Deep Learning for Perception – Final Project

A Smoother Segmentation-Based Style Transfer – With Multiple Styles

Netanel Tamir – 204900062

A. Introduction to the Learning Problem:

Neural Style Transfer was first introduced in the 2015 paper "*A Neural Algorithm of Artistic Style*" [1] which introduced a way to combine the content of one image, with the style of another. Since then, there have been multiple attempts to branch out and test the boundaries. In 2016, a Stanford project named "*Show, Divide and Neural: Weighted Style Transfer*" [2], tackled the task of applying different styles to different segments of the content image. In this project, the authors managed to combine 2 styles into one image. Their work is impressive however it can be improved upon. The following image was their best result:

*Figure 1.Best result in "*Show, Divide and Neural: Weighted Style Transfer*", Combined Gram Matrix Manipulation and Capped Gradients*

Specifically, I will attempt to create a smoother output image with 3 styles (one per segment) each segment can have multiple instances – doesn't change the algorithms. The main challenges involve designing a learning architecture that will be able to create a smooth image which is not a simple task considering the 3 distinct style images I will use.

B. Previous work on this issue:

In the Stanford paper mentioned above [2], the authors' methodology consisted of segmenting the image, followed by two techniques: 1) Capped Gradients – Splitting the learning process into two. 2) Gram Matrix manipulation – Manipulation of the gram matrix using the mask created in the segmentation process.

However, these techniques are not explained clearly enough to duplicate and so I will create my own techniques based on the core ideas taken from the aforementioned paper that will also suit the problem I am undertaking.

C. The methodology

The Loss:

I will first begin with introducing the loss used. In Attempts 1 and 2, because the learning process is divided into several parts, so is the loss. In Attempt 3 however, I use a single loss function. Each of these loss functions is some combination of the content loss, style loss of a certain style image, and total variation loss – an addition to the vanilla style transfer loss, used to smooth the output image.

Attempt 1:

On my first attempt I began with segmenting the content image into cat/dog/background.

Figure 2. Content image used featuring a dog and a cat

Figure 3. Segmentation of the content image

Afterwards I used the following architecture to add different styles to each segment.

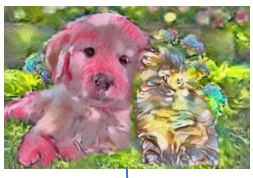
Content Image

(1) Vanilla Style Transfer with Style Image 1.

$$\mathcal{L} = \alpha \times Loss_{Content} + \beta \times Loss_{Style\_1}$$

(2) Vanilla Style Transfer with Style Image 2/3. Copy only pixels of cat/dog.

$$\mathcal{L} = \alpha \times Loss_{Content} + \beta \times Loss_{Style2/3}$$



(3) Smoothing Image using TV loss

$$\mathcal{L} = \gamma \times Loss_{TV}$$

Figure 4. Style Image 1 - A watercolor painting of The Shire from LOTR

Figure 5. Style Image 2 - A stained glass mosaic

Figure 6. Style Image 3 - A painting of cherry blossoms

If we take a look at an enlarged version of the output image, we can see that indeed each of the styles was applied as intended however the picture isn't very smooth. Increasing the number of epochs of the smoothing phase begins to create a haze effect on the image.

Figure 7. The output of Attempt 1

In my next attempt I will try to find a way to smooth the image, in a way that feels more natural.

<u>Attempt 2:</u>

Here, I used 2 techniques to improve the smoothness of the output image. First, during the smoothing part of the architecture described previously, I also added content loss – to keep the content clear, and style loss – with the background style image, to better blend the parts of the image together.

New Loss function for stage 3:

$$\mathcal{L} = \alpha \times Loss_{Content} + \beta \times Loss_{Style\ 1} + \gamma \times Loss_{TV}$$

Secondly, I smoothed the image again, this time only with TV loss and only on the edges between the different segments, to blur the boundaries better. The TV loss is calculated on an element-wise product between the blur and the target image. The blur is maximized on the border and decreases according to the distance from it.

Loss function for stage 4

$$\mathcal{L} = \gamma \times Loss_{TV}$$

Figure 8. Blur image created for the content image

The final result is as follows:

Figure 9. The output image of Attempt 2

We can see that there is an improvement over the output of Attempt 1. This output seems less "artificial" than its predecessor.

Attempt 3 – Single Shot Segmented Style Transfer:

Here I will attempt to transfer the multiple styles to the content image with only one learning process. This will result in a much shorter learning time.

The loss for the learning process here will be the sum of the content loss with the content image, style loss of a surrogate style image, and TV loss. Additionally, I added the content loss using the surrogate style image to the loss. This helps add the correct style to the certain segment, since just regular style loss creates a mixture of the style image.

$$\mathcal{L} = \alpha \times Loss_{Content} + \beta \times (Loss_{Style} + \delta \times Loss_{Content\ with\ style\ image}) + \gamma \times Loss_{TV}$$

Note: I changed the style image 2 in this part to the following picture to be able to better distinguish between the different styles in the output image.

Figure 10. New style image 2 for Attempt 3

I began with creating the surrogate style image by combining the different style images using the segmentation created previously. The result is shown below.

Figure 11. Surrogate style image for Attempt 3, Created using the segmentation and the 3 style images

Afterwards I trained the model similarly to Vanilla Style Transfer, with the only change being the loss mentioned earlier. In this case the Content Image is the same one used in previous attempts (Figure 2) and the Style Image is the surrogate style image (Figure 11).

The output image is shown below.

Figure 12. Output image of Attempt 3

Here we can see that the segments indeed received their correct styles. Furthermore, we can also see that the styles from the style images were transferred and not the actual style image, which is a concern if the content loss on the surrogate style image is weighted too high.

D. Results

We will now discuss and compare the results of Attempt 2 and Attempt 3, using the same style images.

Figure 13. Output of Attempt 2 using the same style images

Similarly to the Stanford paper [2], the only way to truly evaluate the results is by visual inspection and not loss values, and judge whether or not the outputs achieved our stated goal or not. Furthermore, throughout this paper I haven't mentioned hyperparameter values, this is because I attempted to optimize the hyperparameters per learning process and different images needed different hyper parameter values to produce a quality output. My search for hyperparameters was limited due to the run time and a more extensive search would probably result in hyper parameters that would've produced better looking output images.

Firstly, both methods succeeded in creating an output image that applies a different style to each segment.

Secondly, both methods managed to transfer the content of the content image to the output image, which was not obvious that that would be the case, considering the learning processes utilized instead of a Vanilla Style Transfer.

Thirdly, it appears that neither attempt manages to successfully merge the styles to create one coherent style that transforms between different segments. Even though the variation between styles is more obscure than the image created in Attempt 1 without the smoothing process, there is still work to be done. Particularly the colors that could benefit from less variations.

Interestingly, the two outputs have a different style to them. The output of Attempt 2 has a more natural, water color look. However the output of Attempt 3 has a more mosaic style to it.

It appears that my output images (especially figures 9, 12) are smoother and more precise than the one in the Stanford paper [2] (Figure 1).

Next we take a look at the loss values in the learning process of Attempt 3 (SSSST) and the output image as a function of the number of epochs in order to better understand how it works.

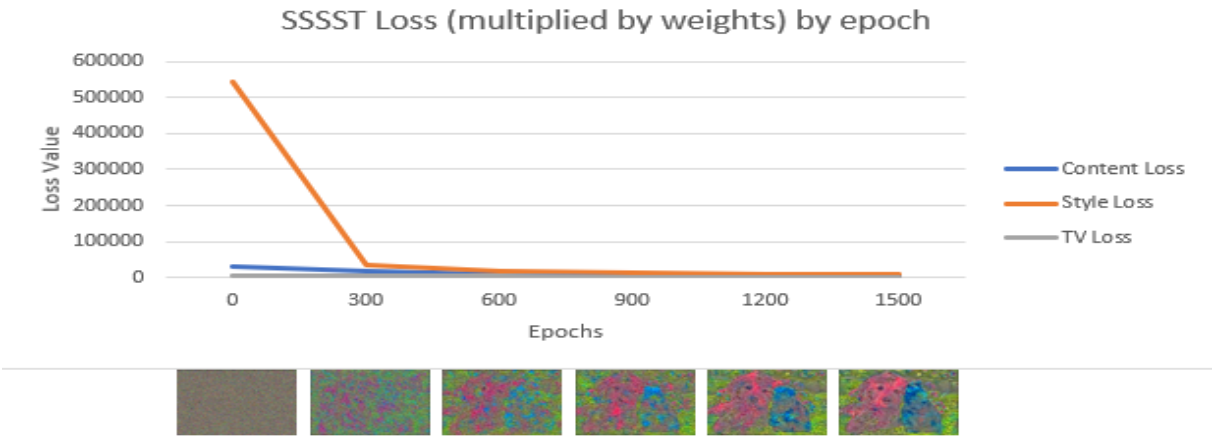
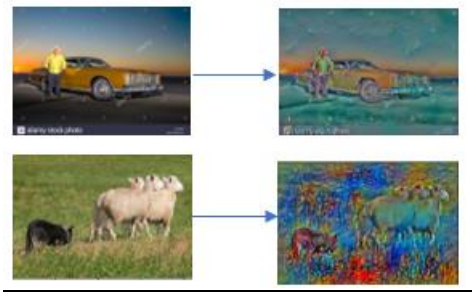


Figure 14. Graph of loss in Attempt 3 by epoch

In the graph above we can see the loss as a function of the number of epochs in Attempt 3. More epochs result in diminishing losses and even more result in a visually worse output image. We can see that the style initially looks scattered and a mix of the different styles in the surrogate style image, but as the number of epochs increases, the content and each segment's matching style appear.

The reason that each segment ends up with the correct style is due to the content loss used on the surrogate style image. Too little of it and we get a mixture of style like after 300 epochs. Too much and we get the actual content of the surrogate style image and that too is undesirable. However, if we can manage to calibrate it to a suitable level, like we've done here, then we get the style of the style images without the actual content of them.

By altering the code slightly, we can use other content and style images to create other output images and prove that our algorithm is versatile. For example (using the Attempt 3 method):



GitHub Link:

https://github.com/Netanel-Tamir/SegmentedStyleTransfer

References:

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

https://arxiv.org/abs/1508.06576

[2] E. Chan, R. Bhargava. *Show, Divide and Neural: Weighted Style Transfer.* Stanford University, 2016.

http://cs231n.stanford.edu/reports/2016/pdfs/208_Report.pdf