Cloud MoNet: An Adversarial Approach to Robust Unobserved Neural Style Transfer

Akhil Jalan, Noah Wu
CS 194-129: Deep Neural Networks, Spring 2018

Introduction

Goal: Increase the quality of **unobserved neural style transfer**.
Input: **Content image** $C \in \mathbb{R}^{H \times W \times 3}$ and an **arbitrary style image** $S \in \mathbb{R}^{H \times W \times 3}$
Output: **Stylized Image** $X = f(C, S) \in \mathbb{R}^{H \times W \times 3}$

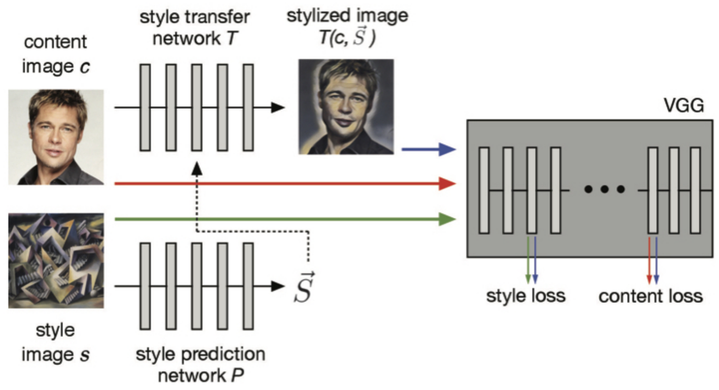

Figure 1: Unobserved neural style transfer applies stylization to any content, style image pair.[2].

Un-Stylization as Cycle-Consistency

Hypothesis: Every image can be decomposed into a content and style component. If true, then the "perfect" neural style transfer network should be able to not only stylize an image, but also un-stylize it. Give it the stylized image as input, and the original content image as style.

We call this property **cycle consistency**, in the spirit of [4].

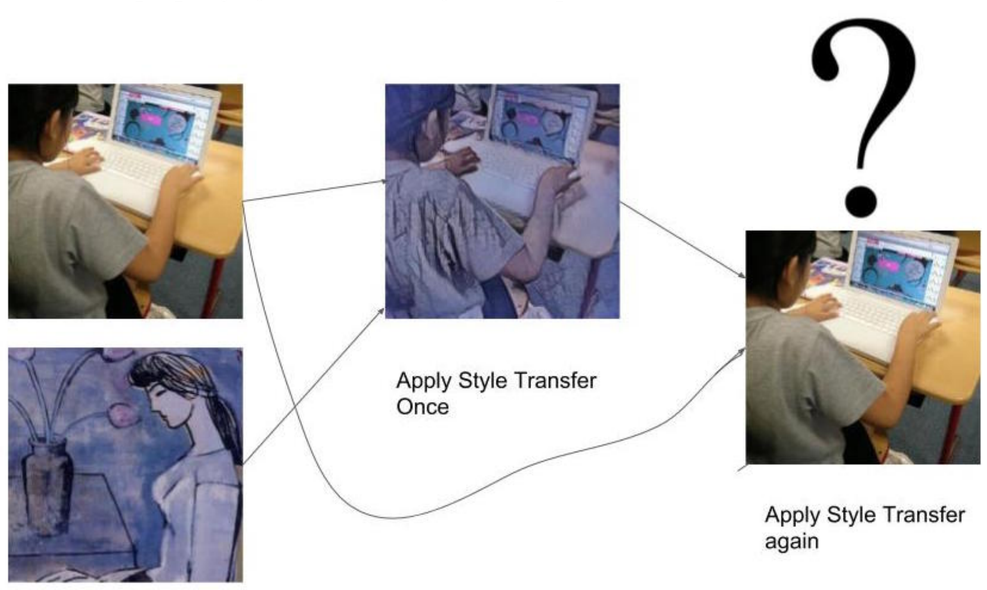

Figure 2: Apply style transfer once to create the image second from the left. We hypothesize that applying style transfer again, this time with the stylized image as "content" and the original content image as "style" should recover the original content image.

Metrics

To measure quality of image stylization, we use a **content loss**, **style loss** [1], and **total variation loss** for image smoothness [3].

$$\mathcal{L}(X, C, S) = \mathcal{L}_c(X, C) + \lambda_s \mathcal{L}_s(X, S) + \lambda_v \mathcal{L}_v(X) \leftarrow \text{Total Loss}$$

$$\mathcal{L}_c(X, C) = \frac{1}{n_c} \sum_{j \in N_C} \|f_j(X) - f_j(C)\|_2^2 \leftarrow \text{Content Loss}$$

$$\mathcal{L}_s(X, S) = \frac{1}{n_s} \sum_{i \in N_S} \|\mathcal{G}[f_i(X)] - \mathcal{G}[f_i(S)]\|_F^2 \leftarrow \text{Style Loss}$$

$$\mathcal{L}_v(X) = \frac{1}{3W(H-1)}[\sum_{i=1}^{H-1} \sum_{j=1}^{W} \sum_{k=1}^{3} (X_{i,j,k} - X_{i+1,j,k})^2]^{1/2} \leftarrow \text{Variation Loss}$$
$$+ \frac{1}{3(W-1)H}[\sum_{i=1}^{H} \sum_{j=1}^{W-1} \sum_{k=1}^{3} (X_{i,j,k} - X_{i,j+1,k})^2]^{1/2}$$

On the other hand, the discriminator wants to distinguish "real" from "fake" content images. "Real" images are original content images, while "fake ones" are those that were stylized and then un-stylized. We use typical cross-entropy loss.

Input: Image $Z \in \mathbb{R}^{H \times W \times 3}$.

Output: Prediction $\hat{p} = \begin{bmatrix} p_{real} & p_{fake} \end{bmatrix}$, $p_{real} + p_{fake} = 1.0$.

Label: $p = \begin{bmatrix} 1 & 0 \end{bmatrix}$ if real, $p = \begin{bmatrix} 0 & 1 \end{bmatrix}$ if fake.

Baseline Model

We use the style-transfer network from [1]. See Figure 1 on the left. The model consists of:

- A **style transfer network**: Takes in $C, \vec{S}$. Outputs a stylized image $T(C, \vec{S})$.
- A **style prediction network** Takes in S. Outputs embedding vector $\vec{S}$. This is used to compute the normalization constants $\{\gamma_s, \beta_s\}$. Uses the Inception v3 architecture[6].
- A **comparison network** which performs a forward pass on the style, content, and stylized image. Differences in its inner-layer activations are used to compute content, style los. Uses the VGG 16 architecture [5].

Final Model

We use the same style transfer network as the baseline model as our **generator**. Our **discriminator** is a simple convolutional network whose goal is to distinguish actual content images (which are "real") and reconstructed images which have experienced two rounds of style transfer (these are "fake").

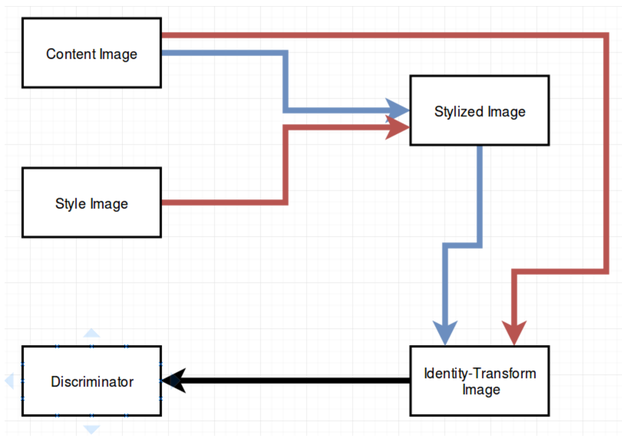

Figure 3: The complete model architecture. We apply style transfer to a pair of content, style images. We then apply another round of style transfer to the stylized image, with the original content image now being the style input. Finally, the discriminator outputs probabilities of real, fake. The model can thus be trained end-to-end.

Results

We found virtually no difference between the two models with respect to the original stylization's content or style loss. While the discriminator was able to easily achieve near-perfect accuracy quickly, the style transfer network was unable to learn to beat the discriminator.

Figure 4: From left-to-right: Content, style, stylized, and reconstructed image. The baseline model has no visual difference from the GAN version, although more training or a different procedure might result in a different outcome.

Conclusion

Style transfer networks are a rich model which deserve further study. Given that training GANs is so tricky, we think better results are possible given enough exploration of model architectures, training procedures, and hyperparameters. We hope that our work sheds light on the cyclic properties of style transfer, and illustrates how much room for growth exists in this field.

References

[1] Ghiasi, Golnaz, et al. "Exploring the structure of a real-time, arbitrary neural artistic stylization network." arXiv preprint arXiv:1705.06830 (2017).

[2] `https://github.com/tensorflow/magenta/tree/master/magenta/models/arbitrary_image_stylization`

[3] Chambolle, Antonin. "An algorithm for total variation minimization and applications." Journal of Mathematical imaging and vision 20.1-2 (2004): 89-97.

[4] Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." arXiv preprint arXiv:1703.10593 (2017).

[5] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. IEEE Computer Vision and Pattern Recognition (CVPR), 2015.

Email: {*akhiljalan@berkeley.edu*, *noahywu@berkeley.edu*}
Acknowledgments: We thank Professor John Canny and the Graduate Student Instructors Erin Grant and Carlos Florensa for their advice and support.