Multi-style Generative Network for Real-time Transfer

Hang Zhang[1,2], Kristin Dana[2]

[1]Amazon AI, [2]Rutgers University

Example of transferred images and corresponding styles using MSG-Net

Overview:

- Introduce MSG-Net with a novel CoMatch Layer learning to match the feature statistics with the target styles at run time.
- Achieve the trinity of style transfer, including image quality, style flexibility and real-time performance.
- Enable run-time controls, including content-style interpolation, color-preserving, spatial control and brush stroke size control.

Method:

- Content and style representation (*Gatys et al.*) for input image x:
  - Activation of descriptive network $\mathcal{F}(x) \in \mathbb{R}^{C\times H\times W}$
  - Gram Matrix of the featuremap $\mathcal{G}(\mathcal{F}(x)) = \sum_{h=1}^{H}\sum_{w=1}^{W}\mathcal{F}(x)\cdot\mathcal{F}(x)^T$
- Ideal solution $\hat{y}$ for style transfer of input content image x_c and style imagex_s:

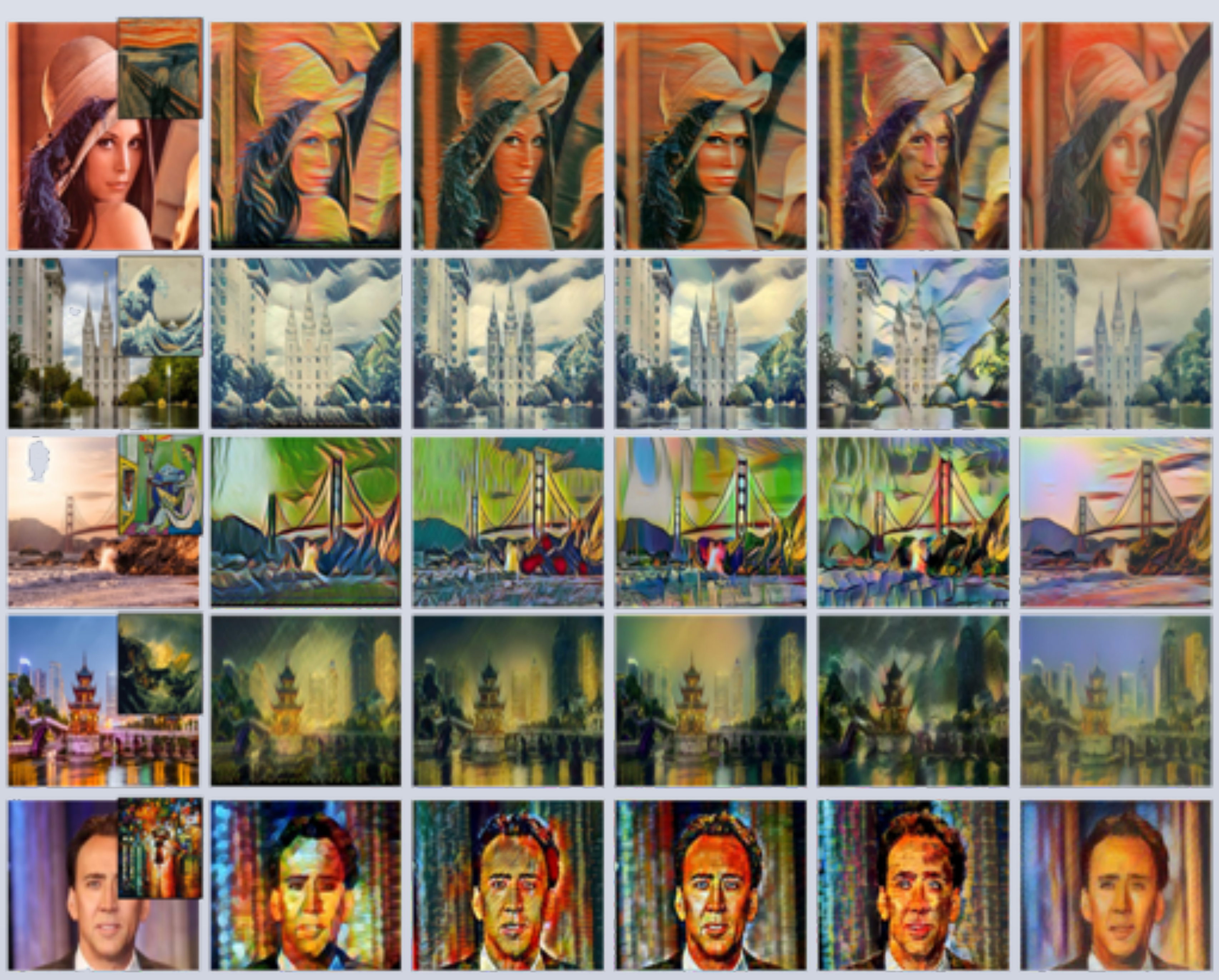
$$\hat{y} = \underset{y}{\operatorname{argmin}}\{\|y-\mathcal{F}(x_c)\|_F^2 + \alpha\|\mathcal{G}(y)-\mathcal{G}(\mathcal{F}(x_s))\|_F^2\}$$

- CoMatch Layer:

$$\hat{y} = \Phi^{-1}[\Phi(\mathcal{F}(x_c))^T W\mathcal{G}(\mathcal{F}(x_s))]^T$$

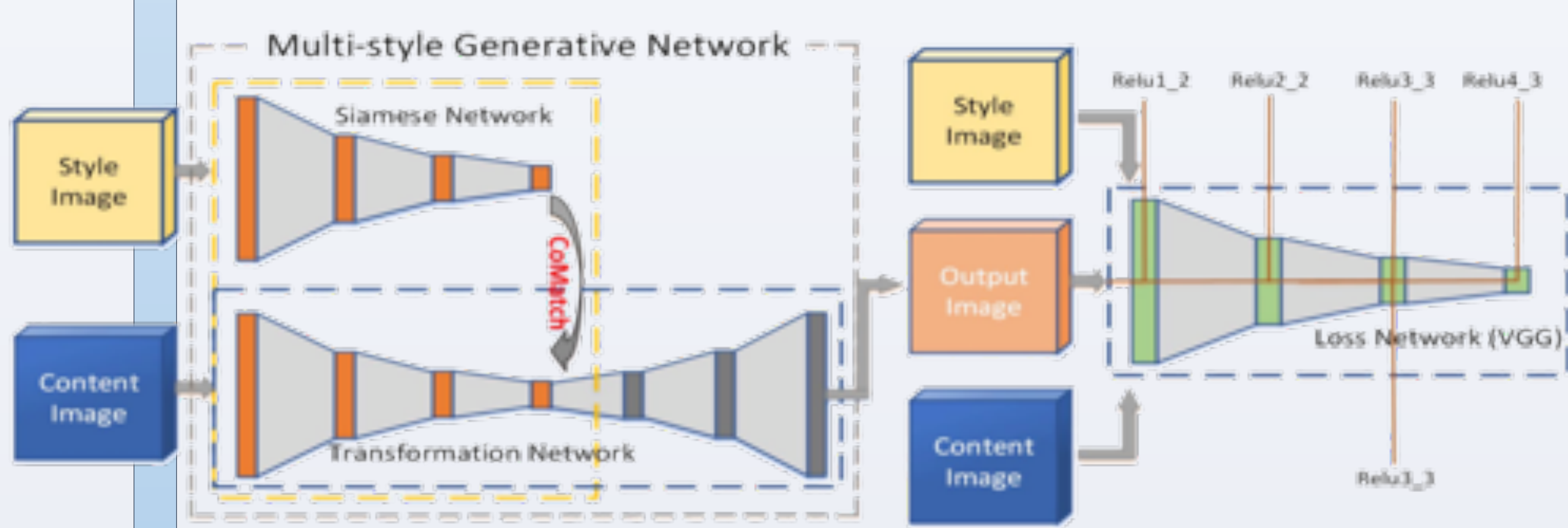
where $W \in \mathbb{R}^{C\times C}$ is a learnable weight matrix and $\Phi()$ is a reshaping operation.

- Intuition for learnable parameter W:
  - Let $W = \mathcal{G}(\mathcal{F}(x_s))^{-1}$, then $\|y-\mathcal{F}(x_c)\|_F^2$ is minimized
  - Let $W = \Phi(\mathcal{F}(x_c))^{-T}\mathcal{L}(\mathcal{F}(x_s))^{-1}$, where $\mathcal{L}(\mathcal{F}(x_s))$ is obtained by the Cholesky Decomposition of $\mathcal{G}(\mathcal{F}(x_s)) = \mathcal{L}(\mathcal{F}(x_s))\mathcal{L}(\mathcal{F}(x_s))^T$, then $\|\mathcal{G}(y)-\mathcal{G}(\mathcal{F}(x_s))\|_F^2$ is minimized.
  - We don't set W manually, but let it learned directly from the loss function instead.



input | Dumoulin *et al.* [9] | MSG-Net (ours) | Gatys *et al.* [13] | Huang *et al.* [21] | Chen & Schmidt [5]

Qualitative comparisons with other approaches, MSG-Net achieves superior performance.



An overview of MSG-Net, Multi-style Generative Network. The transformation network explicitly matches the features statistics of the style targets captured by a Siamese network using the proposed CoMatch Layer (introduced in Section 3). A pre-trained loss network provides the supervision of MSG-Net learning by minimizing the content and style differences with the targets

Results:

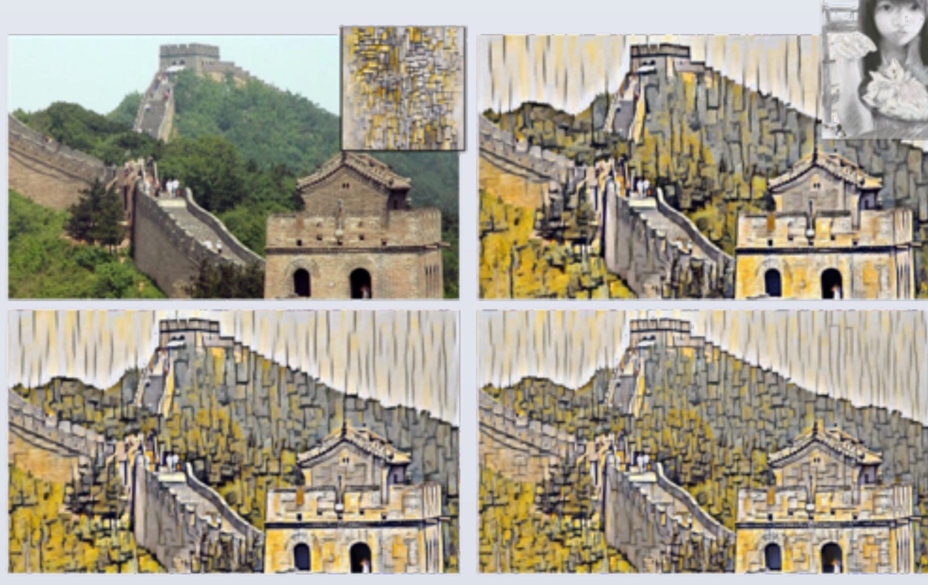
- Network learning:

Let the generative network be denoted as $G(x_c, x_s)$. The loss function is given:

$$\widehat{W}_G = \underset{W_G}{\operatorname{argmin}}\, E_{x_c,x_s}$$
$$\lambda_c\|\mathcal{F}(G(x_c,x_s)) - \mathcal{F}(x_c)\|_F^2$$
$$\lambda_s\sum_{i=1}^{K}\left\|\mathcal{G}\left(\mathcal{F}(G(x_c,x_s))\right) - \mathcal{G}(\mathcal{F}(x_s))\right\|_F^2$$
$$+\lambda_{TV}\ell_{TV}(G(x_c,x_s))$$

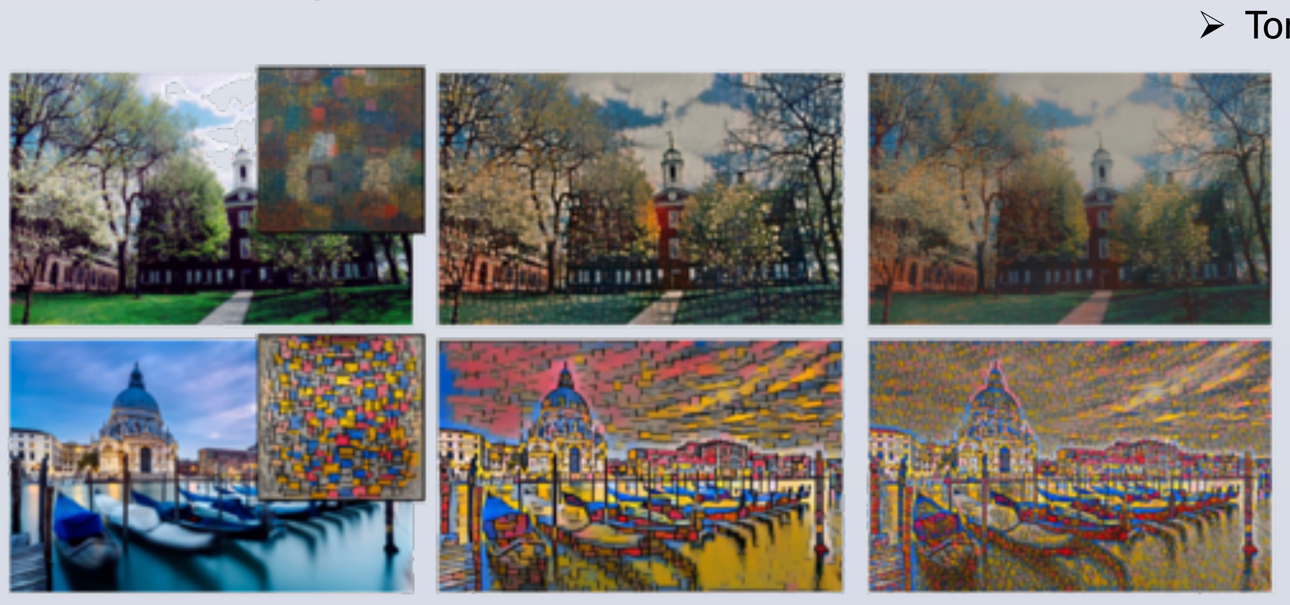
where λ_c and λ_s are the balancing weights for content and style losses. $\ell_{TV}()$ is the total variation regularization.

Content and style trade-off and interpolation.



Brush-size control using MSG-Net. Top left: High-resolution input image and dense style. Others: Style transfer results using MSG-Net with brush-size control.

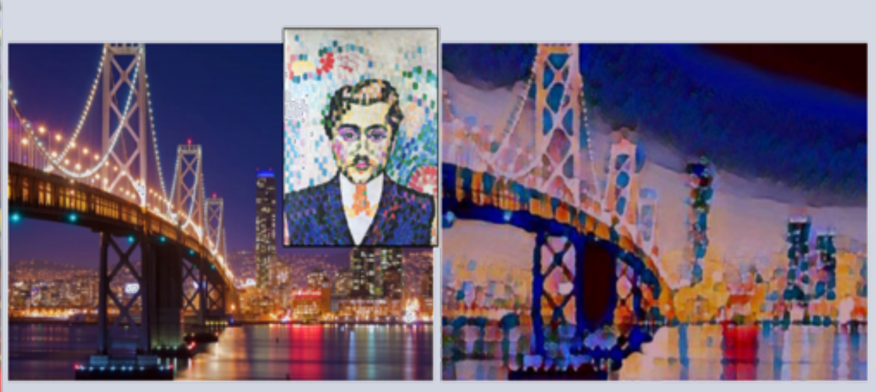
- Code Implementations:
  - PyTorch:
  - MXNet:
  - Torch:



(a) input | (b) MSG-Net (ours) | (c) baseline

Comparing Brush-size control. a) High-resolution input image and dense styles. b) Style transfer results using MSG-Net with brush-size control. c) Standard generative network without brush-size control.

Spatial control using MSG-Net. Left: input image, middle: foreground and background styles, right: style transfer result.



Color control using MSG-Net, (left) content and style images, (right) color-preserved transfer result.

Acknowledgment

This work was supported by National Science Foundation award IIS-1421134. A GPU used for this research was donated by the NVIDIA Corporation.