Artistic Style-Transfer

Elias Wang (elias.wang@stanford.edu)
Nicholas Tan (tan.nicholas@yahoo.com)

Proposal:

Figure 1. (Leon Gatys)

Description:

It has been shown that a convolutional neural net algorithm can distinguish between the content of a piece of art and the style that it was painted in ([1] Leon Gatys). For our final project, we would like to explore this idea in the context of image processing. Specifically, we want to create an image filter that transfers the style of one painting to the content of another image. A clear motivation for finding a solution is for applications that handle many images at a time or for applications that cannot tolerate the training/learning time overhead that comes with the CNN ([2] Michael Elad).

Proposed Methods:

0. Define style transfer from style of image 2 to content of image 1
1. Extract the “content” from image 1
    a. This may be achieved by feature detection (edges, corners, lines, circles, uniform areas, etc)

    b. Rank which features are more important to the scene overall - some metrics that may be considered are nominal size of the feature, centricity, brightness, color, contrast, etc
2. Extract the “content” from image 2
3. Extract the “style” from image 2
    a. This may be accomplished by segmenting the image and trying to find key similarities between different patches i.e. themes throughout the image
    b. Map styles to features
4. Find sufficiently matched areas of “content” between image 1 and 2 and apply the appropriate “style” that fits the content given : its ranked importance in the scene, how similar it is to the “content” of image 2,
5. Fill in empty spaces and stitch / blend the various areas of the image

Overview of style-transfer algorithm from [2]:

Objective: Create the style-transfer image, $\underline{X}$.

Input: Use the following ingredients and parameters:

- $\underline{C}$ and $\underline{S}$ - Content and Style images
- $\mathbf{W}$ - Content segmentation map
- L_{max} - Number of resolution layers
- $n_1, n_2, \ \dots \ , n_m$ - patch sizes
- $d_1, d_2, \ \dots \ , d_m$ - Ω subsampling gaps
- I_{IRLS} - number of IRLS iterations (chosen as 10)
- I_{alg} - number of update iterations per patch-size
- r - robust statistics value to use.

Initialization:

- Apply color transfer from $\underline{S}$ to $\underline{C}$
- Build the Gaussian pyramids of $\underline{C}$, $\underline{S}$, and $\mathbf{W}$
- Optional (for fast and approximate NN): Prepare the patches from the style image for each resolution layer and each patch size as tree structure
- Initialize $\underline{X} = \mathbf{D}_L^C \underline{C} + V$, where $V \sim \mathcal{N}(0, 50)$

Loop Over Scales: For $L = L_{max}, \ \dots \ , 1$ do:

Loop Over Patch-Sizes: For $n = n_1, \ \dots \ , n_m$, minimize $E_{L,n}$ w.r.t. $\underline{X}$, by

Iterate: For $k = 1, 2 \ \dots \ , I_{alg}$ do:

1. *Patch Matching*: For the current image $\underline{X}$, match NN from the style image by solving (16). The style patches should be of the same size, taken from the corresponding resolution layer.
2. *Robust Aggregation*: Compute $\tilde{\underline{X}}$ as the robust patch aggregation result, by solving (14) using I_{IRLS} iterations.
3. *Content Fusion*: Combine the content image $\mathbf{D}_L^C \underline{C}$ to $\tilde{\underline{X}}$ using Equation (15) to obtain the updated $\underline{X}$.
4. *Color Transfer*: Apply color-transfer from $\underline{S}$ to $\underline{X}$.
5. *Denoise*: Filter the obtained $\underline{X}$ by the domain-transform.

Scale-Up: As we move from one resolution layer to the next, scale-up $\underline{X}$.

Result: The output of the above algorithm is $\underline{X}$.

Figure 3: The overall Style-Transfer Algorithm.

Overview of style-transfer algorithm from [3]:

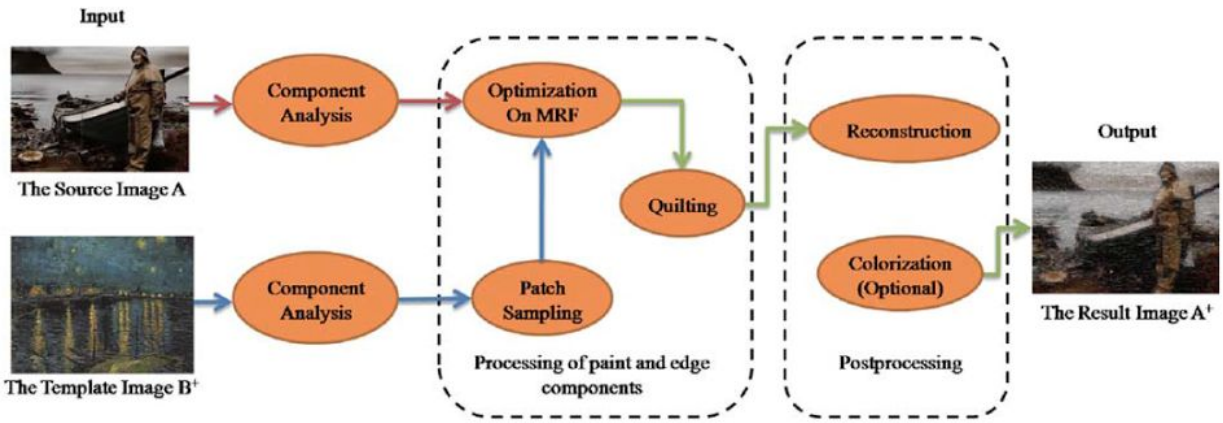

Fig. 2. The framework of image style transfer.

We do not plan to use an Android Device

References:

[1] Leon A. Gatys, Alexander S. Ecker,Matthias Bethge. “A Neural Algorithm of Artistic Style.” https://arxiv.org/pdf/1508.06576.pdf

[2] Michael Elad and Peyman Milanfar “Style-Transfer via Texture-Synthesis” arXiv:1609.03057

[3] Wei Zhang, Chen Cao, Shifeng Chen, Jianzhuang Liu, Senior Member, IEEE, and Xiaoou Tang, Fellow, IEEE “Style Transfer Via Image Component Analysis" ieeexplore.ieee.org/iel7/6046/6630084/06522845.pdf

[4] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, “Texture Optimization for Example-Based Synthesis”, ACM ToG, Vol. 24, No. 3, pp. 795-802, 2005.