Localized Style Transfer

Alex Wells
Stanford University
awells2@stanford.edu

Jeremy Wood
Stanford University
jwood3@stanford.edu

Minna Xiao
Stanford University
mxiao26@stanford.edu

Abstract

Recent groundbreaking work has demonstrated how convolutional neural networks can be used to perform style transfer. Style transfer is a method of transferring the style of one "style" image to the entirety of a "content" image, resulting in a newly rendered image. We outline a strategy for performing Localized Style Transfer, or the transfer of style from one image to only a portion of the content image. By using a combination of semantic segmentation, a modification to the canonical neural style loss function, and Markov Random Field Blending, we show how desired regions of an image can be altered while leaving the rest of the image intact.

1. Introduction

1.1. Motivation

With the seminal publication of "A Neural Algorithm of Artistic Style" in 2015, Leon Gatys et al. [7] introduced the application of convolutional neural networks to the problem of image style transfer. Style transfer involves the transfer of the style of an image to another, while preserving the content of the target image. Research using the Gatys et al. algorithm, and subsequent improvements, have produced impressive results of images in the rendering of various artistic styles, from Van Gogh to Kandinsky. Figure 1 depicts an application of the neural algorithm.

In the past couple of years, style transfer has made the leap from the pages of academia into the hands of the masses. In 2016, Russian-based Prisma Labs launched the mobile-phone application Prisma, which allows users to upload photos and videos and transfer them in the style of various artistic filters. The app's technology is built off the neural algorithm introduced by Gatys et al. Within months of its release, the app had over two million daily active users[12]. Researchers at Facebook AI Research (FAIR) developed the lightweight Caffe2go deep-learning framework, which allows for the real-time running of style transfer on mobile phones [9], processing up to 20 frames per second. Facebook has begun rolling out a new creative-effect camera in the mobile app that gives users the ability to apply style transfer on videos and images. Such industry applications of style transfer are bringing deep-learning-powered artistic expression to billions of social-media users worldwide.

1.2. Problem Statement

Most of the applications of style transfer thus far have focused on style transfer onto the entire image. We would like to center our attention instead on approaches for achieving visually pleasing partial-image style transfer. This would allow for the selection of specific regions in the original image to be altered, while the rest of the original image maintains its appearance.

2. Related Work

Semantic Segmentation Research in the field of semantic segmentation has made rapid gains in the past decade. The application of a fully convolutional network trained end-to-end on semantic segmentation achieved state-of-the-art results on the PASCAL-VOC dataset in 2015 [13]. The CRF-RNN network (conditional random fields as recurrent neural networks) [21] takes advantage of CRFs to formulate the semantic label assignment problem as a probabilistic inference problem to achieve finer segmentations. The MNC framework introduced by Dai et al. [5] achieved state-of-the-art results on the 2015 Microsoft COCO dataset, performing instance-aware semantic segmentation using multi-task network cascades. Just recently, He et al. [8] presented the Mask R-CNN architecture, which improves upon Fast R-CNN and Faster R-CNN to perform pixel-level segmentation.

Style Transfer Gatys et al. [7] pioneered the application of deep convolutional networks for whole-image artistic style transfer, introducing a custom loss function that includes both a content loss and style loss. Johnson et al. [10] improved the performance time of real-time style transfer by optimizing a perceptual loss function instead of per-pixel loss. Recent work has built off the algorithm of Gatys et al. to explore the extensions of neural style transfer to various specialized tasks. Luan et al. [14] introduced an approach

(a) Original image of a cow in nature.

(b) Style: a Pollock-esque image, *Jump In*

(c) Vanilla style transfer

Figure 1: Style transfer on a photograph of a cow using Gaty et al.'s neural algorithm for style transfer

for photographic style transfer that preserves the photorealistic quality of an output image when transferring the style of one photograph to another. Their work is a modification upon the Gatys et al. technique, which in comparison yields a painterly style transfer. Selim and Elgharib [16] introduced a technique for portrait style transfer that imposes spatial constraints using gain maps during transfer; their implementation reduced the facial deformations that would otherwise occur during neural style transfer. The authors' research is a neural-based approach to the application of headshot portrait stylization [18]. In 2016, Gatys et al. introduced an extension to their own previous work, presenting two different methods, color histogram matching and luminance-only transfer, for the task of color preservation of the source image during neural style transfer [6, 17].

Some recent work has begun to focus on the task of partial-image style transfer, as opposed to whole-image style transfer [3, 4]. A paper by Castillo et al. [3] presented work on targeted style transfer using instance-aware semantic segmentation, in which only part of an image is altered by the style of a template source image. The authors' implementation combines the Gatys et al. algorithm for style transfer with MNC instance segmentation [5] to create the generated images using simple mask transfer and blending the source style image with the target content image using Markov Random Fields (MRF).

Markov Random Fields are used throughout image analysis and computer vision as a way to model structures within images. Describing the statistical relationships between pixels and/or areas within an image. Previous work has shown that it is possible to optimize such fields efficiently using stochastic gradient descent [20], which allows for easier development in deep-learning frameworks such as Google's Tensorflow. Finally, generative MRFs have previously been used for image synthesis in order to create smoother images [15]. It has also been used in combination with convolutional neural nets to generate images with fewer over-excitation artifacts and implausible feature mixtures [11]

3. Methods

We aim to improve upon the technique for semantic style transfer to create images with more naturalistic and fine-grained style transfer. To achieve this goal, we split the targeted style transfer task into three main components, which are detailed in this section.

3.1. Semantic Segmentation

First, we utilized a Caffe implementation of a CRF-RNN (conditional random field-recurrent neural network) network trained on the Pascal VOC dataset to generate a mask for a given input image [19, 22, 1]. Broadly, the CRF for pixel-wise labeling models each pixel in an image as random variables that form a Markov Random Field when conditioned on a global observation. In this context, the global observation is the image upon which we wish to perform semantic segmentation. A CNN predicts labels for each pixel without accounting for the smoothness or consistency of the label assignments. However, after minimizing over a loss function that encourages similar pixels to have the same label, while conditioning its prediction on the entire input image, this formulation can output the most probable class label for each pixel in the image. By using a RNN to perform the tasks required for CRF inference, we can utilize this CRF-RNN strategy for semantic segmentation. This method has achieved state-of-the-art results on the Pascal VOC dataset.

We used the CRF-RNN to generate a mask for a given input image. We then binarized the output masks such that each pixel in the mask was set to a value of 0 or 1. Pixels with a value of 0 correspond to regions in the original image that we want to leave untouched by style transfer, whereas pixels with a value of 1 denote regions to which we wish to transfer style. We use this mask when calculating our style loss function to perform localized style transfer.

3.2. Baseline: Naive Masked Transfer

The naive implementation of partial-image style transfer, which we refer to as Naive Masked Transfer, involves

first performing whole-image style transfer on the target image. Then, a mask of the original image is generated using segmentation, and the pixels of the stylized image corresponding to the portion of interest delineated by the mask are transferred onto the non-stylized image. While this is a straightforward approach to tackling the partial-image style transfer problem, the results often look quite crude and almost completely depend on the quality of the performance of the semantic segmentation used, i.e. how fine-grained the segmentation is.

Thus we explore two options for achieving a more natural localized style transfer. First, we introduce a modification to the canonical style loss function using masks from semantic segmentation. Second, we explore the use of Markov Random Fields for the smoothing of boundaries between regions subjected to style transfer and regions that are left untouched.

3.3. Masked Style Loss

The algorithm introduced by Gatys et al. minimizes a custom loss function that sums over a style loss, content loss and total-variation loss:

$$L_{total} = \alpha L_{style} + \beta L_{content} + \gamma L_{tv} \tag{1}$$

. In order to achieve style transfer on a segment of the image rather than the entire image, we implement a modification to the style loss calculation, where the original style loss calculation is the sum of the style losses for a set of layers $\mathcal{L}$, (conv1_1, conv2_1, conv3_1, conv4_1, conv5_1):

$$L_{style} = w \sum_{\mathcal{L}} (G - A)^2 \tag{2}$$

where G and A are gram matrices at a given layer for the content and style images. At each layer, for the current image x and the source style image s, we have feature maps F_x and $F_s \in \mathbb{R}^{WxHxD}$ that represent the activations of the D filters for each of the spatial positions in x and s. Prior to computing the Gram matrices representing the feature correlations for x and s, we create a mask volume, $mask \in \mathbb{R}^{WxHxD}$, from the binary mask generated by our segmentation system, which we then apply to the feature maps in order to mask over the spatial extent of each filter in the volume.

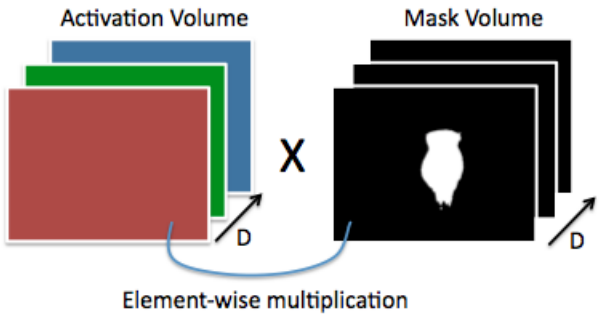


Figure 2: Masked style loss. In order to apply a mask over the style loss, we use a mask generated by our CRF-RNN and extend it over the dimensions of the activation volume. For an activation volume with depth D, we stack D masks on top of each other to obtain the same spatial extent. We then performed element-wise multiplication to calculate a masked activation volume.

This results in the zeroing of regions of the activation volume correspond to pixel locations that we want to leave untouched by style transfer, while maintaining values in regions which we wish to transfer style.

3.4. Markov Random Field Blending

As noted in Related Work, Markov Random Fields describe a markov relationship between nodes in a random field (a set of random variables). By constraining the relationships between these variables we can create emergent effects. These fields can be used in image processing to blend two images together (i.e. a style transfered image and the original) such that they make the boundary between the two images as imperceptible as possible. One instance of blending proposed by Castillo et al. [3] minimizes a unary constraint on pixels and a binary relationship between pixels.

In our case, we first determined a border region on the edge of the segmentation mask and then used a unary cost that encouraged pixels along the border to be assigned to foreground or background based on which set-assignment zone (foreground/background) they are closest to. Given a pixel p and a potential label for that pixel l, the equation used by Castillo et al. is as follows, where c^l is the closest label:

$$U(p, l) = ||p - c^l|| \tag{3}$$

We modified this to allow for the labeling to be a continuous value [0, 1]. Our intuition was that true blending would result from merging images into each other along their boundary gradually, not through a binary labeling. Therefore we instead use l^0 and l^1 ($l^0 = 1 - l^1$) to represent the relative amount that a pixel should be assigned to the background and foreground respectively. Ultimately, when blending,

given a pixel's original value p_o and its style-transfered value p_s we blended them with

$$p = l^0 p_o + l^1 p_s = l^0 p_o + (1 - l^0) p_s$$

. Noting this, we developed the following equation for a given pixel and its labelings:

$$U(p, (l^0, l^1)) = l^0 ||p - c^0|| + l^1 ||p - c^1|| \quad (4)$$

We then applied a binary cost that minimized the resulting contrast within the border region after the masked style transfer has been applied to the original image (anti-aliasing it). Loosely, this encourages pixels in the border region to adapt a foreground or background label (now continuous) based on what creates the smoothest blend between stylized image and original image. Specifically, it computes the difference in intensity if two neighboring pixels were given the label of the other pixel. The equation used by Castillo et al. was:

$$B(p_1, l_1, p_2, l_2) = |I_{l_1}(p_1) - I_{l_2}(p_1)|^2 + |I_{l_2}(p_2) - I_{l_1}(p_2)|^2$$

Where $I(x)$ denotes the intensity (aka grayscale value) of pixel x. However given our continuous labels we had to use a continuous version of this equation:

$$\begin{aligned} B(p_1, (l_1^0, l_1^1), p_2, (l_2^0, l_2^1)) = {} & l_1^0 l_2^0 B(p_1, 0, p_2, 0) \\ & + l_1^1 l_2^0 B(p_1, 1, p_2, 0) \\ & + l_1^0 l_2^1 B(p_1, 0, p_2, 1) \\ & + l_1^1 l_2^1 B(p_1, 1, p_2, 1) \end{aligned}$$

Naively, this approach can be used on a fully stylized image (no segmentation), the segmentation mask, and the original image to blend them all together. However we also applied this approach to our Masked Style Loss approach to revert the non-masked portion of the image to the original content.

4. Datasets

To generate the masks needed for our localized style transfer, we use a CRF-RNN network (condition random fields as recurrent neural networks) for semantic image segmentation. We trained this network on the PASCAL-VOC dataset, which consists of 20 classes, including people, birds, cats, cows, and dogs. This dataset contains 11,530 images containing 27,450 ROI annotated objects and 6,929 segmentations.

To perform the style transfer portion of our project, we used a VGG-19 network pretrained on the ImageNet dataset.

5. Results

In the field of neural style transfer, there is no established quantifiable methodology for evaluating the performance of our approaches. In fact, the appraisal of an artistic image can be quite a subjective experience. For the purposes of our paper, we rely on qualitative assessment of our generated localized style transfer images to gauge the success of our methods. For instance, to see how well we achieve our task of localized style transfer, we look at whether our system can transfer style onto the segmented portion without transferring that same style (colors, patterns, etc.) to the background. Additionally, we benchmark our Masked Style Loss and MRF-blending methods against the results of Naive Masked Transfer.

5.1. Semantic Segmentation

The goal of semantic segmentation is to label each pixel with a given class. Since our CRF-RNN model was trained on the Pascal VOC dataset, it can label each pixel in a given input image as one of twenty different classes. To determine how well this model performed, we qualitatively examined the outputs of the CRF-RNN on a variety of images.

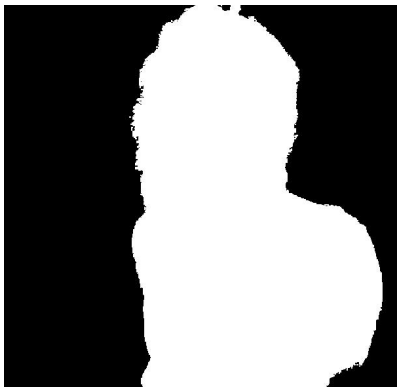
Figure 4: Result of CRF-RNN semantic segmentation on an image of an cow.

The Figure 4 model does a good job of correctly labeling all pixels in the cow except its horns, which are erroneously labeled as part of the background.

(a) Original photograph

(b) Masked Style Loss technique

(c) Style: Warhol's *Marilyn Monroe 31*



(d) Segmentation mask

(e) Masked Style Loss technique

(f) Style: Van Gogh's *Starry Night*

Figure 3: Masked Style Loss, Marilyn Monroe

5.2. Baseline: Naive Masked Transfer

Figure 5: Naive Masked Transfer

The result of the Naive Masked Transfer is shown in Figure 5. To obtain this result, we performed style transfer over the entire image of an cow (Figure 4,left image) with a Pollock-style painting (Figure 6). Using a mask generated by our CRF-RNN, we then replaced pixels in the resulting image with pixels from the original image. This enabled us to recover the background from the content image, while performing style transfer on the cow. We can see that the resulting style transfer does not do a great job of capturing the intricate features of the style image. Furthermore, the placement of the cow in the rendered image looks artificial, as if the cow were Photoshopped into the center of the landscape: the edges of the cow are distinctly jagged. Instead, we wish to smooth the edges of the cow, in order to achieve a more natural look.

5.3. Masked Style Loss

We modified the style transfer loss function using a mask in order to perform localized style transfer, adapting a Tensorflow implementation of vanilla neural style transfer [2]. Figure 6 is the result of Masked Style Loss on the input image of an cow using the *Jump In* painting.

Figure 6: Masked Style Loss technique using the *Jump In* style image.

We can see that much of the unique style of the Pollock-esque painting is transferred to the cow in the center of the content image. Compared to the baseline Naive Masked Transfer, the Masked Style Loss method more precisely transfers the elements of the style image to the cow. We can also see that the background of the cow image is slightly distorted. In particular, the clouds and the sky in the background appear darker and more "painterly" in the newly rendered image, but still without significant style influence from *Jump In*. This demonstrates that although Masked Style Loss does not perfectly maintain the integrity of the original image, the technique does manage to stop the colors and patterns of the style source from transferring over to the background.

Figure 3 also visualizes the output of Masked Style Loss on a photograph of Marilyn Monroe. After using the Masked Style Loss technique with Figure 3c as the style image, we obtain the result shown in Figure 3b. Again we see that the style transferred from the Warhol artwork is localized to Marilyn Monroe herself. The background is lighter in color but otherwise retains most of its original style and content. We found the effects of the Warhol style transfer particularly pleasing as the bright-yellow color of Warhol Marilyn's hair is transferred over to the localized Marilyn's hair. A similar result using Van Gogh's *Starry Night* is shown in Figure 3e.

5.4. Markov Random Fields

Figure 8 shows the effects of first applying no MRF blending (just masked style transfer), then applying only unary costs to encourage the boundary to fade out between foreground and background, and then applying the full MRF. The full MRF, which applies the mask shown in Figure 7, shows clear improvements over pure masked style transfer. Results from this and other images suggest that applying a secondary blending on top of masked style transfer has strong potential for creating better merging of stylized and unstylized segments in images. In the future it could also be used to blend together multiple different styles into one contiguous image.

Figure 7: The MRF-generated mask used in the final image in Figure 9

6. Discussion

Overall we saw the best results from combining both of our methods, as depicted in Figure 5. Note the uniformity of the texture applied and the extremely weak boundary zone between the cow and the background.

Masked Style Loss on its own, as can be seen in Figure 6, applies a more interesting palette to the segmented image. However the blending between foreground and background is still a little rough in places, as shown in 8, and the process altered the background slightly. Applying MRF-blending to blend the new image produced by Masked Style Loss with the original using the mask and Markov random fields allowed us to regain the photographic background while retaining the improved style transfer. Moreover, the Masked Style Loss technique already made the borders more amenable to blending than naive transfer. Figure 9 shows the final result of combining the methods: a more colorful and well-blended version of our earlier transfer attempts.

7. Conclusion and Future Work

While it is hard to objectively evaluate style transfer – as artistic merit is largely a subjective domain – we feel confident that this paper contributes toward creating seamless style transfer that can be applied to a localized region of images based on semantic segmentation in such a way that the image modifications blend well into the background. For future work, we would like to implement the strategy outlined in Mask R-CNN by He et al. [8] for the semantic segmentation part of our localized style transfer pipeline. This framework has achieved state-of-the-art results on the

(a) Naive masked transfer (b) Unary-cost-only MRF blending (c) Full MRF blending

Figure 8: From left to right: no MRF blending, unary-cost-only blending, full blending with unary costs and binary costs. Particularly, note the differences within the respective highlighted sections of the image. These are all parts of the image along the boundary between the segmented image and the background. As one can see, slight differences exist between the first and second image but the largest improvement is from the second to the third – after binary costs are introduced to the MRF.

Figure 9: Final result of using both Masked Style Loss and Markov Random Field blending

COCO 2016 challenge, and would bolster our ability to extract regions of interest in our input images, particularly from more challenging images. Another interesting direction would be to play around with more specific domains, such as the application of localized style transfer on human faces. Such a task could involve training off a dataset composed entirely faces, instead of the full ImageNet dataset, for the VGG-19 model, and also the further modification of the style transfer loss functions to account for the spatial considerations of facial features.

References

[1] A. Arnab, S. Jayasumana, S. Zheng, and P. H. S. Torr. Higher order conditional random fields in deep neural networks. In *European Conference on Computer Vision (ECCV)*, 2016.

[2] A. Athalye. Neural style. `https://github.com/anishathalye/neural-style`, 2015.

[3] C. Castillo, S. De, X. Han, B. Singh, A. K. Yadav, and T. Goldstein. Son of zorn's lemma: Targeted style

transfer using instance-aware semantic segmentation, 2017. arXiv:1701.02357v1.

[4] E. Chan and R. Bhargava. Show, divide and neural: Weighted style transfer, 2016.

[5] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016.

[6] L. A. Gatys, M. Bethge, A. Hertzmann, and E. Shechtman. Preserving color in neural artistic style transfer. *arXiv preprint arXiv:1606.05897*, 2016.

[7] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *arXiv preprint arXiv:1703.06870*, 2017.

[9] Y. Jia and P. Vajda. Delivering real-time ai in the palm of your hand. code.facebook.com/posts/196146247499076/delivering-real-time-ai -in-the-palm-of-your-hand, 2016.

[10] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[11] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. *CoRR*, abs/1601.04589, 2016.

[12] N. Lomas. Prisma launches a social feed to see if style can transfer into a platform. https://techcrunch.com/2016/12/20/prisma-launches-a-social-feed-to-see-if-style-can-transfer-into-a-platform/, 2016.

[13] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. `https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf/`, 2015.

[14] F. Luan, S. Paris, E. Shechtman, and K. Bala. Deep photo style transfer. *arXiv preprint arXiv:1703.07511*, 2017.

[15] R. Paget and I. D. Longstaf. Texture synthesis via a noncausal nonparametric multiscale markov random field. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 7(6), 1998.

[16] A. Selim, M. Elgharib, and L. Doyle. Painting style transfer for head portraits using convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 35(4):129, 2016.

[17] X. Shen, A. Hertzmann, J. Jia, S. Paris, B. Price, E. Shechtman, and I. Sachs. Automatic portrait segmentation for image stylization. In *Computer Graphics Forum*, volume 35, pages 93–102. Wiley Online Library, 2016.

[18] Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand. Style transfer for headshot portraits. 2014.

[19] P. Torr. Crf-rnn. `https://github.com/torrvision/crfasrnn`, 2015.

[20] S. V. N. Vishwanathan, N. N. Schraudolph, M. W. Schmidt, and K. P. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 969–976, New York, NY, USA, 2006. ACM.

[21] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[22] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision (ICCV)*, 2015.