Artistic Neural Style Transfer

Yiheng Zhang, Yichen Li, Yuyan Wang

Stanford University

Introduction

Image style transfer is an useful approach to create a synthesized artistic styled image with an input pair consisted of a content image and a style image. The synthesized artistic styled image is expected to maintain the semantic content of the content image while displaying global and local style pattern similar to the style image.

We can utilize CNN to encode and separate the content feature and style feature of our content image, blend the original style feature with target style feature extracted from the style image, then synthesize the output by decoding them back.

Task Definition

Image style transfer task can be formulated in the following way:

- Input image pair of a content image *Ic* and a reference image *Ir*
- Trained CNN *N* will be used to produce the output synthesized artistic styled image *Io*.

The CNN model model is supposed to have an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$,which functions are defined as follows,

$$(\phi_c(I), \phi_s(I)) = \mathcal{E}(I)$$
$$I = \mathcal{D}(\phi_c(I), \phi_s(I))$$

- $\phi_c(I)$: content feature of image *I*; $\phi_s(I)$: style feature of image *I*.
- Our target is to ensure the $\phi_c(I_o)$ is close to $\phi_c(I_i)$ while $\phi_s(I_o)$ is close to $\phi_s(I_r)$. We formulate the content loss we will use for content similarity evaluation as $\mathcal{L}_c$while the style loss for style similarity evaluation as $\mathcal{L}_s$. Eventually, the combined loss function $\mathcal{L}$ for style transfer task can be defined as

$$\mathcal{L} = \alpha\mathcal{L}_c(\phi_c(I_c), \phi_c(I_o)) + \beta\mathcal{L}_s(\phi_s(I_r), \phi_s(I_o))$$

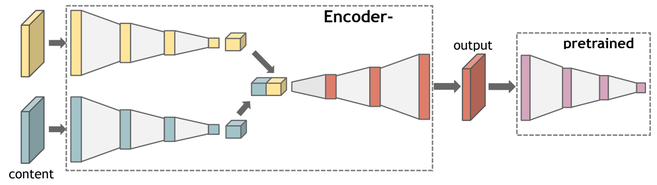
- where α and β are two hyper parameters controls the weights of $\mathcal{L}_c$ and $\mathcal{L}_s$

Our Method

Two approaches for artistic neural style transfer

First approach: we designed two encoders for extracting features from content image and style image: $\mathcal{E}_S$ and $\mathcal{E}_C$, and one decoder $\mathcal{D}$ to reconstruct the output image from extracted features.

- Use G(M) to be the gram matrix of M which measures texture continuity of two features.
- ImageNet pertained VGG-19 to supervise whether *Io* is close to *Ic*. We used concatenation to combine style and content features.



$$\mathcal{L}_C = \|\mathbf{G}(\mathcal{E}_C(I_c)) - \mathbf{G}(\mathcal{E}_C(I_o))\|_2 \quad \mathcal{L}_S = \|\mathbf{G}(\mathcal{E}_S(I_s)) - \mathbf{G}(\mathcal{E}_S(I_o))\|_2$$
$$\mathcal{L}_B = \|\mathbf{V}(I_c)) - \mathbf{V}(I_o))\|_2$$
$$\mathcal{L} = \alpha\mathcal{L}_S + \beta\mathcal{L}_C + (1 - \alpha - \beta)\mathcal{L}_B$$

- α denotes the dedicated style loss weight, and β denotes the dedicated content loss weight correspondingly.
- Constraints: $0 \le \alpha \le 1$, $0 \le \beta \le 1 - \alpha$

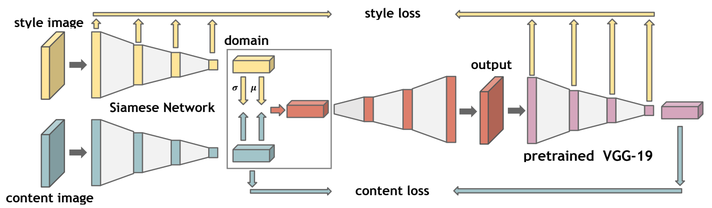
Second approach: similarly we use encoders $\mathcal{E}_S$ and$\mathcal{E}_C$, decode $\mathcal{D}$
Our style loss has two terms:

Normalization: $\mathcal{L}_S^N = \sum_{i=1}^{L}\|\mu(\mathcal{E}_C^i(I_o)) - \mu(\mathcal{E}_S^i(I_s))\|_2 + \sum_{i=1}^{L}\|\sigma(\mathcal{E}_C^i(I_o)) - \sigma(\mathcal{E}_S^i(I_s))\|_2$

Gram matrix: $\mathcal{L}_S^G = \|(Gram(I_o)) - (Gram(I_s))\|_2$

Content loss: $\mathcal{L}_c = \|(\mathcal{E}_C(I_c)) - (\mathcal{E}_C(I_o))\|_2$

Loss summary: $\mathcal{L} = \alpha(\mathcal{L}_S^N + \mathcal{L}_S^G) + (1 - \alpha)\mathcal{L}_C$



- **Domain adjustment**: apply the affine transformation of the style input s to the content image c and compute the layer-wise mean μ_s
- Especially normalize the content image extract: its relative position with respect to the center of its Gaussian distribution, then apply the extracted σ_s and apply a shift of μ_s

$$\text{Norm}(s, c) = \sigma_s N_c + \mu_s \quad N_c = \left(\frac{c - \mu_c}{\sigma_c}\right)$$

Experiment

- We use MSCOCO and WikiArt dataset as content image and style image dataset respectively
- We use 0.2 as content loss weight and 0.8 as style loss weight. The learning rate is 0.001
- The training process takes 120 hours to iterate through 23674 content images and 21 style images for 10 epochs to finish on Google Cloud server with Tesla K80 GPU

Result

Visual Analysis:

- **Baseline**: do basic tone mapping in global pattern and apply some general strokes but fail in color/texture mapping in local patches
- **Our Method**: consider a more fine-grained style application to an input image, with more drastic alignment, better tone mapping
- **Oracle**: do a better job at separating the foreground and background of the content image, apply a more detailed style transfer semantically

From top to bottom: style image, content image, baseline, our method, oracle

User Study:

- Collect the ranking of output images among 5 students (blind test)
- Average ranking shows that our method is considered to surpass other two in terms of style similarity and aesthetics

Runtime Comparison:

- Baseline: take relatively long to finish (can't run in realtime)
- Our Method: fastest style transfer (can run in realtime)
- Oracle: a bit slower than our method (can run in realtime)

Method	Time (256 px)	Time (512 px)
Baseline	16.941s	61.424s
Ours	**0.012s**	**0.041s**
Oracle	0.019s	0.060s

Method	Fast (1s)	Slow (10 s)
Baseline	2.612	2.680
Ours	**1.545**	**1.527**
Oracle	1.843	1.793