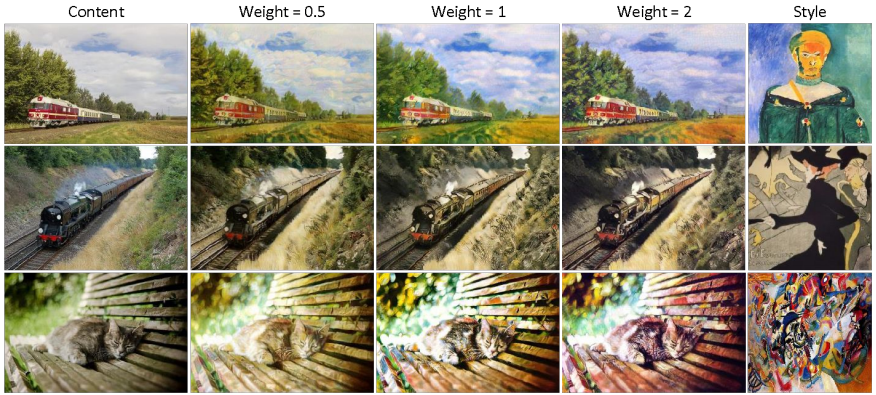
- Based on the previous functions, we design the multi-restoration loss, which can not only calculate the feature difference between restored images and input images but also measure the differences among multi-restored images.

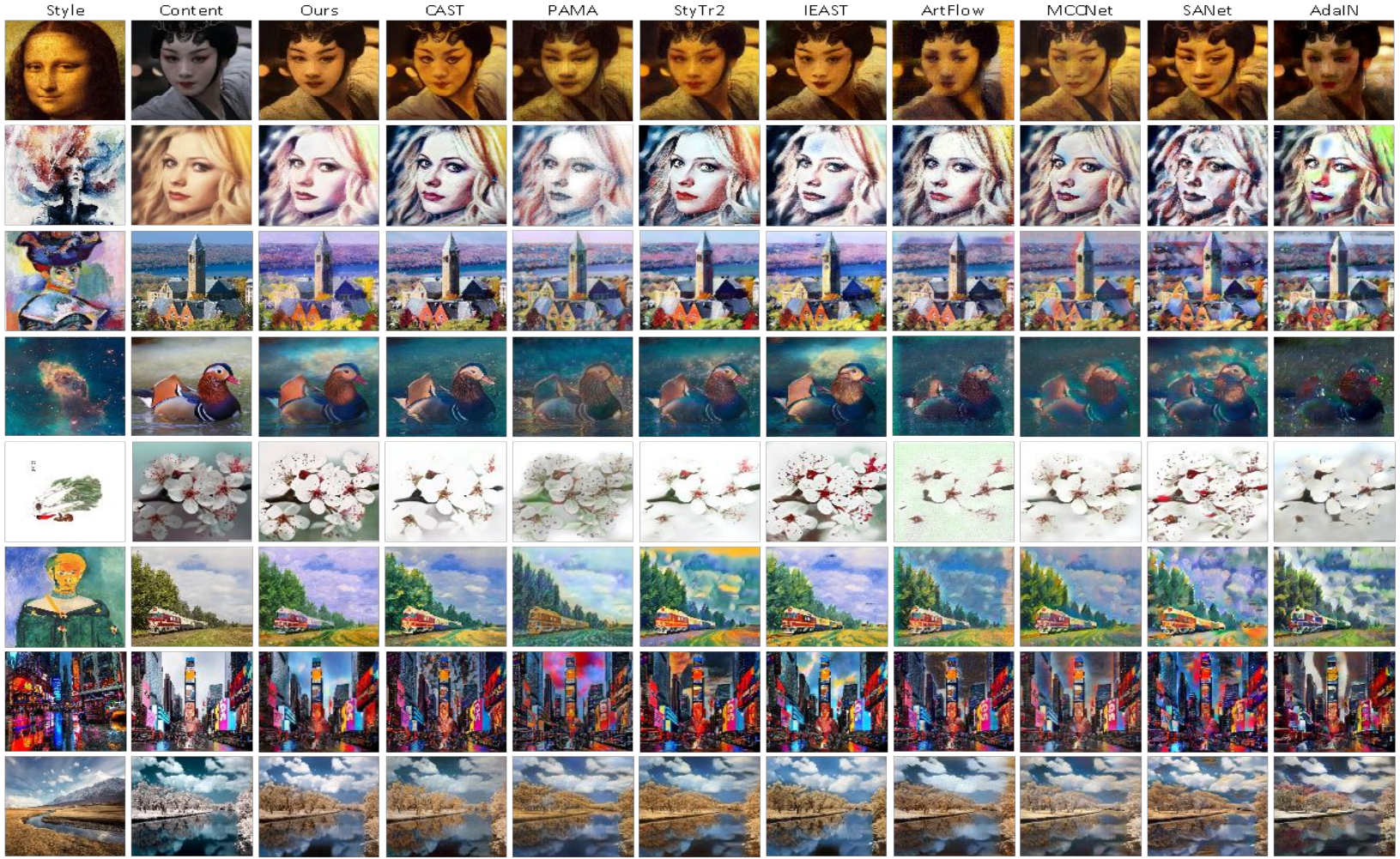
$$\mathcal{L}_{multi} = f_c(\mathbf{I}'_c, \mathbf{I}_c) + f_c(\mathbf{I}''_s, \mathbf{I}_s) + f_s(\mathbf{I}'_s, \mathbf{I}_s) + f_s(\mathbf{I}''_c, \mathbf{I}_c) + \alpha[f_c(\mathbf{I}'_c, \mathbf{I}''_c) + f_s(\mathbf{I}'_c, \mathbf{I}''_c) + f_c(\mathbf{I}'_s, \mathbf{I}''_s) + f_s(\mathbf{I}'_s, \mathbf{I}''_s)].$$



- In addition, we also design the style difference loss. To avoid the stylized images converging to content images, we design the style difference loss to maximize the style difference between content images and stylized images.

$$\mathcal{L}_{diff} = f_s(\mathbf{I}_o, \mathbf{I}_c) + f_s(\mathbf{I}'_o, \mathbf{I}_s).$$




Style
Content
Ours
CAST
PAMA
StyTr2
IEAST
ArtFlow
MCCNet
SANet
AdaIN

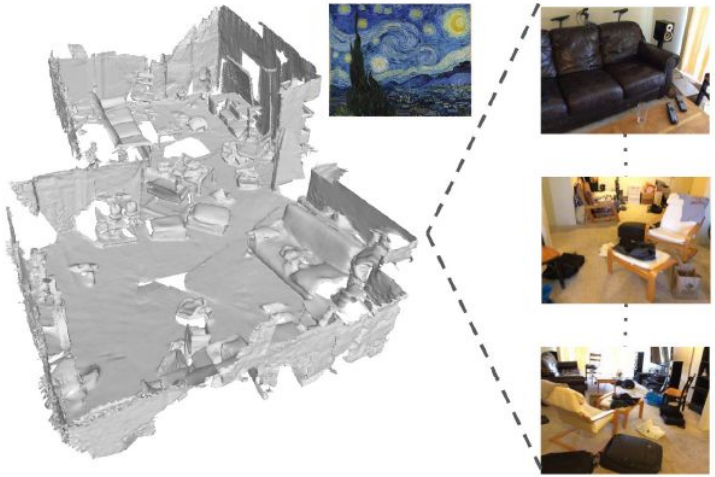
Methods

- 2D Style Transfer
  - Artistic Style Transfer
  - Photo-realistic Style Transfer
  - Other Style Transfer

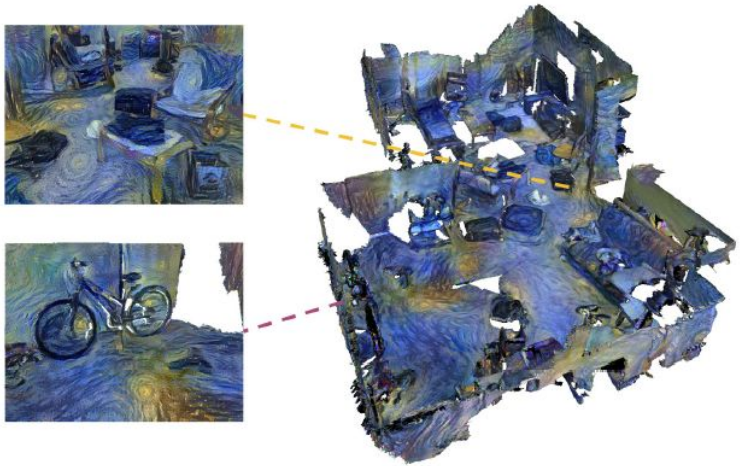
- 3D Style Transfer
  - Mesh Style Transfer

3D Style Transfer

Neural Style Transfer (NST) shows great results for stylization of images or videos, but stylization of 3D content like meshes has been underexplored. In 2022, StyleMesh is proposed, which can synthesize a texture for the mesh which is a combination of observed RGB colors and a painting's artistic style.



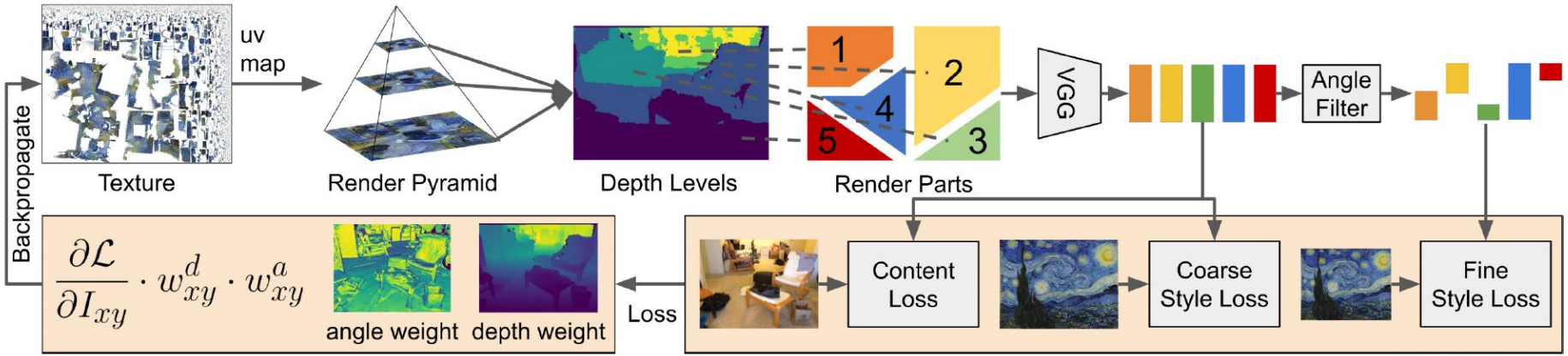
Input: RGB-D Scan + Style Image



Output: Mesh + Styled Texture

StyleMesh

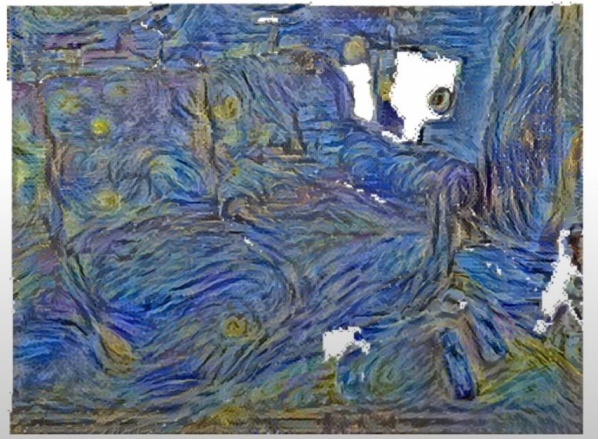
- It collects uv map from the texture and samples the texture at multiple resolutions, resulting in a rendered pyramid.
- By using the Depth information, it divide the higher resolution image into parts to get the coarse results.
- Encode each part by VGG network.



StyleMesh

- It uses the lower resolution image for the optimization to get the fine results.
- Use angle weight and depth weight to do the backpropagation.

stylized with fine details



stylized with coarse details

Limitations

- Lack of 3D training data.
- Limitation of computing resources

Thank you for listening