A closer look at Neural Style

Anonymous Author(s)
Affiliation
Address
email

Abstract

The neural style algorithm is successful and popular in transferring styles from one image to another. The key insight of this algorithm is that the "style" component and the "content" component can be extracted and separated so that a new image can be created based on the style of one image and on the content of the other. However, the current neural style algorithm is not able to transfer different styles to different parts of the images. This report demonstrates a technique that makes image segmentation techniques and the neural style algorithm work together to produce images with appropriate styles to different segments. Four approaches are proposed to address different issues.

1 Introduction

Visual creation is full of human being footprints. In the history, visual creation witnesses the power of style transferring. Observing mother nature's work, human beings learn to create artwork with similar textures and patterns, like decorations that look like grass and leaves. Observing human bodies and social life, human beings learn to create the images of the Gods. Nowaday, human beings are able to teach machines to learn and transfer styles of an image since the release of the Neural Style algorithm [1]. There are a number of applications that use it to do visual creation.

Despite the success and popularity of the algorithm, one limitation stops it from the room of real creation: it only applies the style to the whole image. This is not a problem for images that contain only one type of "objects," like a scenery, a portrait, or a still life painting. But when the image contains multiple objects, typically a photograph from everyday life, the limitation renders the algorithm useless: how likely people will like their selfie looks like the Starry Night painting? Figure 1. is one such failure style transfer. On the other hand, it is delighting and meaningful to apply different styles to different parts of the image. One may like to use a portrait and a scenery painting from the same artist for a group photo during camping. Others may just like the starry night as a background for a photo of tall buildings in a city.

Figure 1: Failure case of the neural style algorithm

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

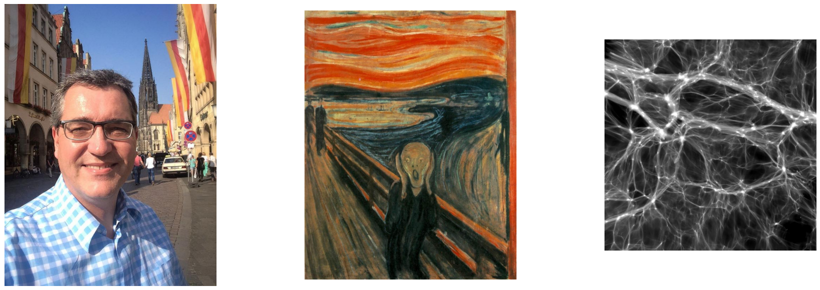

Figure 2: Left: input/content/original image, right: style images used for style transfer

To address this issue and extend the algorithm for more possibilities, this report demonstrates 4 different approaches that leverage image segmentation technique as preprocessing to apply neural nets on different segments. The rest of the report begins with a review of the neural style algorithm and the segmentation technique. The report then introduces the methodology, the results and the discussion of the 4 different approaches. This report uses the following **content image** (left) and **style images** The Scream and the Black Matter (right) as input to this work. In the demonstration of each approach, the person and the background are 2 segments and the task is to transfer the style of the The Scream to the background and the Black Matter to the person. This work can also take multiple segments as input. Due to the lack of GPU support, the result images are obtained from up to 100 iterations. Although this makes the styles applied not visually obvious, it is enough to demonstrate this work.

2 Review of previous work

2.1 Neural Style [1]

Neural style algorithm "uses image representations derived from Convolutional Neural Networks (CNN) optimised for object recognition, which makes high level image information explicit." The key insight of this algorithm is that given an original image to transfer to the style of a style image, the content information and style information can be separated and represented by a CNN. The content information comes from a deep layer of the feature maps, which is a deep representation of the content image that is able to do reconstruction. The style information is built on top of response of a layer of feature map using the Gram matrix as the inner product between the vectorised feature maps i and j in layer l:

$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l} \tag{1}$$

The algorithm is built on top of a VGG-Network and uses conv_4 layer for content representation and conv_1, conv_2, ..., to conv_5 for style representation. The algorithm starts from capturing the content representation from the original image and capturing the style representation from the style image. The algorithm then uses a random white noise image, computes the content representation and style representation using the same CNN, and performs optimization by measuring the total loss between the difference of the content representation $\mathcal{L}_{content}$ and the difference of the style representation $\mathcal{L}_{style}$. The framework is shown in figure 3.

2.2 Semantic Segmentation [2]

Semantic segmentation is the task of clustering parts of images together which belong to the same object class. Taking an image as input, a semantic segmentation algorithm assign labels to each pixel and produces segments corresponding to each label. It is a pixel level classification of image objects. It is also a necessary step for producing semantic segments of an image.

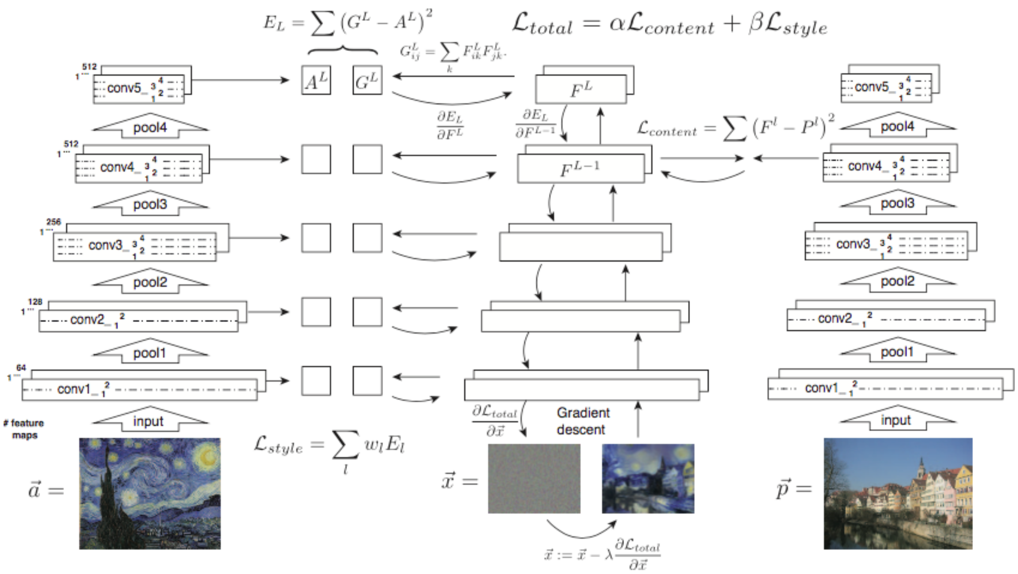

Figure 3: Neural style framework [1]

3 Methodology

3.1 Straightforward Approach

The most straight forward approach is a multi-pass approach that passes each pair of a content image segment and a style image through the neural style algorithm and then combine the corresponding results. The general framework is shown in Figure 4.

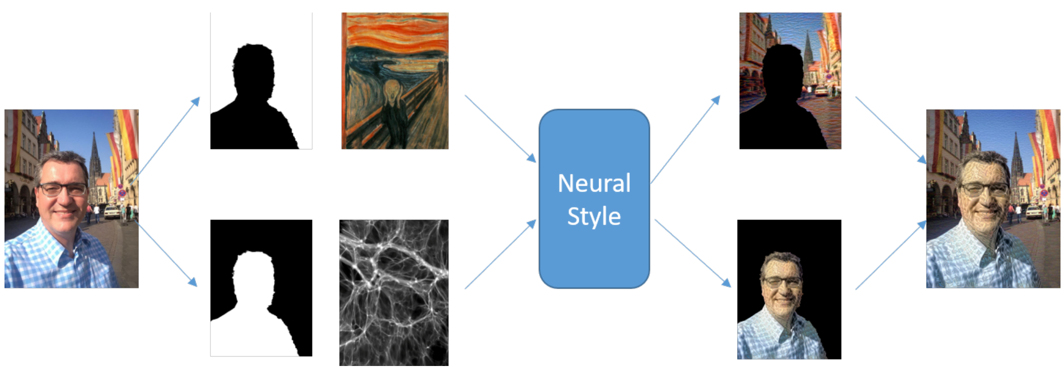

Figure 4: Framework of Straightforward Approach

Although naive, this method addresses the issue of style transfer for different segments, especially when the transfer task requires a "clear cut" among the styles of different segments. However, also because of the absolute distinction between different segments, the approach will fail when dealing with transfer and blend task. For example, an image with *Starry Night* in the background and *The Scream* style in the foreground, the contrast can be uncomfortable to some people's aesthetics. A solution to this issue is blending the styles. The next section illustrates this part.

3.2 One-image Approach

Unlike the previous approach, this approach assumes dependency among different styles. This approach takes one content image and applies different style loss function to different segments and compute the weighted sum of the different loss. That is to say, this new loss function observes the dependency among different styles. The general framework is shown in the left image of Figure 5. The name of this approach follows the fact that it takes only one content image as input. This approach has only one pass and thus has less complexity compared to the straightforward approach.

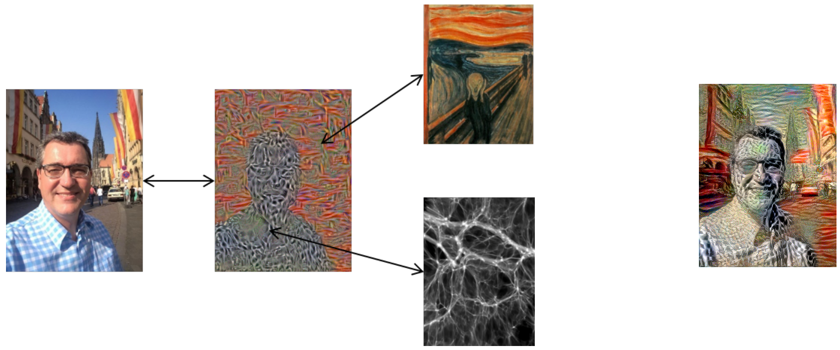

Figure 5: Left: framework of One-image Approach. Right: result image of One-image Approach at 100 iteration

The style loss function changes to:

$$\mathcal{L}_{style}(\vec{\mathbf{a}}, \vec{x}) = \sum_{\vec{\mathbf{a}}} w_a \sum_{l=0}^{L} w_l E_l, \tag{2}$$

Where $\vec{\mathbf{a}}$ is multiple style images, $\vec{x}$ is the input image, E_l is the style loss of the layer l, w_l is the layer weight of layer losses, and w_a is the blending weight of style images. The intuition of this loss function is that the loss from transferring different styles all comes together. Thus in order to minimize the total loss, the gradient descents from different segments need to redeem the exist of others and restrict their own behavior.

From the result image at the iteration 100 (the right image of Figure 5), it is observable that the style transfer from *The Scream* favors the cool color part, with respect to the style transfer from the other image which is black and white. At the mean time, the style transfer from the black-and-white image is not harsh in color and favors the texture, with respect to the style transfer from the colorful *The Scream*.

However, the result is not desirable to some extend in that the result image does not looks like *The Scream* in some parts and the other black-and-white image in other parts. The reason is two-fold. First, the style transfer happens across different layers through out the deep neural network representation and thus the segments in the result image will have slight overlap. But this overlap means redundant loss computation to the boundary area and thus the result image above shows total blending of two styles and renders the shape of the objects start fading. Second, the style loss $\mathcal{L}_{style}$ is uniformly distributed across the image so that a certain style exhibiting in a style image can appear anywhere in the new image. This is why the sky becomes in the result image actually inherits the style from the left and right side part of *The Scream*. The following sections address these issues.

3.3 One-image Approach with Locality Loss

This approach mainly remedies the first issue mentioned above: transfer overlapping. The overlapping issue can be seen from the following figure, which is the result of the 20th iteration from transferring only the person (left) and only the background (right). One the left-hand side, the style that is supposed to apply to only the person also affects the boundary, especially around his shoulder and makes the area cooler in color. On the right-hand side, the style that is supposed to apply to only the background also affects the person, especially alongside the face, making it colorful. This is not an implementation error but because the high-level feature layer in a CNN corresponds to multiple pixels in the low level.

Since the architecture is inevitable in causing this issue, this approach seeks penalty to the repeated loss on the boundary area and introduces a new locality loss that measures the pixel distance to the nearest zero-intensity pixel (pixel in a mask).

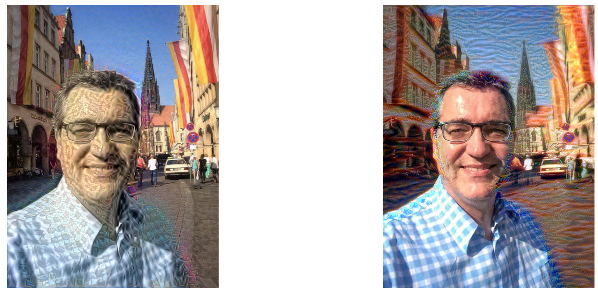

Figure 6: Failure cases of One-image Approach: transfer overlapping

$$\mathcal{L}_{loc}(\vec{p}) = \sum_{i,s} \min_{k} ||P_i^s - P_{0k}^s|| \tag{3}$$

Where P_i^s is the coordinate of a pixel inside a segment s of the content image $\vec{p}$, and P_{0k}^s is the coordinate of a pixel in the corresponding mask image. The idea is that the further to the boundary of different segments, the stronger penalty should apply so as to reduce the repeated loss summation from overlapping of different style transfer. Strictly speaking this $\mathcal{L}_{loc}$ is more like a regularization term than a loss function because no evaluation of its derivative takes place. But for convenience the name works. The computation involves worst case square time in the number of content image pixels, but only takes place once.

The total loss $\mathcal{L}_{total}$ thus changes to:

$$\mathcal{L}_{style} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style} + \gamma \mathcal{L}_{loc} \tag{4}$$

The result images at iteration 0, 20, 40, 60, 80 and 100 are shown below to demonstrate the power of this small but insightful fine-tuning. The issue occurred in the right image of Figure 5 does not show up (both after 100 iterations). Throughout the reconstruction process, the optimization focuses on the area away from the boundaries of two segments.

Figure 7: Results at iteration 0, 20, 40, 60, 80 and 100 using One-image Approach with Locality Loss

3.4 One-image Approach with Localized Style

This approach addresses the second issue mentioned above: uniform distributed style. Strictly speaking, this is not an issue because defining the style as one number for the entire image captures the definition of "style." Style, or genre should be an overall identity of an image; otherwise, a piece of an image can exhibits the same style with a piece of the other image. However, when we have segments from the content image, transferring style from certain areas of an image makes sense. For example, a photo can have a the background looks like the *Starry Night* and the people looks like the person in *The Scream*. In a sense, this extends the creativity of neural style algorithm.

To apply the style locally, this approach computes the style representation $A^{l,s}$ and $G^{l,s}$, takes the derivative of the style loss $E_{l,s}$ at each layer at each segment. With respect to a certain segment s, this derivative

$$\frac{\partial E_l}{\partial F^{l,s}_{i,j}} = \begin{cases} \frac{l}{Z}((F^{l,s})^T(G^{l,s} - A^{l,s}))_{ji} & \mathtt{if} F^l_{i,j} > 0 \\ 0 & \mathtt{if} F^l_{i,j} < 0 \end{cases} \tag{5}$$

Where Z is a normalization factor. The naive straightforward approach also essentially takes care of optimization regarding to each segments, but it does not consider the total loss and also the dependency of the different styles and segments.

Due to the limitation of computation resource, the result of this approach has not been computed.

4 Future Work

The initialization is critical.A large learning rate will lead the algorithm to quickly apply style to the parts where the low level features are the most similar, for example, color and texture but soon gets trapped to local minimum. A slow learning rate will remedy this issue but cause slow computation. Thus, normalization over the low level features can be appealing because these features will be ultimately replaced by the ones from the style image and on the other hand the content representation only focused on the high level deep features.

Also, to compute the loss between representations, i.e. $\mathcal{L}_{content}$ and $\mathcal{L}_{style}$, the original paper uses traditional pixel level 2-norm. Alternatively, a network can be used to compute the similarity by representing the concatenation of the two feature representations with certain non-linearity.

Acknowledgments

Thanks to Alex for this intuitive, problem formulation emphasised course! Thanks to cysmith@Github for his work on the similar topic.

References

References follow the acknowledgments. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. **Remember that you can go over 8 pages as long as the subsequent ones contain *only* cited references.**

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

[2] Thoma, M. A survey of semantic segmentation. *CoRR abs/1602.06541*, 2016.