Various applications for style transfer

Seo Garam 2011130935, Lee Juhyung 2013210112, Park Sunghyun 2013210047
Department of Computer Science
Korea University
skekrkfka@gmail.com, ejh7406@naver.com, psh01087@naver.com

Abstract

This project aims to extend the technique of single style transfer to multi style transfer and apply it various way with mask images. It makes the user transfer the object in the content into desired style image. The user can change the style except for color, or import color and style from different style images. At first, we implemented single-style transfer and transformation network is connected to pre-trained VGG19 network. After training on a desired style, we can input any desired image and render it to new visual genre. Second, we will extend style transfer to multiple styles. Third, by using a single-style transfer generated image and mask image, we applied style transfer to only part of the image. After that, we applied the style except the color, or the color and the style were taken from another image and applied. This research shows the various application of style transfer with mask.

1 Introduction

A pastiche is something such as a piece of art in which the style is copied from somewhere else, or contains a mixture of different styles. Along with computer vision and machine learning has long history to make automatic pastiche.
In 2015, Gatys et al [1] use Neural Networks for style transfer at the first time. Taking the contents of an image and make it into 'style' of another image is the technique of it. The 'style' includes features such as the texture, color scheme, lighting etc. This leads to the result which is visually astonishing and another research improving its accuracy, efficient, and customizable.
In 2016, Gatys et al [1]. published the paper about "Image Style Transfer Using Convolutional Neural Networks." This can make a automatic pastiche with single style. This followed by "Multi-style Generative Network for Real-time Transfer" by Han Zhang & Kristin Dana [2]. This papers shows that multiple style can be applied to make aggregated style image.
With technological advancement, there is much attention for how to apply the style transfer. For example, 'Vincent AI' is the application that helps the user to draw the art such as British landscape paintings J.M.W. Turner's oil paintings or neon-colored pop arts in realtime.
As a result, we are interested in how to apply style transfer in various way with other application and we want to study various application about it.

2 Problem Definition

Conventional methods could create an aggregation by combining styles imported from multiple images. However, It did not apply the style to the sesired part only, or it can not exclude the color when applying the style.
We think that various applications beside simply applying style to images. By this, users would be able to apply the desired combination of style to the desired part according to what they need. For example, we could apply the style of the style image while maintaining the color, or import the color

1st Special Lecture For Computer Science-II (TERM 2017), Seoul, South of Korea.

and style from different images.
To apply this, we need to implement single-style transfer at first which is a feed-forward convolutional neural network. This transformation network is connected to pre-trained VGG19 network. After training on a desired style (or combination of them), we can input any image and make it to new visual genre. Second, we extend single style transfer to multiple styles by training the network to learn new parameter that will mix the weights. Third, by using a single-style transfer generated image and mask image, we applied style transfer to only part of the image. After that we applied the style except the color, or the color and the style were taken from another image and applied.

3 Models

3.1 Style Transfer

3.1.1 Content Loss

Let $\vec{p}$ and $\vec{x}$ be the original image and the image that is generated, and P^l and F^l their respective feature representation in layer l. The content loss is the Mean Squared Error of the feature activations in the given content layers(l) in the model, between the content mixed image and source image. The content loss can be calculated as:

$$L_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} (F^l_{i,j} - P^l_{i,j})^2 \tag{1}$$

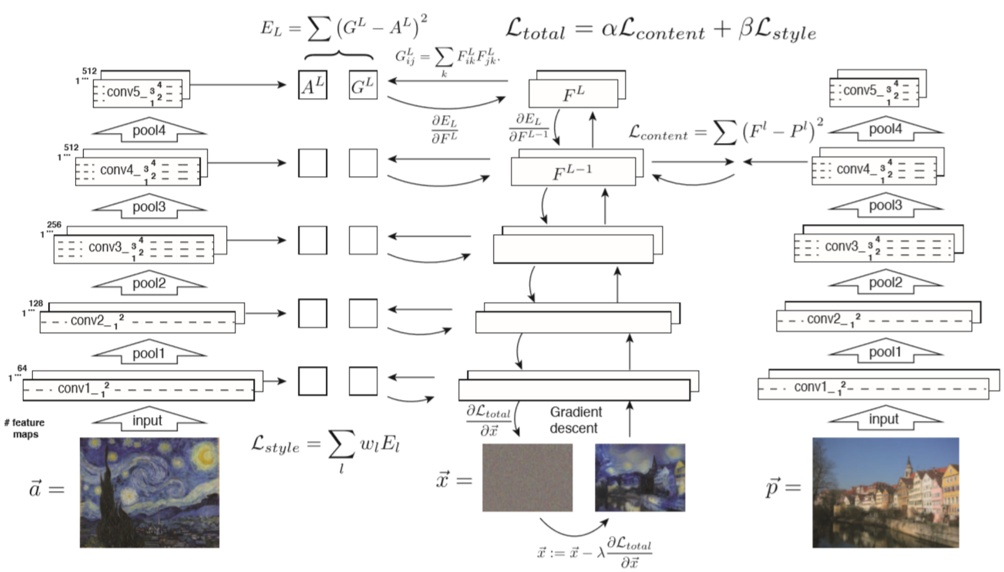


Figure 1: Style Transfer Algorithm [1]

3.1.2 Style Loss

To obtain a representation of the style of an style image, we use a feature space designed to capture texture information. This feature space can be built on top of the filter responses in any layer of the network. It consists of the correlations between the different filter responses. These feature correlations are given by the Gram matrix $G^l \in R^{N_l \times N_l}$, where $G^l_{i,j}$ is the inner product between the vectorized feature maps i and j in layer l:

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk} \tag{2}$$

Let $\vec{a}$ and $\vec{x}$ be the original image and the image that is generated, and A^l and G^l their respective style representation in layer l. The contribution of layer l to the total loss is then:

$$E_l = \frac{1}{4N_l^2M_l^2}\sum_{i,j}(G_{ij}^l - A_{ij}^l)^2 \tag{3}$$

and the total style loss is:

$$L_{style}(\vec{a},\vec{x}) = \sum_{l=0}^{L} w_l E_l \tag{4}$$

where w_l are weighting factors of the contribution of each layer to the total loss.

3.1.3 Single-Style Transfer

To transfer the style image $\vec{a}$ onto content image $\vec{p}$, we synthesis a new image that simultaneously matches the content representation of $\vec{p}$ and the style representation of $\vec{a}$ (Fig 1). Thus we jointly minimize the distance of the feature map of input_image from the content representation of the content image in one layer and the style representation of the style image on a number of layers of the CNN. The loss function is:

$$L_{total}(\vec{p},\vec{a},\vec{x}) = \alpha L_{content}(\vec{p},\vec{x}) + \beta L_{style}(\vec{a},\vec{x})q \tag{5}$$

α and β are the content weight and the style weight.

3.2 Multi Style Transfer

Weighted average of all the input style loss for blending different styles. For example, when the number of style images is 2, the total loss function of multi-style transfer is:

$$L_{total}(\vec{p},\vec{a},\vec{b},\vec{x}) = \alpha L_{content}(\vec{p},\vec{x}) + \frac{1}{2}\beta(L_{style}(\vec{a},\vec{x}) + L_{style}(\vec{b},\vec{x})) \tag{6}$$

3.3 Masked Style Transfer

We implemented Masked-Style Transfer which is able to be used with Semantic Segmentation such as FCN, DeepLab, CRF-RNN etc. By using generated image from single style transfer and mask image, we achieved visually pleasing partial-image style transfer. First performing whole-image style transfer on the target image. Then if the RGB value of the mask image pixels is 0, the pixels of the generated image corresponding to those pixels are transferred on to the non-stylized image. In this way, the selection of specific regions in the original image are altered, while the rest of the original image maintains its appearance.

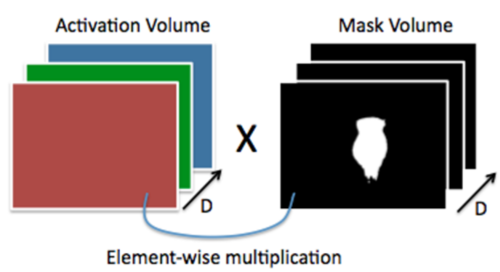


Figure 2: Masked Style Transfer [3]

3.4 Color-off Style Transfer

Using the style image and the content image, we perform style transfer on the content image. Then we transform the color of the generated image(G) with content image. In other words, neural style transfer is computed from the original inputs, and then the output G is color-matched to the content image, producing a new output G'. There are many color transformation algorithms to choose, but we use linear methods, which are simple and very effective for color style transfer. In particular, let $x_i = (R, G, B)^T$ be a pixel of an image. Each pixel is transformed as:

$$X_{G'} \leftarrow AX_G + b \tag{7}$$

where A is a 3X3 matrix and b is a 3-vector.

3.5 Color & Style Style Transfer

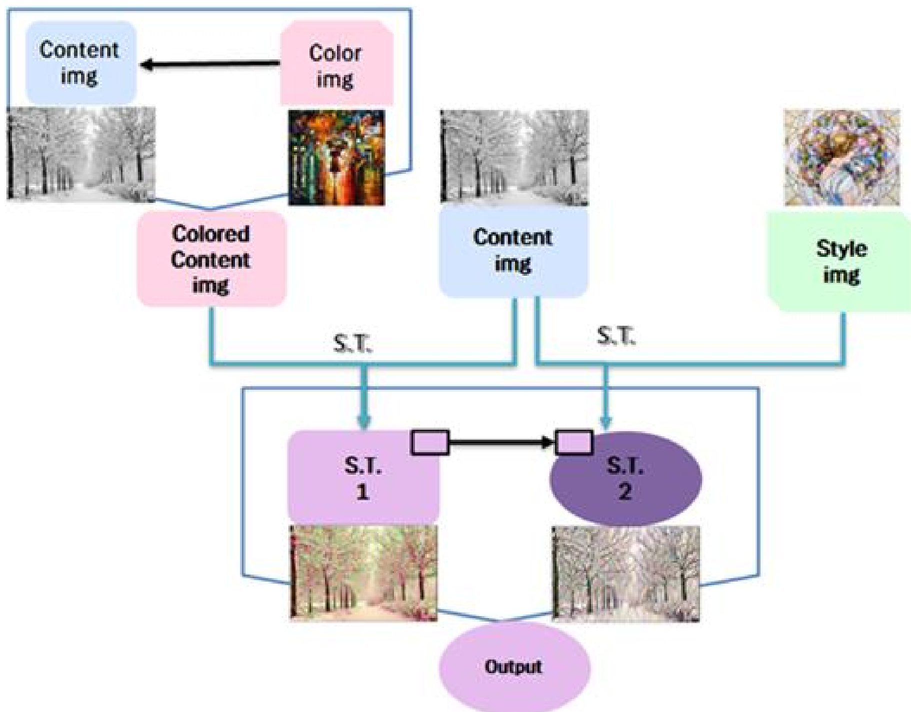


Figure 3: Color&Style Style Transfer Algorithm

Color & Style Transfer is to bring color only in one image and style in another image. First, to bring color only in one image, we apply the color of the style image to the content image, then we perform Style Transfer to this colored content image and content image. And we generate another image from other style image and content image. Then we perform Style Transfer to this colored content image and content image and generate new image(S.T.1). And we generate another image(S.T.2) from other style image and content image. Then we transform the color of the image(S.T.2) with the image(S.T.1).

4 Dataset

We will choose different type of datasets to test and validate our applications :

Data Set for Transfer learning
- pre-trained VGG-19 model weights for loss network

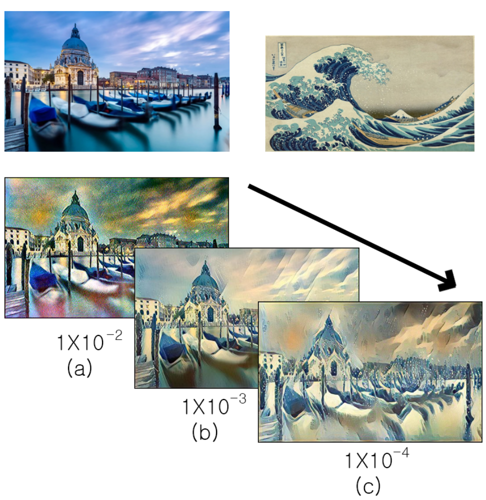
Data Set for Content image
- A wide variety of 30 photos, including portraits, landscape, etc
- 7 Content images which have a masked images from Google image

Data set for Style image
- 20 Images of artist of unique style
- 20 Images which have unique style

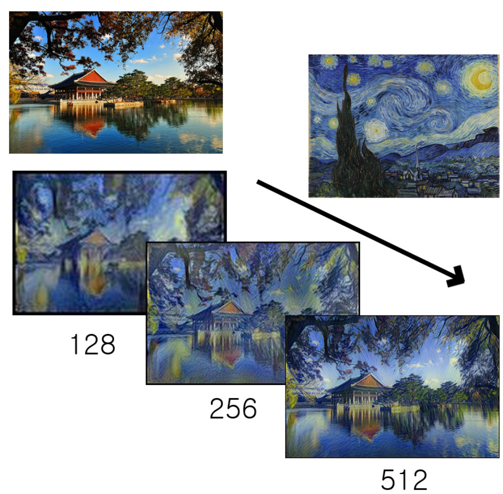
5 Experiment and Result

5.1 Single Style Transfer

Figure 4: Results of Single Style Transfer with various style images



(a) Changing the weights



(b) Changing the size of the images

Figure 5: Single Style Transfer

(a) Relative weighting of matching content and style of the respective source images. The ratio α/β between matching the content and the style increases from left to right. (b) Relative size of content image. The size of the image increases from left to right.

Above we present out results for single-style transfer using the different style images and the different weights of style image and content image and the different size of the generated image. First Fig.4 is the output images which are generated from the various style images. And Fig.5 is the generated images by changing the weights($\alpha \quad and \quad \beta$) of style image and content image. The ratio of α/β is either 1×10^{-2}, 1×10^{-3}, 1×10^{-4} . In $\alpha/\beta = 1 \times 10^{-2}$, the style of the painting is not as well-matched and in $\alpha/\beta = 1 \times 10^{-4}$, a strong emphasis on style result in images that match the appearance of the style image, but show hardly any of the content image. And Fig.5-b is the generated images by changing the size of the images. Resizing style image before running style transfer algorithm can transfer different types of features.

5.2 Multi Style Transfer

Below we present our results for multi-style transfer using two style images. The total loss of multi style transfer using two style images is:

$$L_{total}(\vec{p},\vec{a},\vec{b},\vec{x}) = \alpha L_{content}(\vec{p},\vec{x}) + \beta_1 L_{style1}(\vec{a},\vec{x}) + \beta_2 L_{style2}(\vec{b},\vec{x}) \tag{8}$$

Figure 6: Result of Multi Style Transfer. The ratio β_1/β_2 between matching the style1 and style2 decreases left to right.

5.3 Masked Style Transfer

5.3.1 Masked Style Transfer

Below we present our results for masked-style transfer using the mask images composed of white and black only.

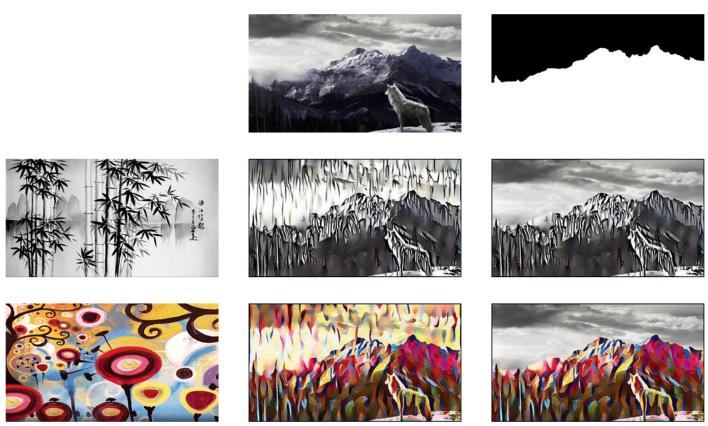


Figure 7: Results of Masked Style Transfer

5.3.2 Using PASCAL VOC 2012 Dataset

Below we present our results for masked-style transfer using the mask images and content images from PASCAL VOC 2012 Dataset. Originally, we were planned to implement the object detection and use the PASCAL VOC 2012 Dataset to train. However, we failed to implement the object detection such as DeepLab, FCN due to a memory shortage on the GPU. So, we only show the application of Style transfer to this dataset.

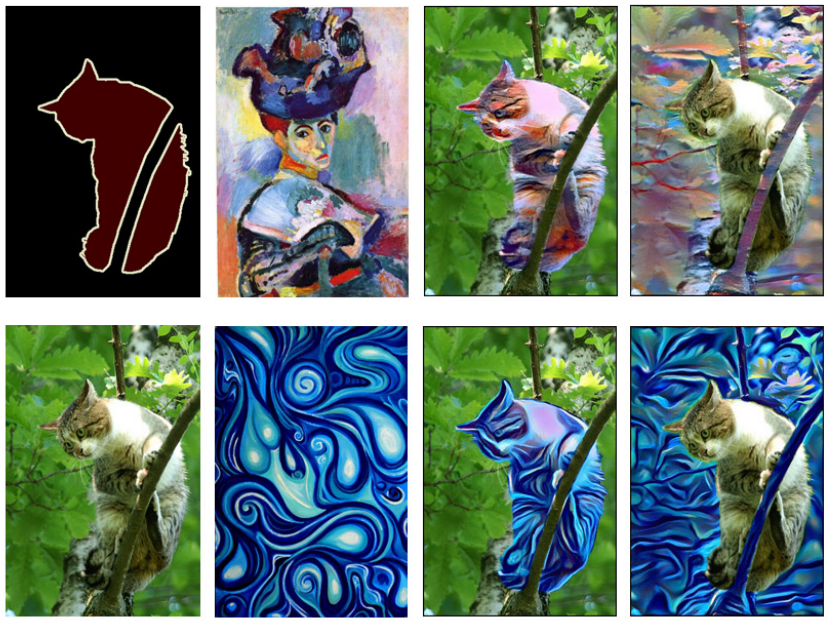


Figure 8: Results of Masked Style Transfer with PASCAL VOC 2012 Dataset

5.4 Color-off Style Transfer

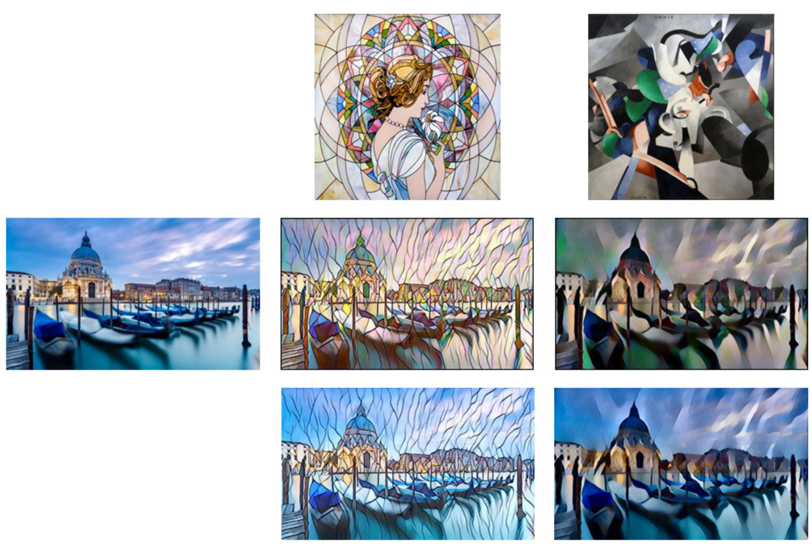


Figure 9: Results of Color-off Style Transfer

We want to improve the point where color changes along the style image when we perform style transfer. We want to keep color and make style transfer which only change style. So we implement Color-off Style Transfer. Below we present our results for color-off style transfer.

5.5 Color & Style Style Transfer

We tried to create a new Style Transfer Network by applying Color-off Style Transfer and color transfer technology. So we implemented a style image that only imports a style and the other style image that imports only color. After implementing this network, we tried many times with various style images, but we found out that style image, which only imports colors, is not effective when the colors do not vary. So we used color image, which has various colors, to produce the results. Below we present our results for color & style transfer.

Figure 10: Result of Color & Style Transfer

6 Conclusion & Future work

Recent research has focused on applying the style that was drawn from painter's work to existing images. However, we have not only applied the style that was taken from a single work to a photograph, we painted it to multi style, and focused on various applications using masks.

We think we have succeeded in expanding the applicability of style transfer using various combinations with existing methods. And traditional transfers were only able to apply to style images to the entire content image.

However, we succeeded in using the mask image to put the part of content image into the desired style image. In addition, we applied style transfer to apply styles other than colors. After that, we succeeded in applying the color and style to the content image in the two style images respectively.

It is meaningful to confirm that it is possible to combine a style image by taking only the desired attributes from several images, rather than a single style image, by applying a style transfer.

We think that it is possible to apply this technique to create a photo that does not actually exist through a combination of various styles. For example, if you want to create a picture of the previous time with the current picture, you can create a more realistic picture by bringing in the desired attributes from multiple pictures as well as simply the style of the desired year.

For example, there is a technique called 'Deep Photo style transfer' [7], in which a distortion occurs when transferring a photorealistic image, resulting in a phenomenon that is not photorealistic. To improve this, It constrain the transformation from the input to the output to be locally affine in colorspace. So, our research could be related to this kind of approach.

In future work, we could advanced our approach based on 'Deep Photo Style Transfer' [4]. Apply style with parameter adjusting way, we would improve the photo style transfer technique.

References

[1] Bethge M Gatys LA, Ecker AS. Image style transfer using convolutional neural networks. *IEEE*, 2016. 1, 1

[2] Hang Zhang and Kristian Dana. Multi-style Generative Network for Real-time Transfer. *arXiv*, 2017. 1

[3] Alex et al. Localized Style Transfer. 2017. 2

[4] Shechtman E Bala K Luan F, Paris S. Deep Photo Style Transfer. *arXiv*, 2017. 6