Published in Towards Data Science

Paul Froehling

Dec 9, 2021 · 7 min read · Listen

HANDS-ON TUTORIALS

Mixed Neural Style Transfer With Two Style Images

An Approach to Extend Neural Style Transfer to Include Two Styles

Fig. 1: Result of my implementation of the mixed neural style transfer (MNST) based on one content image and two style images. (Content image by Alexas_Fotos, style image 1 by Andreas Fickl, style image 2 by Europeana)

Paul Froehling
11 Followers

Introduction

As somebody who is both, an technology and art enthusiast, neural style transfer (NST) is super fascinating to me: An algorithm that creates a painting by using a content and a style template. While implementing and experimenting with the original NST algorithm, I had the idea of combining two styles in one image. Thus, the topic of my story is about a variation of the original neural style transfer, an approach that was described in the paper "A Neural Algorithm of Artistic Style" by Gatys et al. The mixed neural style transfer (MNST) describes an extension of the original algorithm, using two style images and one content image.

In the course of the story, I'll explain how to apply the styles of two images to one photography, analyze the improvement process and show how to extend the NST

optimization process by weighting the given styles based on their individual losses.

Since there are some very good tutorials and explanations to neural style transfer, I would like to leave the introduction to them and rather continue with mixed neural style transfer. (Check the out the Neural Style Transfer Tutorial of Vamshik Shetty, giving a great introduction to gram matrices. This TensorFlow tutorial also could be a good starting point for implementation.)

From Neural Style Transfer to Mixed Neural Style Transfer

The loss function as the central element of NST covers two aspects: content loss and style loss. While the content loss guarantees, that the newly generated image doesn't diverge too far from the subject of the content image, the style loss measures a stylistic difference to a

style image. When implementing the original approach of NST, I was wondering what happens if I extend the loss function of NST by a second style loss to generate a mix between two style images and one content image.

$$L_{NST}(c, s, x) = \alpha L_{content}(c, x) + \beta L_{Style}(s, x)$$
$$L_{MNST}(c, s_1, s_2, x) = \alpha L_{content}(c, x) + \beta(\gamma L_{Style}(s_1, x) + (1 - \gamma) L_{Style}(s_2, x))$$

Fig. 2: α and β are weighting the content and the style loss. γ is weighting style loss 1 and style loss 2 individually, while γ should take values in the range between 0 and 1. C denotes the content image, s the style image for NST and s1 and s2 the style images for MNST. Image by author.

In the original loss function of the NST, α and β are the weights for the content loss and the style loss, respectively. For the MNST, α and β are kept the same, but γ is introduced as a second style weight. It is used to define the degree, to which an individual style influences the overall style loss. If e.g. γ ist set to 0.7, the loss of style image 1 impacts the overall style loss more than the loss of style image 2, being weighted with

0.3. By shifting values between 0 and 1 for γ, it is possible to interpolate between the results of a NST for two style images. So γ should basically works like a crossfader for the styles.

For the following example these images were chosen as content image, style image 1 and style image 2:

Fig. 3: The images are loaded from unsplash.com: Style image 1 by Andreas Fickl, style image 2 by Europeana, content image by by Alexas_Fotos

The Setup

- *Model:* Pre-trained VGG19 model with ImageNet weights
- *Content Output Layer*: block5_conv2
- *Style Output Layers:* block1_conv1, block2_conv1,

block3_conv1, block4_conv1, block5_conv1

- *Layer Weights:* Additionally I configured the weightings of the layer outputs for the style loss, so that higher convolutional layers get a comparably lower weighting than lower layers. The intuition behind this weighting is to give a lower impact to filters, emphasizing complex structures and rather focus on simpler structures, defining the style of an image. [block1_conv1: 1.0, block2_conv1: 1.0, block3_conv1: 0.2, block4_conv1: 0.3, block5_conv1: 0.01]
- *Gram Matrices:* The calculation of the gram matrices has been slightly changed, since their values are not divided by the output dimensions.
- $a = 0.002$ and $\beta = 0.02$
- *Optimizer:* Adam optimizer with initial_learning_rate=12.0, decay_steps=100 and

decay_rate=0.6.

- *Loss*: tf.reduce_mean was used for computing the style and content loss
- *Starting Conditions:* The optimization starts with a duplicate of the content image.

Fig. 4: MNST results for different values of γ, based on the content and style images shown in fig. 3. (Content image by Alexas_Fotos, style image 1 by Andreas Fickl, style image 2 by Europeana)

Assessment of the Results

When comparing the style mixes for different values of γ, it can be

seen clearly how the style of style image 1 increasingly impacts the overall result. While the washed-out style of style image 2 is predominant in the generated image for values of γ close to 0, the squiggles of the graffiti text in style image 1 appear more clearly the further γ is shifted towards 1.

Even though, mixing styles works quite well, the difference between the results for $\gamma \geq 0.5$ is not visible at first sight. While low values of γ clearly impact the overall result, for values greater than 0.5, it required good eyes to see any changes. One explanation could be the colorful, lively style of the graffiti, that makes subtle changes hard to see.

A different explanation results from observing the loss. For the balanced scenario of $\gamma = 0.5$, style loss 1 is 3.6 times as big as style loss 2 at the beginning of the optimization process. At the end of

the optimization process, the relative improvement (start loss/final loss) for style loss 1 is nearly four times as big as for style loss 2. Since my implementation of the style transfer starts optimizing a copy of the content image, the style loss in the first iteration is the style loss between the content image and the style images. Thus, $x=c$ in the first iteration of the optimization process.

Calibrating the Loss for MNST Starting from the Content Image

For $\gamma=0.5$, Style 1 and Style 2 should influence the final result equally strongly. Therefore, it should be avoided that one style loss is optimized while the other remains the same or even increases. Finding the right method for this required some tinkering. My intuition is that for $\gamma=0.5$ both style losses should make the same relative progress. If style loss 1 improves by 50%

(compared to its starting loss), style loss 2 should also improve by 50%.

The relative improvement is computed as follows:

$$RI_1 = L_{Style}(s_1, c)/L_{Style}(s_1, gen)$$
$$RI_2 = L_{Style}(s_2, c)/L_{Style}(s_2, gen)$$

Fig. 5: RI denotes the relative improvement for style loss 1 and 2 during the optimization process. It compares the initial loss of the first iteration against the loss after the last iteration. While c stands for the content image, gen is the generated image after the last iteration.

To get an impression how RI1 and RI2 develops for different values of γ, I let the MNST run for 100 iterations to calculate the relative improvements for values from γ=0 to γ=1 in steps of 0.1. The ratio between RI1 and RI2 was used, to find out how much more style loss 1 was improved compared to style loss 2.

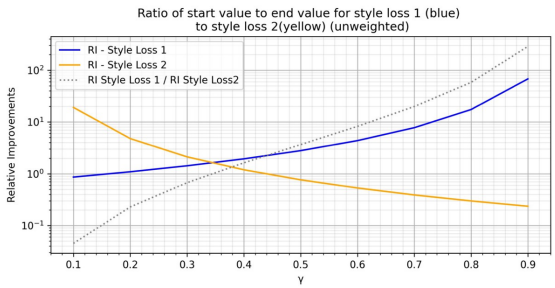

Fig. 6: Observation of RI1 and RI2 for different values of γ shows that style loss 1 and 2 have the same relative improvement for γ~0.34 and not as intended for 0.5. Image by author.

It shows, that RI1 and RI2 are the same for $\gamma \sim 0.34$, while style loss 1 improves better for values of $\gamma \geq 0.34$. This supports the previous impression from the visual inspection of Fig.4, that the impacts of the two styles are not equal for $\gamma = 0.5$.

To shift the attention a bit more to the “weaker” style, I initially used the ratio between the relative improvements (RI1/RI2) of the style losses for $\gamma = 0.5$ as an additional weight, called σ, for style loss 2. But now the result overshoots the value of 0.5 since the styles develop equally good for $\gamma \sim 0.64$. After checking the impact of different values of σ, it turned

out that a change of σ does not impact the shift of RI1/RI2 linearly but by the square root.

Thus, the overall loss function now looks the following:

$$L_{MNST}(c, s_1, s_2, x) = \alpha \cdot L_{content}(c, x) + \beta \cdot (\gamma \cdot \sigma_1 \cdot L_{Style}(s_1, x) + (1-\gamma) \cdot \sigma_2 \cdot L_{Style}(s_2, x))$$

$$\sigma_1 = \begin{cases} 1 & if L_{Style}(s_1, c)/L_{Style}(s_1, gen) >= L_{Style}(s_2, c)/L_{Style}(s_2, gen) \\ \sqrt{\frac{L_{Style}(s_2,c)/L_{Style}(s_2,gen)}{L_{Style}(s_1,c)/L_{Style}(s_1,gen)}} & otherwise \end{cases}$$

$$\sigma_2 = \begin{cases} 1 & if L_{Style}(s_1, c)/L_{Style}(s_1, gen) <= L_{Style}(s_2, c)/L_{Style}(s_2, gen) \\ \sqrt{\frac{L_{Style}(s_1,c)/L_{Style}(s_1,gen)}{L_{Style}(s_2,c)/L_{Style}(s_2,gen)}} & otherwise \end{cases}$$

Fig. 7: Additionally to γ σ1 and σ2 are introduced, to balance the loss improvement. Image by author.

By using the new weights called σ1 and σ2, the relative improvements for the two style losses are equal for γ=0.5 (as shown in fig. 8). When observing the grey dotted line it can also be seen, that the ratio between the relative improvements develops symmetrically around γ=0.5.

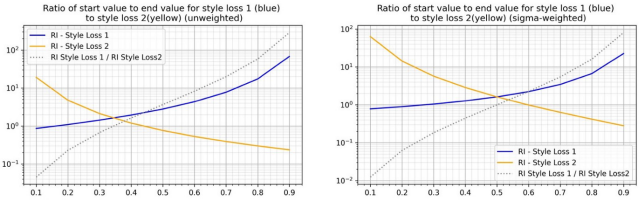

Fig. 8: Left: Ratios of start to end value for style loss 1 and 2 for different values of γ after 100 iterations without sigma weights. Right: Same

scenario with sigma weights. Image by author.

Comparing the results of the MNST with and without σ1 and σ2 (fig. 4 vs. fig. 9), it can be seen that the style of style image 1, fades in continuously and does not dominate the resulting image from γ=0.5 on.

Fig. 9: Results for MNST for different values of γ with σ1=1 and σ2~1.9, based on the content and style images shown in fig. 3. (Content image by Alexas_Fotos, style image 1 by Andreas Fickl, style image 2 by Europeana)

Conclusion

While trying out different style and content images, I realized, that MNST leads to nice results, when style images are different concerning their structure and coloring. If in contrast two style images are chosen that are both super colorful and similar in their textures, results are often not very interesting in the sense, that each style contributes uniquely to the overall result. It's also worth thinking about which style could emphasize which parts of the content image to increase the expressiveness of the newly generated graphic.

Finally, I would like to show you two more examples of MNST, which were created during my experiments.

Enjoy :)

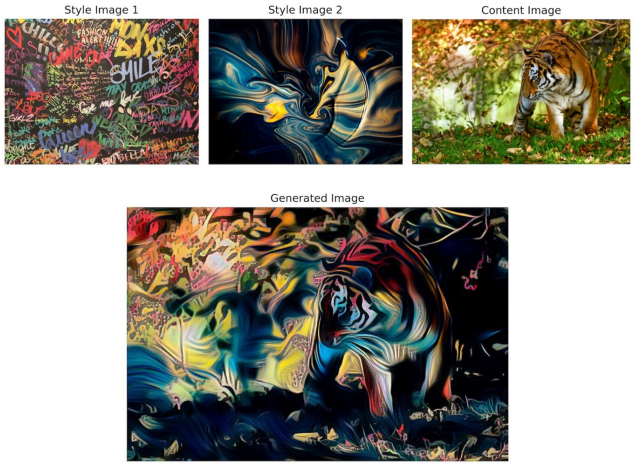

Fig. 10: γ = 0.4, σ1 = 1, σ2 ~ 1.503 || style image 1 by Andreas Fickl, style image 2 by Dan-Cristian Pădureț, content image by Alexas_Fotos. All references are loaded from unsplash.com

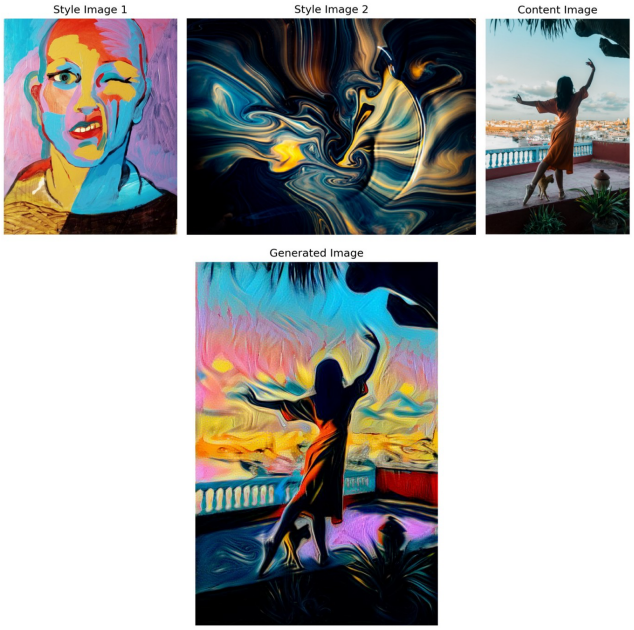

Fig. 11: γ = 0.5, σ1 = 1, σ2 ~ 1.150 || style image 1 by Nicola POWYS, style image 2 by Dan-Cristian Pădureț, content image by mehdi lamaaffar. All references are loaded from unsplash.com

Thanks to Elliot Gunn