Neural Style Transfer Notes

November 28th, 2017

Tammy Qiu, Rex Wang

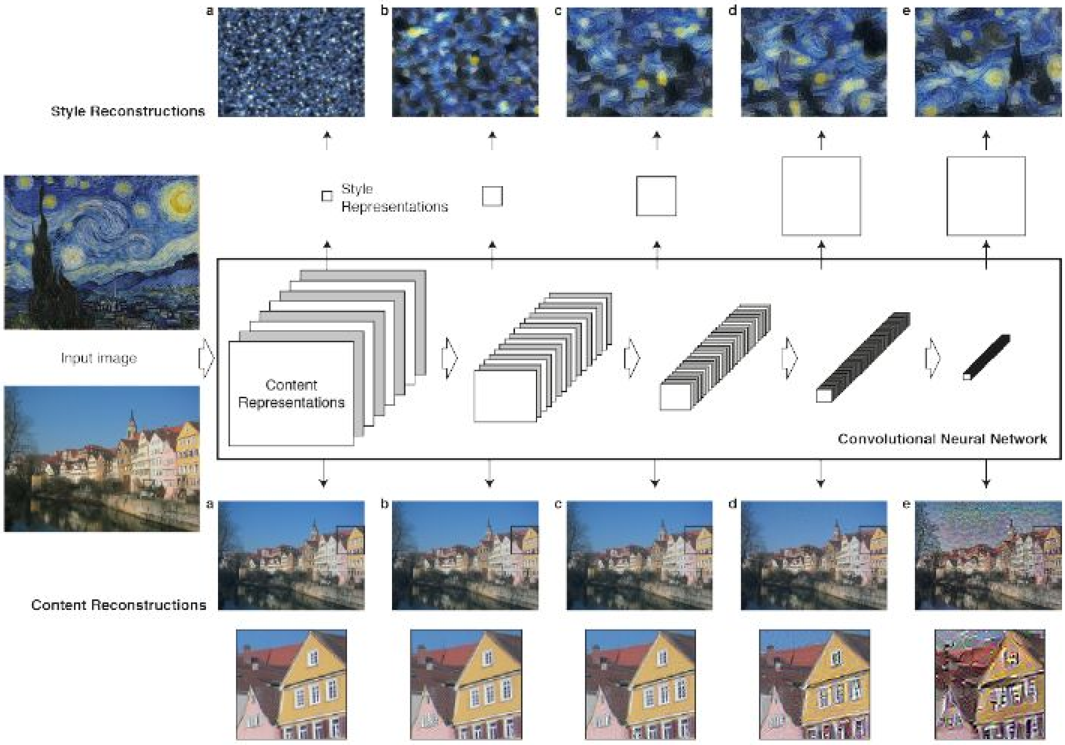
1 Introduction and Motivation

1. Artificial system based on a Deep Neural Network that creates artistic images of high perceptual quality
2. Uses neural representations to separate and recombine content and style of arbitrary images
3. Offers path forward to algorithmic understanding of how humans create and perceive artistic imagery

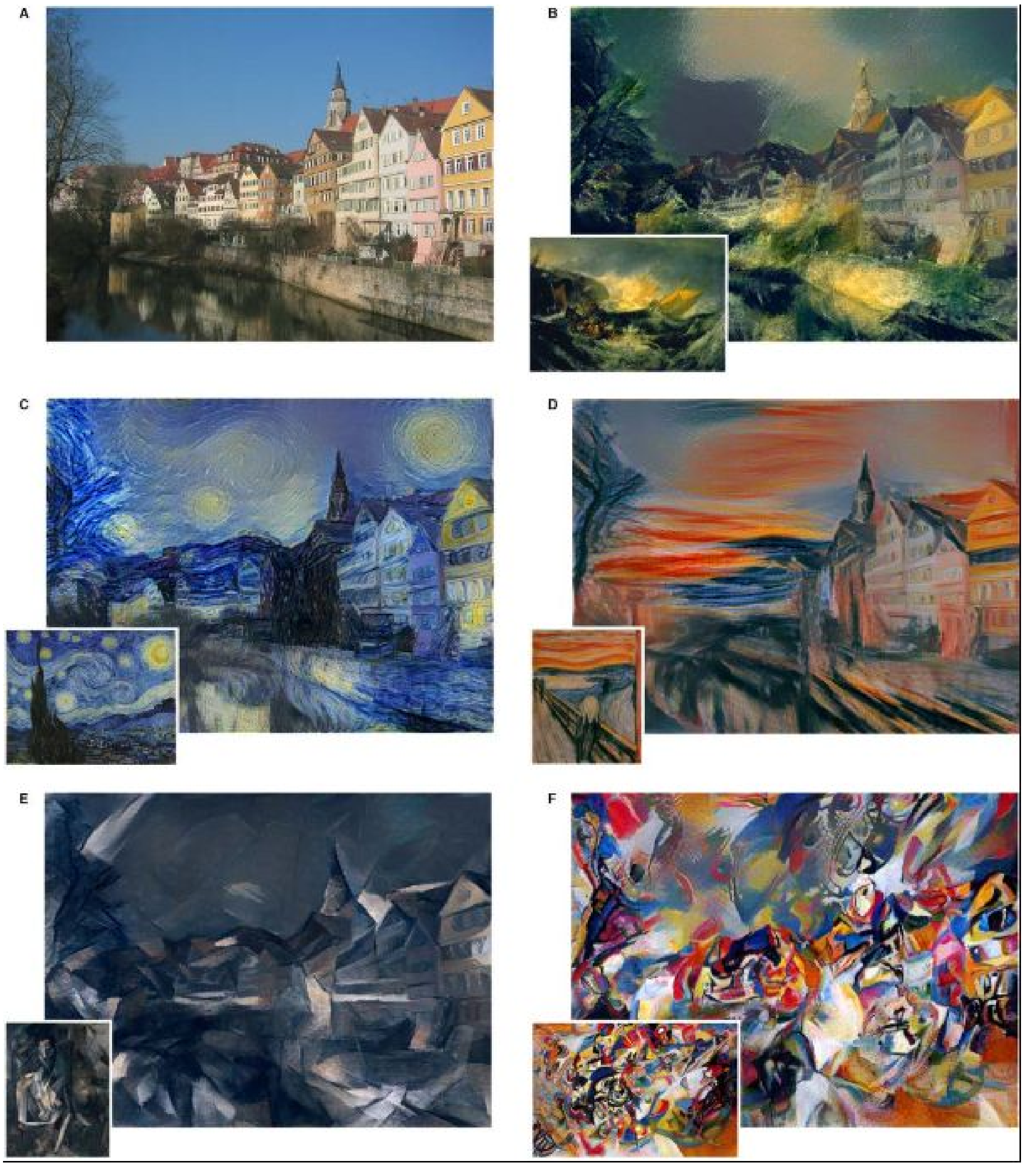
2 Convolutional Neural Networks

1. Most powerful in image processsing tasks
2. Consist of layers of small computational units
    a. Process visual information hierarchically
    b. Feedforward manner
3. Each layer: collection of image **filters**
    a. Each filter extracts a certain feature from the input image
4. **Feature maps**: output of a given layer
    a. Differently filtered versions of the input image
5. When trained on object recognition, CNN develops a representation of the image that makes object information increasingly explicit along processing hierarchy
    a. Representations that care about the content of the image vs. detailed pixel values
    b. Visualize information from each layer about input image by reconstructing image from feature maps in that layer--Content Reconstruction
        i. **Content representation**: Higher layers in network = higher level content in images
        ii. Lower layers: reproduce exact pixel values of input
6. **Style representation**
    a. Use feature space originally designed to capture *texture* information

        i. Built on top of filter responses in each layer
        ii. Correlations between different filter responses over the spatial extent of the feature maps
        iii. Captures texture information but not global arrangement
    b. Visualize information of style feature spaces by constructing an image that matches the style representation (style reconstruction)
        i. Reconstructions produce texturized versions of input capturing general appearance
            1. Color
            2. Localized structures
            3. Size and complexity of local image structures increases along hierarchy



7.
    a. Input image represented as a set of filtered images at each processing stage
    b. Number of different filters increases along processing hierarchy
        i. Size of filtered images reduced by downsampling mechanism (.e. max-pooling)
        ii. Decrease in total number of units per layer of the network
    c. **Content Reconstructions**: reconstruct input image from the network's responses in a particular layer
        i. Reconstruction at lower layers is almost perfect (a, b, c)
        ii. Reconstruction at higher layers preserves high-level conent bu loses detailed pixel information (d, e)

8. **Key finding**: Representations of content and style in CNNs are separable
    a. Can manipulate both representations independently to produce new images
    b. Generate images that mix the content and style representation from two sources (Composite image)
        i. Global arrangement of original is preserved
        ii. Colors and local structures that compose global scenery are provided by artwork
    c. 

    d. **Loss function** contains 2 terms for *content* and *style* that are well separated
        i. Can smoothly regulate the emphasis on either reconstructing the content or the style
            1. Tradeoff between more style (less content) or more content (less style)

3 Descriptive Neural Methods Based on Image Iteration

1. Method proposed by Gatys, et al.

2. Transfers style by directly updating pixels in the image *iteratively* through **backpropagation**
3. Objective: minimize total loss such that the stylized image simultaneously matches the content representation of the content image and the style representation of the style image
    a. Passed with content + style into several layers of a network pretrained on image classification
    b. Use outputs of various intermediate layers to compute 2 types of losses
        i. Style loss: ho close pastiche is to style image in style
        ii. Content loss: how close pastiche is to content image in content
    c. Losses minimized by directly changing pastiche image
    d. By the end: pastiche has style of style image, content of content image
        i. Stylized version of original content version
4. Generated on the basis of the VGG Network
    a. CNN that rivals human performance on object recognition
    b. 16 convoltional layers
    c. 5 pooling layers
    d. No fully-connected layers
    e. Average pooling improves gradient flow and obtains more appealing results than max pooling
    f. Use a pretrained network
        i. For a network to do image classification, it has to understand the image
        ii. Doing transformations to turn the image pixels into an internal understanding of the content of the image
        iii. Internal understanding = intermediate semantic representations of the initial image
5. 3 different types of images in Neural Style Transfer
    a. **Pastiche**: stylized image we wish to achieve
    b. **Style image**: input artwork whose style we want to transfer
    c. **Content image**: picture we want to transfer style onto
6. This method is called the **Maximum Mean Discrepancy (MMD)**
    a. MMD: popular metric of discrepancy between 2 distributions
    b. Style transfer can be considered a distribution alignment process from content-->style
    c. Using image reconstruction strategy and texture synthesis algorithm
        i. Overall procedure of iterating newly stylized images using gradient descent-->image reconstruction
        ii. Style matching-->texture synthesis
7. Each layer defines a non-linear filter bank
    a. Complexity increases with position of layer in network
    b. Input image: $\vec{x}$ is encoded in each layer of the CNN by the filter responses to that image
    c. A layer with N_l distinct filters
        i. Has N_l feature maps each of size M_l

ii. M_l = height x width of feature map

d. Responses in layer $l = F^l \in R^{N_l \times M_l}$

i. F^l_{ij} is the activation of the i^{th} filter at position j in layer l

8. To visualize image information encoded at each layer (content)

a. Perform gradient descent on a white noise image to find another image that matches the feature responses of the original image

b. Let $\overrightarrow{p}$ be the original image

i. P^l: feature representation in layer l

ii. $\overrightarrow{x}$: image that is generated

1. F^l: its feature representation in layer l

iii. Squared error loss between the two feature representations:

1. $$L_{content}(\overrightarrow{p}, \overrightarrow{x}, l) = \frac{1}{2}\sum_{ij}(F^l_{ij} - P^l_{ij})^2$$

iv. Derivative of Loss with respect to activations in layer l:

1. $$\frac{\partial \mathcal{L}_{content}}{\partial F^l_{ij}} = \begin{cases} \left(F^l - P^l\right)_{ij} & \text{if } F^l_{ij} > 0 \\ 0 & \text{if } F^l_{ij} < 0 . \end{cases}$$

9. To generate texture that matches the style of a given image

a. Style representation: Gram matrix $G^l \in R^{N_l \times N_l}$

i. Gives feature correlations

ii. G^l_{ij} is the inner product between the vectorised feature map i and j in layer l.

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

iv. Every column is multiplied with every row in the matrix-->spatial info contained in original representations have been *distributed*

v. Contains non-localized information

vi. Contains information about style

1. Textures
2. Shapes
3. Weights

b. Use gradient descent from a white noise image to find another image that matches the style representation of the original image

c. Minimize mean-squared distance between entries of Gram matrix from style image and Gram matrix of pastiche

i. $\overrightarrow{a}$: original image

1. A^l: style representation in layer l

ii. $\overrightarrow{x}$: image that is generated

1. G^l: style representation in layer l

iii. Contribution of that layer to the total loss

1. $$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G^l_{ij} - A^l_{ij})^2$$

iv. Total loss:

1. $$L_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l$$

2. w_l: weighting factors of the contribution of each layer to the total loss

v. Derivative of E_l w.r.t. activations in layer l can be computed analytically

vi. $$\frac{\partial E_l}{\partial F^l_{ij}} = \begin{cases} \frac{1}{N_l^2 M_l^2} \left((F^l)^{\mathrm{T}} \left(G^l - A^l \right) \right)_{ji} & \text{if } F^l_{ij} > 0 \\ 0 & \text{if } F^l_{ij} < 0 \,. \end{cases}$$

10. To generate images that mix content and style

a. Jointly minimize distance of a white noise image from the content representation of the photograph in one layer of the network

b. and the style representation of the painting in a number of layers of the CNN

i. Layer for content representation (high or low in hierarchy) determines how much of content will remain

ii. Loss function

1. $L_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha L_{content}(\vec{p}, \vec{x}) + \beta L_{style}(\vec{a}, \vec{x})$

2. α, β: weighting factors for content and style reconstruction respectively

11. Other networks that are trained to perform object recognition tasks are also capable of achieving similar performance

a. E.g. Resnet

12. Limitations

a. Instabilities during iterations

b. Requires manually tuning parameters

4 Generative Neural Methods based on Model Iteration

1. Use of perceptual loss functions for training feed-forward networks for image transformation tasks
2. Gives similar qualitative results to method above but is 3 orders of magnitude faster
3. Train feedforward transformation networks for image transformation tasks

a. Train using perceptual loss functions (depend on high-level features from a pretrained loss network)

b. Perceptual losses measure image similarities more robustly than per-pixel losses

4. Method:

a. Two components

        i. **Image transformation network** f_W
            1. Deep residual CNN parameterized by weights W
            2. Transforms input images x into output images $\hat{y}$ via mapping $\hat{y} = f_W(x)$
            3. Trained using SGD to minimize a weighted combination of loss functions:
                a. $$W := argmin_W E_x, y_i[\sum_{i=1} \lambda_i l_i(f_W(x), y_i)]$$
        ii. **Loss network** ϕ
            1. Used to define several loss functions $l_1, \ldots, l_k$
            2. Each loss function computes a scalar value $l_i(\hat{y}y_i)$ -->measuring the differece between the output image $\hat{y}$ and a target image y_i

    b. Loss network defines a feature construction loss l^{ϕ}_{feat} and a style reconstruction loss l^{ϕ}_{style} that measure differences in content and style
        i. Each input image x:
            1. Content target y_c
                a. Input image x
                b. Output image $\hat{y}$ should combine the content $x = y_c$ with the style of y_s
            2. Style target y_s
                a. Train one network per style target

    c. Image Transformation Networks
        i. No pooling layers
            1. Using strided and fractionally strided convolutions for in-network downsampling and upsampling
        ii. Network body = 5 residual blocks using architecture of ResNets
            1. All non-residual convolutional layers are followed by spatial batch normalization and ReLu nonlinearities
                a. Exception: output layer--scaled tanh to ensure that output image has pixels in range [0, 255]
            2. Other than first and last layers which use 9x9 kernels
                a. All convolutional layers use 3x3 kernels
            3. Inputs and Outputs
                a. Both color images of shape 3 x 256 x 256
            4. Downsampling and upsampling
                a. Networks use stride-2 convolutions to downsample input
                b. Followed by several residual blocks
                c. 2 convolutional layers with stride ½ to upsample
                d. Benefits of downsampling then upsampling
                    i. Computational

1. Naive implementation: 3 x 3 convolution with C filters on an input of size $C \times H \times W$ requires $9HWC^2$ multiply-adds
2. Same cost as 3 x 3 convolution with DC filters on an input of shape $DC \times H/D \times W/D$
3. After downsampling-->can use a larger network for the same computational cost

ii. Effective receptive field sizes

1. High quality style transfer: changing large parts of the image in a coherent way
2. Advantageous for each pixel in output to have a large effective receptive field
3. Without downsampling: each additional 3 x 3 convolutional layer increases effective receptive field size by 2
4. After downsampling by a factor of D: each 3 x 3 convolution increases size by 2D-->larger effective receptive fields with the same number of layers

5. Residual connections
    a. Use residual connections to train very deep networks for image classifications
    b. Residual connections make it easy for the network to learn the identity function
    c. Output image should share structure with the input image

d. Perceptual Loss Functions

i. 2 perceptual loss functions that measure high-level perceptual and semantic differences between images

ii. Make use of a loss network ϕ pre-trained for image classification

1. Perceptual loss functions are themselves deep convolutional neural networks
2. ϕ: 16-layer VGG network pretrained on ImageNet

iii. Content reconstruction loss

1. Encourage similar content representations > similar per-pixel
2. $\phi_j(x)$: activations of jth layer of network ϕ when processing image x
3. If j is a convolutional layer
    a. $\phi_j(x)$ is a feature map of shape $C_j \times H_j \times W_j$
4. Content reconstruction loss: (squared, normalized) Euclidean distance between content representations

        a. $l_{feat}^{\phi,j}(\hat{y}, y) = \frac{1}{C_jH_jW_j}||\phi_j(\hat{y} - \phi_j(y)||_2^2$

    5. Minimizing content reconstruction loss for early layers tends to produce images that are virtually indistinguishable from y
        a. Perceptually similar
    6. Reconstruct from higher layers:
        a. Image content and overall spatial structure preserved
        b. Color, texture and exact shape are not

iv. Style Reconstruction Loss

    1. Content reconstruction loss penalizes when output deviates in content from target
    2. Penalize differences in style: colors, textrures, common patterns
    3. $\phi_j(x)$: activations at the jth layer of network ϕ for x
        a. A feature map of shape $C_j \times H_j \times W_j$
    4. **Gram Matrix** $G_j^\phi(x)$ (same as the one in the previous method)
        a. $C_j \times C_j$ matrix whose elements are
        b. $G_j^\phi(x)_{c,c'} = \frac{1}{C_jH_jW_j}\sum_{h=1}^{H_j}\sum_{w=1}^{W_j}\phi_j(x)_{h,w,c}\phi_j(x)_{h,w,c'}$
        c. $\phi_j(x)$: giving C_j-dimensional features for each point on a $H_j \times W_j$ grid
        d. $G_j^\phi(x)$: proportional to the uncentered covariance of the C_j-dimensional features
            i. Treating each grid location as an independent sample
            ii. Captures information about which features tend to activate together
        e. Can be computed efficiently
            i. Reshape $\phi_j(x)$ into a matrix ψ of shape $C_j \times H_j \times W_j$ and $G_j^\phi(x) = \frac{\psi\psi^T}{C_jH_jW_j}$
    5. **Style reconstruction loss:** squared Frobenius norm of difference between Gram matrices of output and target images
        a. $l_{style}^{\phi,j}(\hat{y}, y) = ||G_j^\phi(\hat{y} - G_j^\phi(y)||_F^2$
        b. Well defined even when $\hat{y}$ and y have different sizes, since their Gram matrices will both have the same shape
    6. Minimize style reconstruction loss
        a. Preserves stylistic features from target image but not its spatial structure

b. Reconstructing from higher layers transfers large-scale structure from the target image

7. To perform style reconstruction from a set of layers J:

a. $l_{style}^{\phi,J}(\hat{y},y)$: sum of losses for each layer $j \in J$

5. Experiments

a. Baseline: Gatys' method above

i. Solving $\hat{y} = argmin_y \lambda_c l_{feat}^{\phi,h}(y,y_c) + \lambda_s l_{style}^{\phi,J}(y,y_s) + \lambda_{TV} l_{TV}(y)$

ii. y is initialized with white noise

iii. Optimization using L-BFGS

iv. Converges to satisfactory results within 500 iterations

1. Slow because L-BFGS requires forward and backward pass

b. Training Details

i. Microsoft COCO dataset

ii. Resize each of the 80k training images to 256 x 256

iii. Ran networks with a batch size of 4 for 40,000 iterations

iv. Roughly 2 epochs over the training data

v. Use Adam with a learning rate of 1×10^{-3}

vi. Output images regularized with total variation regularization with a strength of between 1×10^{-6} and 1×10^{-4}

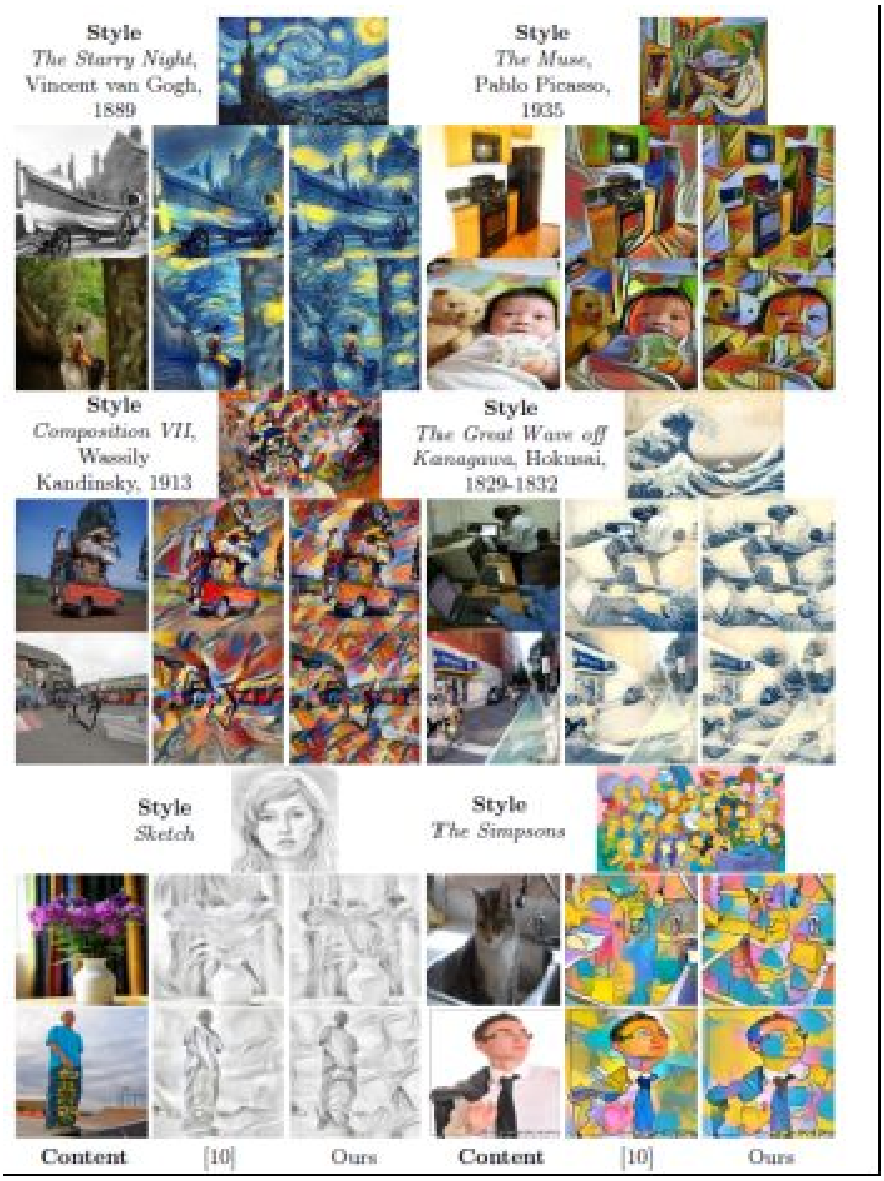
1. Chosen via cross-validation per style targt

vii. No weight dcay or dropout

1. Model doesn't overfit within 2 epochs

c. Qualitative Results

i. Qualitatively similar to the baseline ii.


Style
The Starry Night, Vincent van Gogh, 1889
Style
The Muse, Pablo Picasso, 1935
Style
Composition VII, Wassily Kandinsky, 1913
Style
The Great Wave off Kanagawa, Hokusai, 1829-1832
Style
Sketch
Style
The Simpsons
Content
[10]
Ours
Content
[10]
Ours

d. Quantitative results

   i. Both equations minimize:

$$\hat{y} = argmin_y \lambda_c l^{\phi,h}_{feat}(y, y_c) + \lambda_s l^{\phi,J}_{style}(y, y_s) + \lambda_{TV} l_{TV}(y)$$

   ii. Baseline: performs explicit optimization over the output

   iii. This method: trained to find a solution for any content image y_c in a single forward pass

   iv. Run method and the baseline on 50 images from MS-COCO (using *The Muse* by Picasso)

      1. Achieved loss comparable in 50-100 iterations

5 References

1. Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." arXiv preprint arXiv:1508.06576 (2015).
2. Jing, Yongcheng, et al. "Neural style transfer: A review." arXiv preprint arXiv:1705.04058 (2017).

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