Painting Style Learning based on Neural Style Transfer

--- Exampled on Kristy Chu

Name: Yulin Zhou
ID: i6336858

Contents

Introduction

Task: Re-drawing an image in a given style

Mid-1990s

Manually = A well-trained artist + Lots of time

Automatically

NPR

(non-photorealistic rendering)

2016

Arbitrarily

NST

(Neural style transfer)

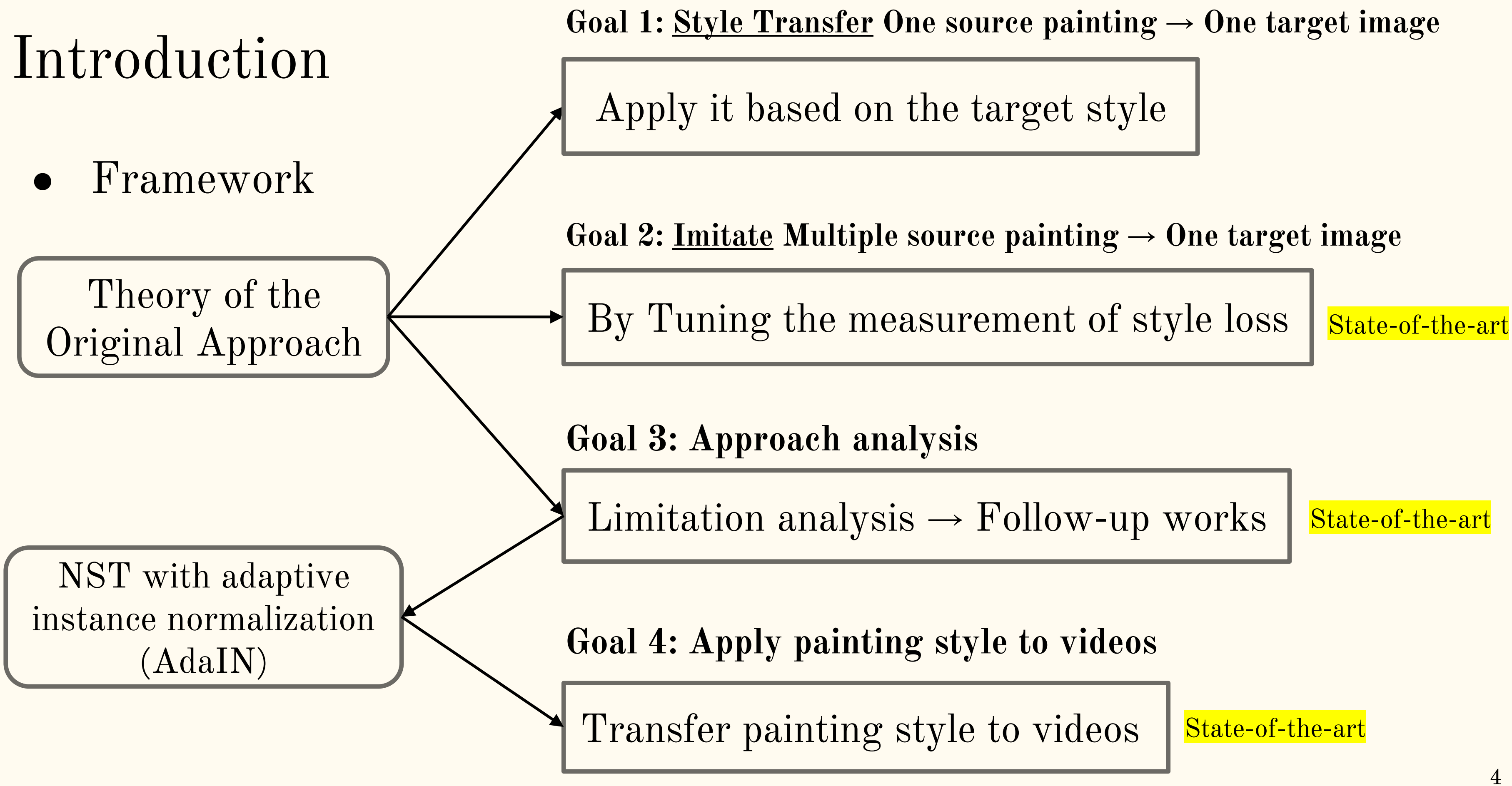
Limitation:

1. Designed for particular artistic styles
2. Cannot be easily extended to arbitrary styles

Finding: CNN can model the content and style independently
Idea: Iteratively optimize the output image with the objective of matching the desired feature distribution

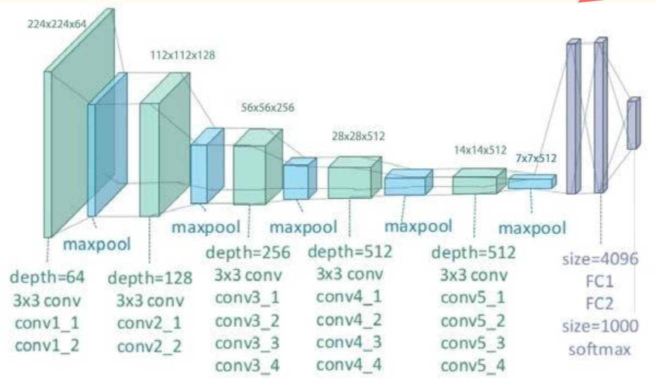
Introduction

- Framework

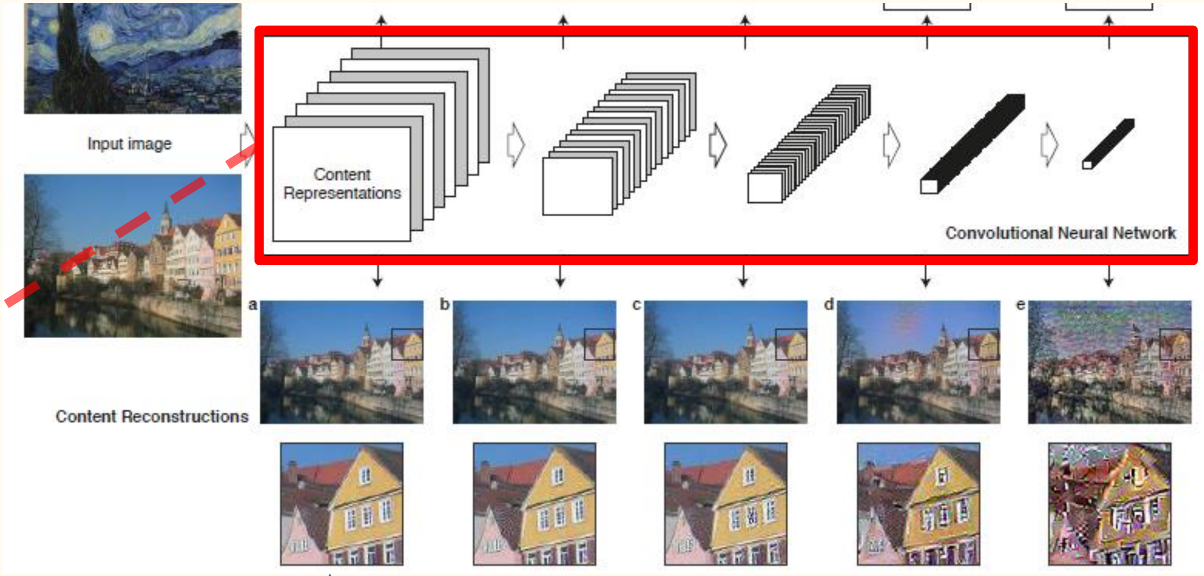


Original Approach

- Key Finding - Content



VGG-19 Network

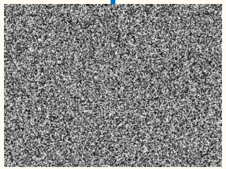


Loss function

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

Conv1_1 Conv1_1

$\vec{x}$

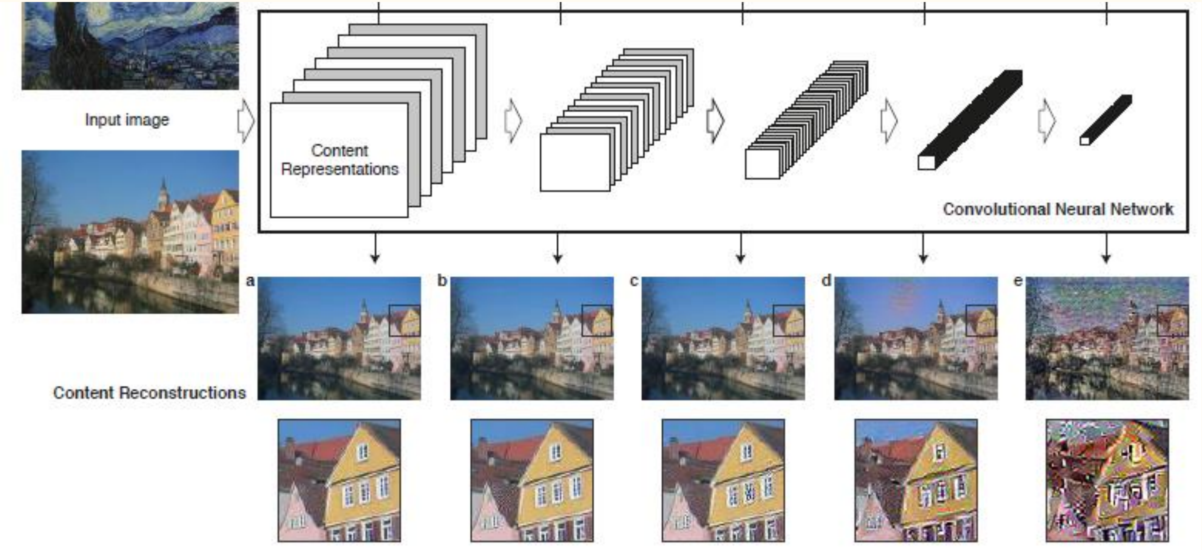


$\vec{p}$

F_{ij}^{l} (P_{ij}^{l}): the activation of the i^{th} filter at position j in layer l of $\vec{x}$ ($\vec{p}$)

Original Approach

- Key Finding - Content



Higher layer
→ Lose more exact pixel values
→ High-level content

Content Representation

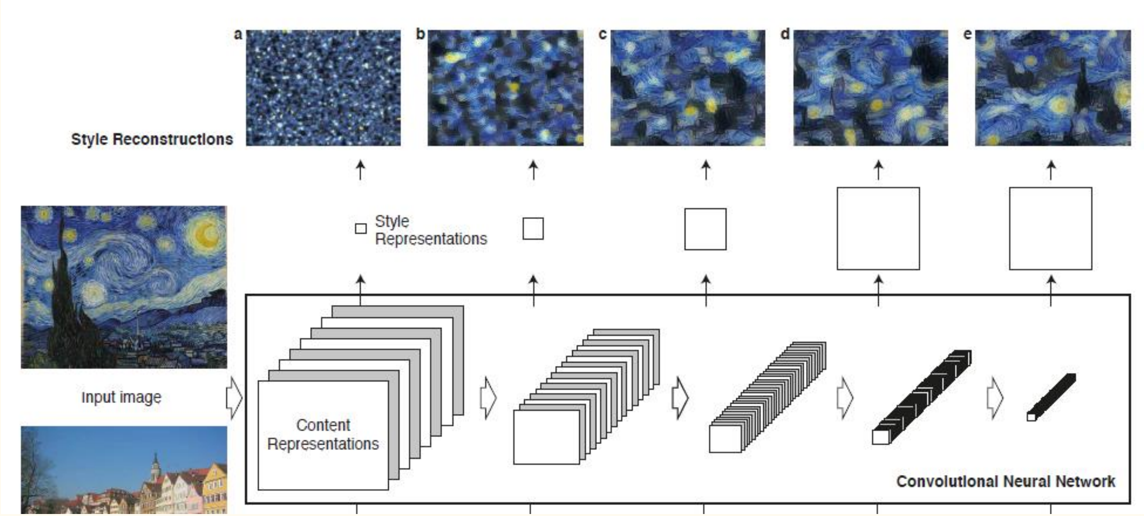
E.g. Conv5_2

Original Approach

- Key Finding - Style

→ Capture feature **correlations** of multiple layers

→ Gram matrix



$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l}.$$

G_{ij}^{l}: the inner product between the vectorized feature maps i and j in layer l.

F_{ik}^{l}: is the activation of the i^{th} filter at position k in layer l.

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G_{ij}^{l} - A_{ij}^{l}\right)^2$$

Generated image Artistic image

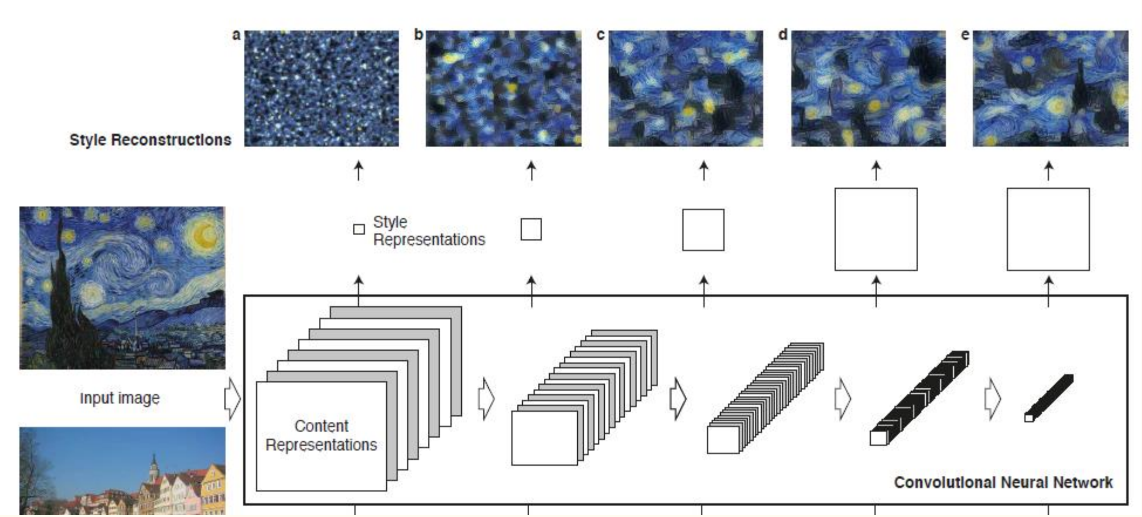
N_l: number of filters;

M_l: size of feature maps

$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l$$

Original Approach

- Key Finding - Style



Higher/More layers involved:
→ A smoother and more continuous visual experience
→ Closer to painting style

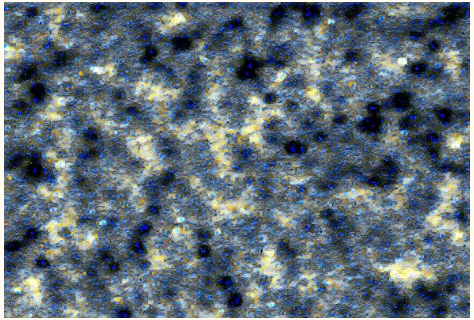
Style Representation

E.g. From Conv1_1 to Conv5_1

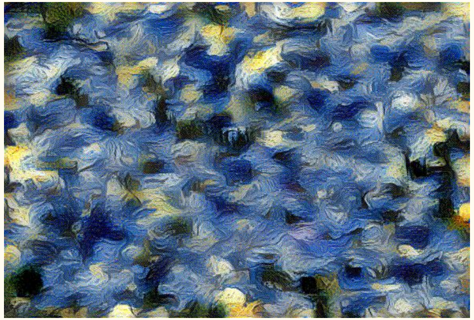
Original Approach

- Validation – Style

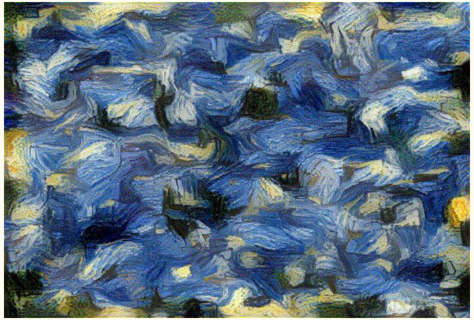
→ The trends are True!



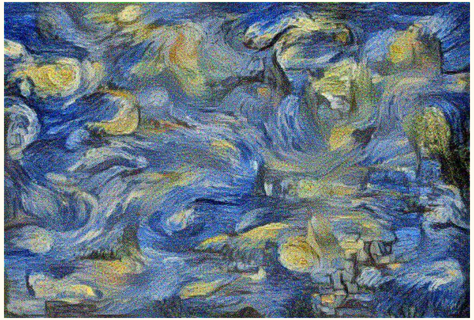
First layer



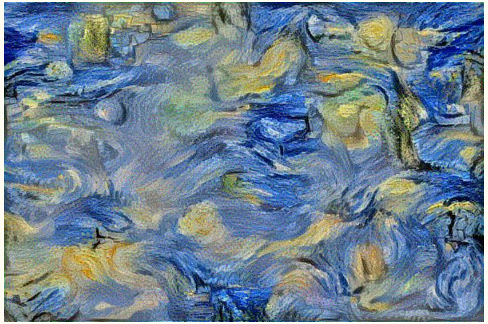
First 2 layers



First 3 layers



First 4 layers



First 5 layers

Original Approach

- The representations of **content** and **style** in the CNN are **separable**.

  Combine two loss functions!

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

➡ Combine the content & style from different images!

Application

Style Image

Content Image

$\alpha/\beta = 1$

$\alpha/\beta = 10^{-2}$

$\alpha/\beta = 10^{-1}$

$\alpha/\beta = 10^{-3}$

Application

Sensitive!

- The generated images are **not locally coherent!**

  → Add third loss function: Total variance function

$\alpha/\beta = 1$

Weight of ‘total variance function’

Imitation

After 1k iterations

Style 1

+

Content Image

=

Style 2

+

Content Image

=

Imitation

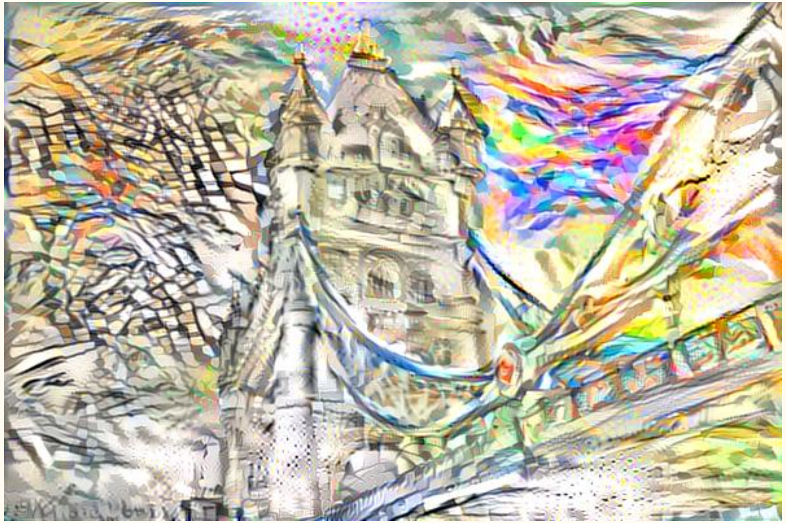
Gram matrix of style 1 image

Idea 1

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

$$G^l_{ij} = G1^l_{ij} + G2^l_{ij}$$

Works!

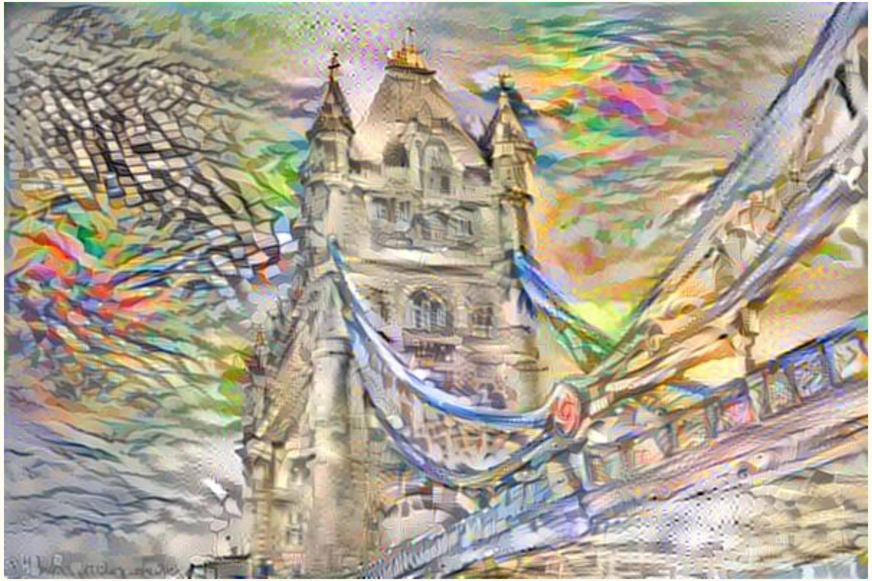


After 1k iterations

Imitation

Idea 2 $\mathcal{L}_{style}(\vec{a}, \vec{x})$ $\longrightarrow$ $\sum_i w_i \, L_{style}(\vec{a_i}, \vec{x})$

Works!



After 1k iterations; $w_1 = 0.5$

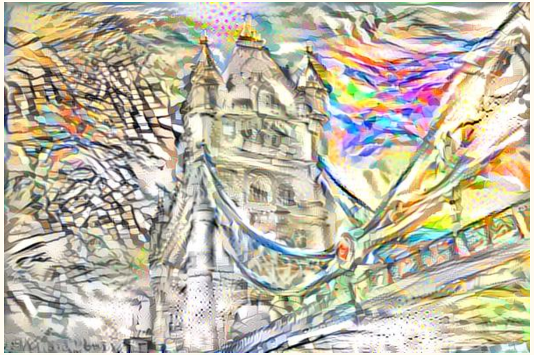
Imitation - Comparison

1k iterations

1.5k iterations

3k iterations

Change gram matrix

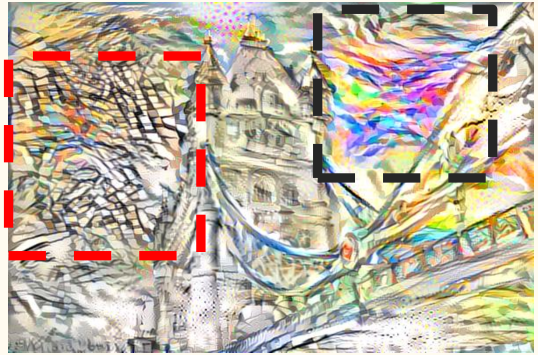


Change loss function

Imitation - Comparison

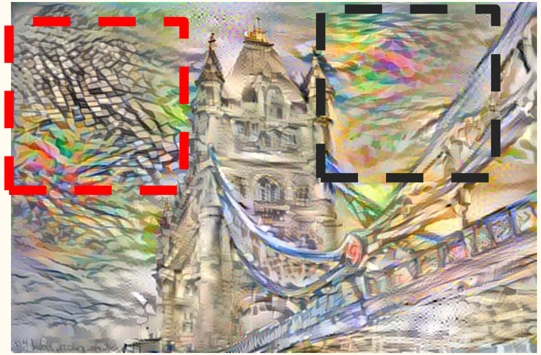
1k iterations

Change gram matrix



Change gram matrix:
1. The generated images are lighter;
2. Each style is strongly expressed

Change loss function



Change loss function :
1. darker;
2. Each style is relatively weakened

Limitation

- Cannot perform well in preserving the coherence of fine structures and details

  ← CNN features inevitably lose some low-level information

- Rely on the optimization process that is slow…

  → Original work: take up to **an hour** on a Nvidia K40 GPU[2,3]

Pre-trained models: A single style / a finite set of styles

AdaIN: Use Adaptive Instance Normalization to get the pre-trained model[4]

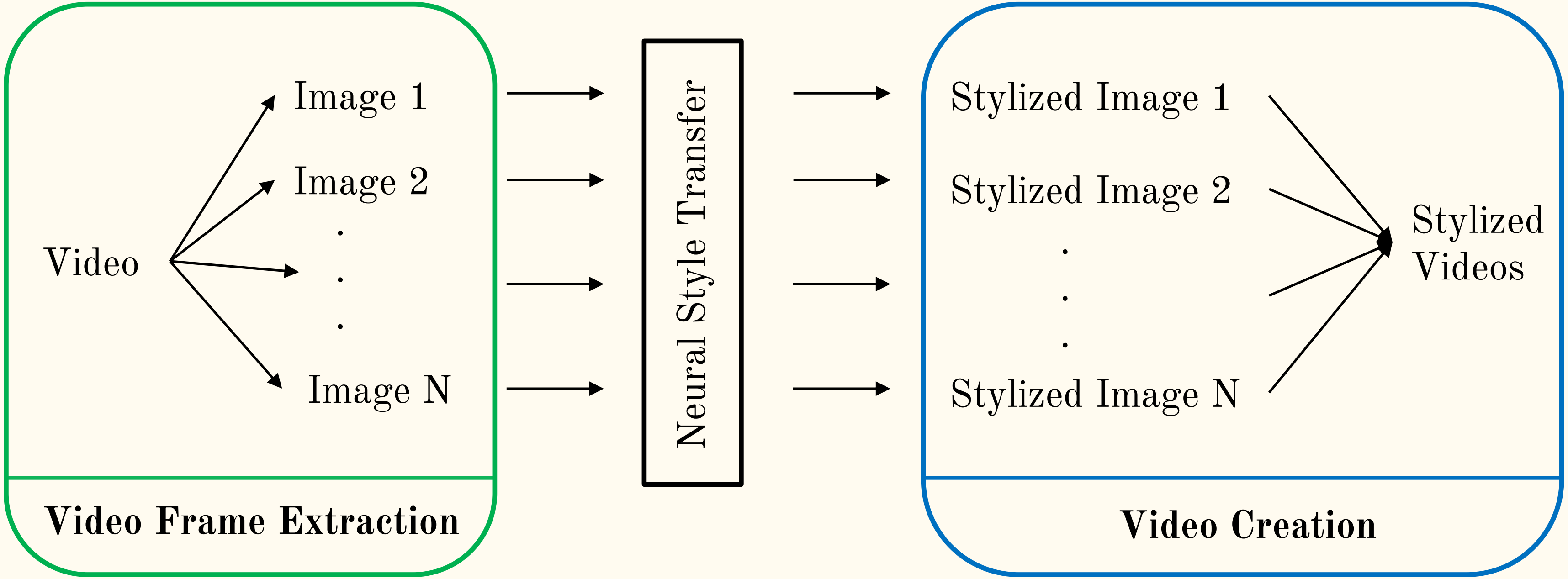
- Fail for photorealistic synthesis

  ← limitation of Gram-based style representation

Method	Time (256px)	Time (512px)	# Styles
Gatys *et al.*	14.17 (14.19)	46.75 (46.79)	∞
Ours	**0.018** (0.027)	**0.065** (0.098)	∞

Application to Video

- Idea



Application to Video

- Use AdaIN! → Fast & Reasonable Output

Style Image

Learning trajectory

Before

1. Only learned the theory of deep learning method;
2. Neural style transfer learning seems **magic** to me;
3. Extremely **little hands-on experience** with computer vision: only did classification model on digital pictures(0-9).

After

1. Carefully read at least 4 related papers;
2. Fully understand the theory behind the NST;
3. Can fine-tune the original method to satisfy my own goal;
4. Confidence gained to work on an unfamiliar topic

Resource & Reference

- Links to the starter code:
  - The seminal idea: https://github.com/jeffheaton/t81_558_deep_learning
  - Application of AdaIN: https://github.com/tg-bomze/Style-Transfer-Collection/blob/master/(Video)_pytorch_AdaIN.ipynb
- Codes contribution:
  - Adjust the style information representation and style loss measurement of the codes about the seminal idea.
- Source of style images: https://eslitegallery.viewingrooms.com/viewing-room/22/
- AI help: only used Google translator in very few parts, no other help from AI tools.

Resource & Reference

[1] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. arXiv preprint arXiv:1610.07629, 2016.
[2] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576, 2015.
[3] Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2414-2423).
[4] Huang, X., & Belongie, S. (2017). Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the IEEE international conference on computer vision (pp. 1501-1510).
[5] Li, S., Xu, X., Nie, L., & Chua, T. S. (2017, October). Laplacian-steered neural style transfer. In Proceedings of the 25th ACM international conference on Multimedia (pp. 1716-1724).
[6] Risser, E., Wilmot, P., & Barnes, C. (2017). Stable and controllable neural texture synthesis and style transfer using histogram losses. arXiv preprint arXiv:1701.08893.
[7] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
[8] Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2017). Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6924-6932).

Thanks for your attention!

Any feedback?