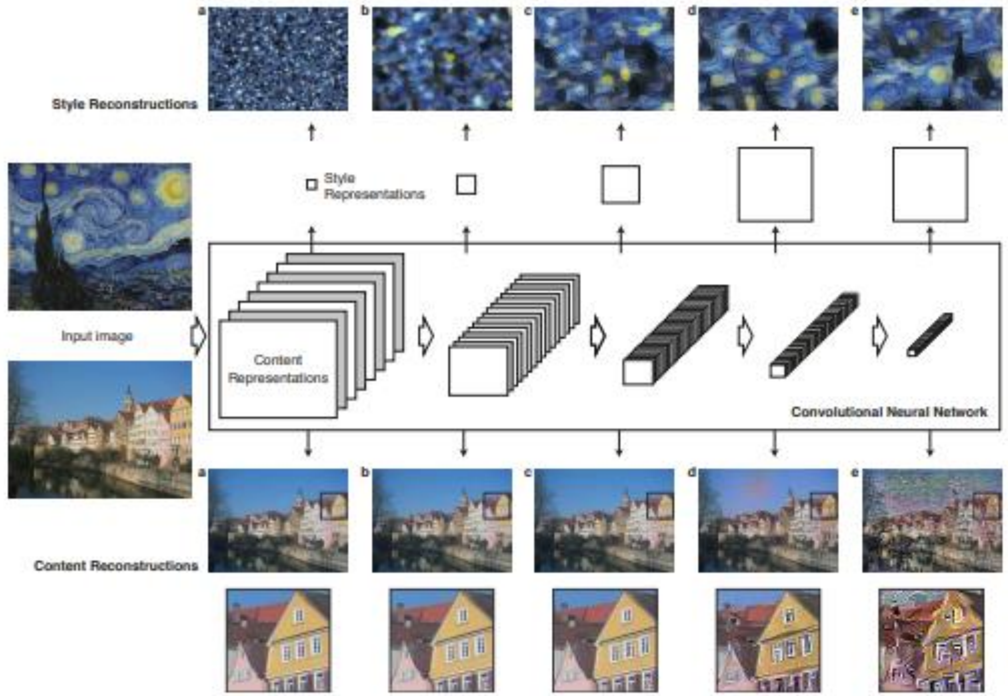
Style Transfer

By Wei Gao

Optimization Method

Image Style Transfer Using Convolutional Neural Networks



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

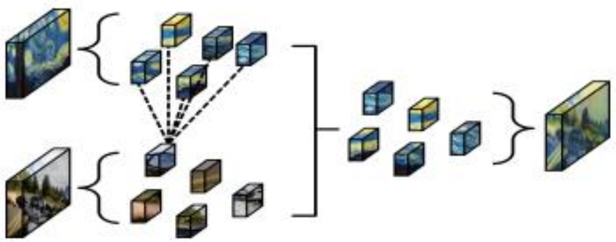
$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G_{ij}^{l} - A_{ij}^{l}\right)^2$$

Swap Method

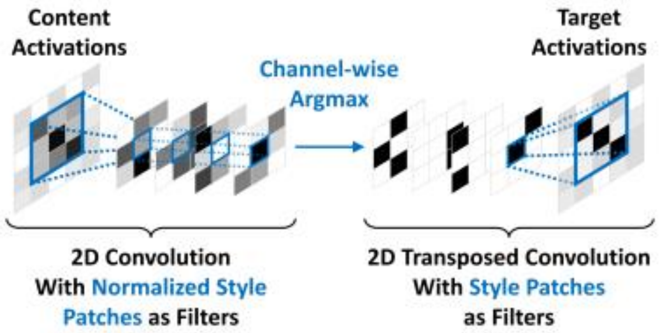
Fast Patch-based Style Transfer of Arbitrary Style

For every content patch, swap it with the best matching style patch, which we define using the normalized cross-correlation:

$$\mathrm{BestMatch}(c) = \arg\max_{s \in S} \frac{< c, s >}{||c|| \cdot ||s||}$$



This operation can be implemented efficiently using a 2D convolutional layer and a 2D transposed convolutional layer.



Learning Method

Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization

eplacing BN layers with IN layers:

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

$$\mathrm{AdaIN}(x, y) = \sigma(y) \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y) \quad (8)$$

$$\mathrm{CIN}(x; s) = \gamma^s \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \beta^s \quad (7)$$

Learning Method

Perceptual Losses for Real-Time Style Transfer and Super-Resolution

$$\ell_{feat}^{\phi,j}(\hat{y},y) = \frac{1}{C_j H_j W_j} \|\phi_j(\hat{y}) - \phi_j(y)\|_2^2$$

$$G_j^{\phi}(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'}.$$

Zero-Shot Method

Universal Style Transfer via Feature Transforms

Whitening transform. Before whitening, we first center f_c by subtracting its mean vector m_c. Then we transform f_c linearly as in (2) so that we obtain $\hat{f}_c$ such that the feature maps are uncorrelated ($\hat{f}_c \hat{f}_c^\top = I$),

$$\hat{f}_c = E_c \, D_c^{-\frac{1}{2}} \, E_c^\top \, f_c \,, \tag{2}$$

where D_c is a diagonal matrix with the eigenvalues of the covariance matrix $f_c \, f_c^\top \in \Re^{C \times C}$, and E_c is the corresponding orthogonal matrix of eigenvectors, satisfying $f_c \, f_c^\top = E_c D_c E_c^\top$.

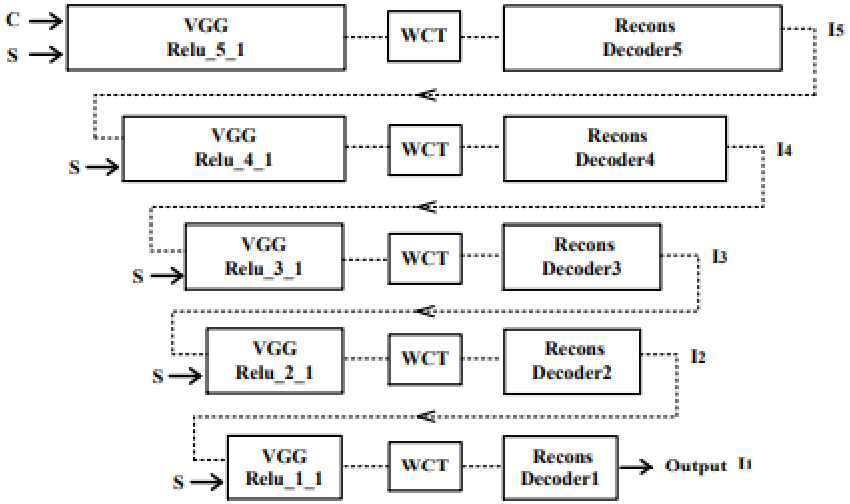
Coloring transform. We first center f_s by subtracting its mean vector m_s, and then carry out the coloring transform [14], which is essentially the inverse of the whitening step to transform $\hat{f}_c$ linearly as in (3) such that we obtain $\hat{f}_{cs}$ which has the desired correlations between its feature maps ($\hat{f}_{cs} \, \hat{f}_{cs}^\top = f_s \, f_s^\top$),

$$\hat{f}_{cs} = E_s \, D_s^{\frac{1}{2}} \, E_s^\top \, \hat{f}_c \,, \tag{3}$$

where D_s is a diagonal matrix with the eigenvalues of the covariance matrix $f_s \, f_s^\top \in \Re^{C \times C}$, and E_s is the corresponding orthogonal matrix of eigenvectors. Finally we re-center the $\hat{f}_{cs}$ with the mean vector m_s of the style, i.e., $\hat{f}_{cs} = \hat{f}_{cs} + m_s$.

Zero-Shot Method

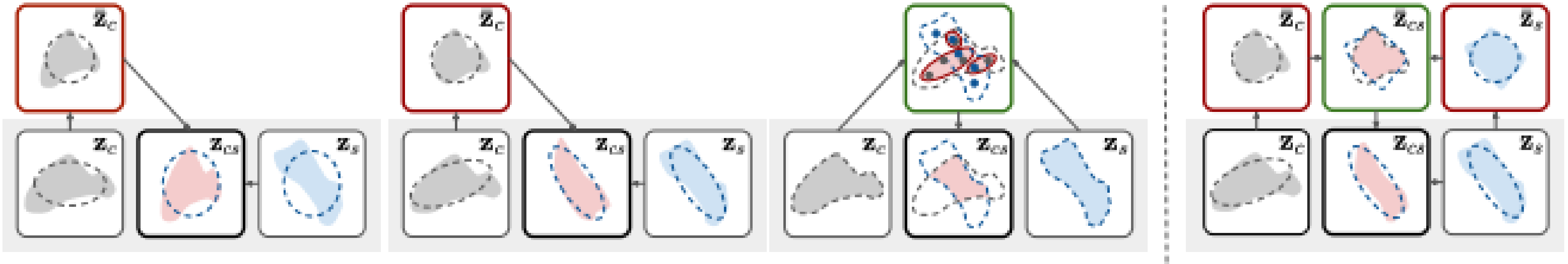
Universal Style Transfer via Feature Transforms



(a) Reconstruction (b) Single-level stylization (c) Multi-level stylization

Zero-Shot Method

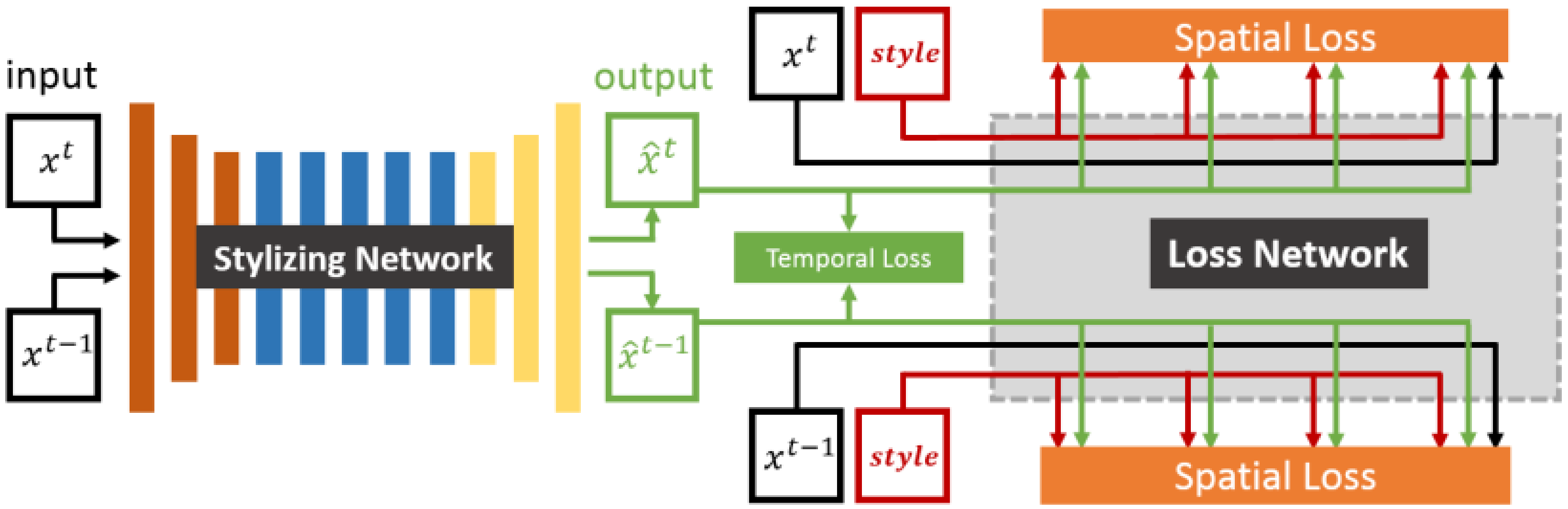
Avatar-Net: Multi-scale Zero-shot Style Transfer by Feature Decoration



(a) AdaIN (b) WCT (c) Style-Swap (d) Style Decorator

Video Style Transfer

Real-Time Neural Style Transfer for Videos



Video Style Transfer

Characterizing and Improving Stability in Neural Style Transfer

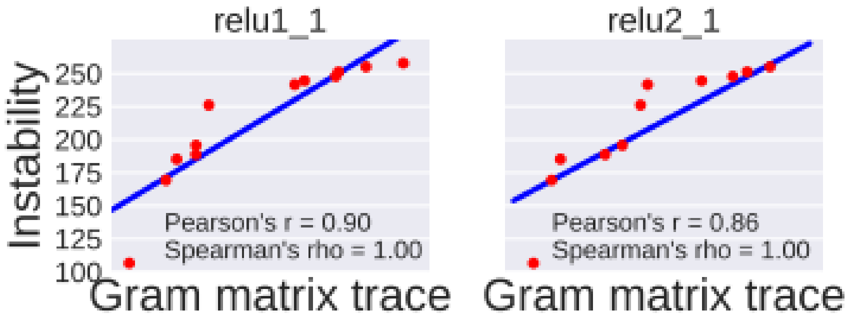
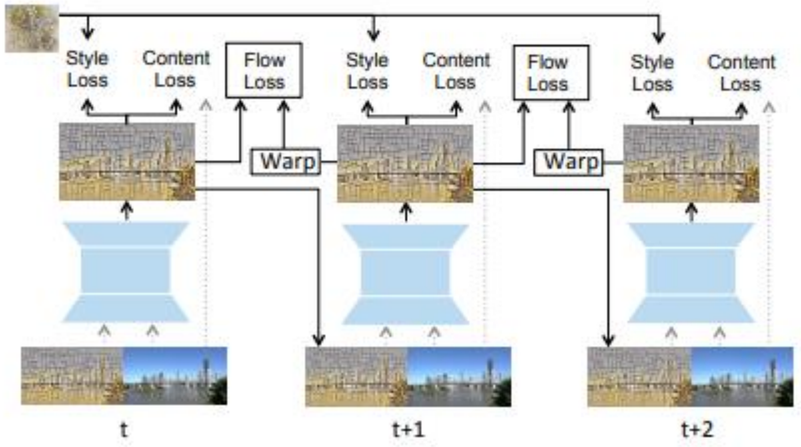


Figure 3. We train feedforward style transfer models for twelve styles, and define the *instability* of a style as the mean squared error between stylized adjacent frames over a dataset of videos with a static camera and no motion. We also compute the trace of the Gram matrix at two layers of the VGG-16 loss network for each style; styles with larger trace tend to be more unstable.



Photorealistic Style Transfer

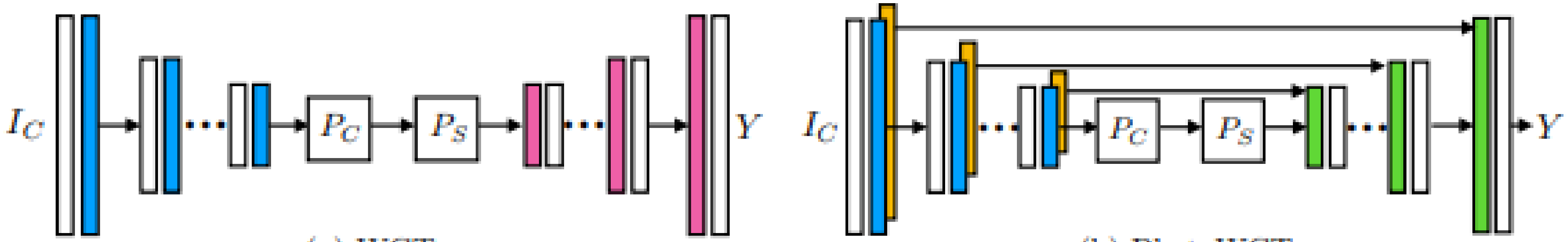
Deep Photo Style Transfer

Formally, we build upon the Matting Laplacian of Levin et al. [9] who have shown how to express a grayscale matte as a locally affine combination of the input RGB channels. They describe a least-squares penalty function that can be minimized with a standard linear system represented by a matrix $\mathcal{M}_I$ that only depends on the input image I (We refer to the original article for the detailed derivation. Note that given an input image I with N pixels, $\mathcal{M}_I$ is $N \times N$). We name $V_c[O]$ the vectorized version ($N \times 1$) of the output image O in channel c and define the following regularization term that penalizes outputs that are not well explained by a locally affine transform:

$$\mathcal{L}_m = \sum_{c=1}^{3} V_c[O]^T \mathcal{M}_I V_c[O] \qquad (2)$$

Photorealistic Style Transfer

A Closed-form Solution to Photorealistic Image Stylization



(a) WCT (b) PhotoWCT

Photorealistic Style Transfer

A Closed-form Solution to Photorealistic Image Stylization

$$\operatorname*{argmin}_{r} \frac{1}{2}\Big(\sum_{i,j=1}^{N} w_{ij} \Big\| \frac{r_i}{\sqrt{d_{ii}}} - \frac{r_j}{\sqrt{d_{jj}}} \Big\|^2 + \lambda \sum_{i=1}^{N} \| r_i - y_i \|^2\Big), \tag{4}$$

where y_i is the pixel color in the PhotoWCT-stylized result Y and r_i is the pixel color in the desired smoothed output R. The variable $d_{ii} = \sum_j w_{ij}$ is the diagonal element in the degree matrix D of W, i.e., $D = \mathrm{diag}\{d_{11}, d_{22}, ..., d_{NN}\}$. In (4), λ controls the balance of the two terms.

which encourages consistent stylization within semantically similar regions. The above optimization problem is a simple quadratic problem with a closed-form solution, which is given by

$$R^* = (1-\alpha)(I - \alpha S)^{-1} Y, \tag{5}$$

where I is the identity matrix, $\alpha = \frac{1}{1+\lambda}$ and S is the normalized Laplacian matrix computed from I_C, i.e., $S = D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \in \mathbb{R}^{N \times N}$. As the constructed graph is often sparsely connected (i.e., most elements in W are zero), the inverse operation in (5) can be computed efficiently. With the closed-form solution, the smoothing step can be written as a function mapping given by:

$$R^* = \mathcal{F}_2(Y, I_C) = (1-\alpha)(I - \alpha S)^{-1} Y. \tag{6}$$