# CS230: Segmented Neural Style Transfer for Stylized Chair Generation

**Sam Premutico**
Computer Science
Stanford University
Stanford, CA 94305
samprem@stanford.edu

**github:** https://github.com/sampremutico/neural-style-tf

## 1 Introduction

Neural style transfer has been able to produce aesthetically pleasing, novel artwork that transplants the style of one image onto the content of another. Using the technique, we are able to generate self-portraits in the style of *da Vinci's* Mona-Lisa or the New York City skyline in the style of Edvard Munch's *The Scream*. While the images generated with neural style transfer generate novel, aesthetically pleasing images, typical models extract the *entire* content of the content image and the *entire* style of the style image. In our project, we attempt to augment neural style transfer techniques so as isolate a portion of the content image and style it in the style of a portion of the style image. Specifically, we apply this augmented neural style transfer technique to generate images of *a chair* styled in the style of another chair. This allows us to generate, for example, an Ikea Poang chair in the style of a classic Eames Chair. If succesfull, this technique allows us to quickly generate novel, cross-generational chair designs. This has applications to a variety of industries that make use of collaborative design, including furniture, fashion, and architecture.

Chair design specifically benefits from localized style transfer due to the fact that when we feed an image of a chair into our model, we want as output *the chair* in the style of the style image, not the chair and the background content of the image. Additionally, we don't want the output to be styled in the style of the background of the style-chair, but rather we want it to be styled in the style of the *chair itself*. Using generic neural style transfer, we end up styling the content of the entire content image, background and all, in the style of the entire style image, background and all. In our work, we build off of previous work on *segmented* neural style transfer that allows the styling of a portion of the content image in the style of a portion of the style image. We specifically modify our style loss function as well as the layers at which we measure style loss in order to produce the most reasonable chair designs.

## 2 Related Work

### 2.1 A Neural Algorithm for Style Transfer

In *A Neural Algorithm for Style Transfer*, Gatys et al. propose a method for generating an image with the content of one image and the style of another. To do so, the authors propose starting with a randomly initialized image and, using a CNN, iteratively updating the pixel values to minimize the difference between the content of the content image (content loss) and the style of the style image (style loss). The content and style losses at layer $l$ are as follows

$$L_{content} = \frac{1}{N_l M_l} \sum_{ij} (X_l - C_l)^2_{ij}$$

$$L_{style} = \frac{1}{4N_l^2} \sum_{ij} (G(X_l) - G(S_l))_{ij}^2$$

where $X_l$ is the generated image at layer $l$, $C_l$ the content image at layer $l$, $G(X_l)$ the Gram matrix of the generate image at layer $l$ and $G(S_l)$ the Gram matrix of the style image at layer $l$. At each layer, we define $F_l(A)$ as the vectorized feature map of image $A$ at layer $l$, with each column being being a feature map. $N_l$ is then the number of feature maps at layer $l$ and $M_l$ is the product of the height and width of the feature maps at layer $l$. Finally, the authors use backpropogation to modify the pixel values of the generated images to minimize the (weighted) sum of the style and content loss to generate the final stylized image. [1]

### 2.2 Controlling Perceptual Factors in Neural Style Transfer

Gatys et al. build on their previous work and implement spatially guided neural style transfer. Specifically, the authors attempt to apply style to different regions of the content image in either the style of (a) different regions of the same style image or (b) multiple style images. To do so, the authors generate masks, or *spatial guidance channels* for each of the *r* styles being applied to the $r$ content regions. They generate $r$ channels, or maps, for the content image and style image(s) each with values $\in$ [0,1] such that the pixels in the $ith$ channel of the content image that are 1 are styled in the style corresponding to the pixels that are 1 in the $ith$ spatial guidance channel of the style image. They then normalize each spatial guidance channel $T_l^r$ s.t. $\sum_i (T_l^r)_i^2 = 1$ and then compute one Gram matrix for each of the *r* channels by taking the element-wise product of the channel and the generated image. The new style loss equation then becomes

$$L_{style} = \frac{1}{4N_l^2} \sum_{r=1}^{R} \sum_{ij} \lambda_r (G^r(X_l) - G^r(S_l))_{ij}^2$$

where $\lambda_r$ is a weight applied to the given style region. The authors propose multiple methods for projecting the masks at each layer, including naively re-sizing them to fit the layers dimensions. [2]

### 2.3 Segmented Neural Style Transfer

Yang et al. work on styling content images in the style of stills of animated films. They work on *segmented* neural style transfer, where style is selectively applied to the foreground or background of the content image. This allows the authors to style the building of the content image in a different style than the surrounding greenery. The authors first generate two segmentation masks for the content image using the GrabCut algorithm proposed by Rother et al. [3] Using the two segmentation masks, the authors then modify the backpropogation equations to apply the style from one style image to the pixels denoted by the first mask and the style of the second style image to the pixels denoted by the second mask, thus selectively applying different styles to different content regions. Their work is a simplified version of spatially guided transfer described above.[1] [4]

## 3 Dataset and Features

Our dataset consists of images of approximately 100 chairs that we have collected online. These are split evenly between four retailers–Ikea, West Elm, Design Within Reach, and CB2. Additionally, these images are split between having all white backgrounds an being in staged setting, i.e. in a living room. In collecting these images, we manually searched across the retailers websites and subjectively determined which images to collect. In creating our dataset, we tried to get a wide variety of styles, colors, and background settings. We then post-processed each image. In doing so, we resized each image to have a maximum height and width of 256 pixels. Next, we applied the GrabCut algorithm to segment the images.[3] Specifically, we generate masks for each image with a 255 at each pixel corresponding to a chair pixel in the original pixel, and a 0 in each pixel corresponding to background. The quality of the masks varied slightly, with images with white backgrounds typically having better segmentation than those with non-white backgrounds (see Fig. 2), allowing us to analyze the impact of the quality of segmentation on the quality of our output.

[1]The authors do not provide equations for implementation details of their masked-loss calculation

Figure 1: CB2 and Ikea chairs and GrabCut segmentation masks

## 4 Method

We make use of the spatial guidance channel work done by Gatys et al. in order to stylize one chair in the style of another. Specifically, we generate masks, or guidance channels, for the content image and the style image, where the mask-activated pixels correspond to the chair in each. However, the problem that we work on is a slightly simplified version of that worked on in the paper, as their are only two channels: the region that contains the chair (foreground) and the region that does not contain the chair. As a result, we compute one mask for each chair that corresponds to the foreground representing the chair.

In order to generate the masks, we make use of the GrabCut segmentation algorithm. [3] This allows us to segment the foreground and background relatvely quickly. Additionally, the OpenCV library has an easy-to-use implementation that allows the user to specificy an initial bounding box to anchor the segmentation with. Once we have generated these masks for each chair, we then use a pre-trianed VGG-19 network with 16 convolutional layers, each followed by one ReLu activiation layer. We use a pooling layer after convolutional layers 2, 4, 8, and 16. Our architecture mirros that implemented in a popular segmented neural style repository.[5]

We then feed into our network a content image, style image, and the mask for each. At each layer at which we compute style loss, we then project the mask into the appropriate dimensions for the layer using OpenCV's resize() method with resampling using pixel-area relation. With the new masks, we then compute *one* Gram matrix for the generated image and the style imag, each. Next, we normalize the masks $T$ s.t. $\sum_i (T_l)_i^2 = 1$. Note that while the original equation presented by Gatys et al. produces one gram matrix per channel and sums losses over these channels, we only care about the loss between the region containing the chair in the content and style images. Thus we do not sum over the background regions of the images; we only compute the loss for the chair regions. Given a normalized content mask $B_l^c$ and style mask $B_l^s$ at layer $l$, we formally compute the style loss as

$$L_{style} = \frac{1}{4N_l^2} \sum_{ij} (G^r(X_l * B_l^c) - G^r(S_l * B_l^s))_{ij}^2$$

where $*$ represents element-wise multiplication. Note that we initialize our generated image with the *content* image, not noise, to quickly generate an image with the background of the content image.

## 5 Experiments, Results, and Discussion

We generate a variety of images using three different transfer methods. Specifically, we use *generic* style transfer (**GS**), *baseline segmented* style transfer (**BS**), and *targeted segmented* style transfer (**TS**). **GS** consists of generic style transfer which simply transfers style from the entire style image to the content of the entire content image. **BS** transfers style from a portion of the style image onto a portion of the content image. However, **BS** applies the same mask to both the style and content image, effectively transferring style from the region in the style image that corresponds to the location of the chair in the content image. **BS** is implemented in the segmented-style transfer repository that we build off of. We then modify this code to implement **TS** by taking in an additional mask for the style image and compute loss in accordance with the spatially guided transfer outlined by Gatys et al. and formulated in section four. Note that **BS** is simply **TS** with the same mask applied to the content and style image and without normalization. Finally, we experiment by searching across a variety of hyperparameters. Specifically, we experiment with layers at which we compute style loss. In style loss group 1, we compute losses at the ReLu activation layer after the first convolution layer

in each layer group, where each layer group is defined as a sequence of convolutional and ReLu activation layers separated by one pooling layer. In style loss group 2, we compute losses only at the first two of the layers, and in style group 3 only the last two of these layers. We refer to the style loss group by SLG. Note that we omit *GS* and *BS* outputs after layer group 1, as these suffice to show the relatively poor performance of the two transfer methods. [2]

### 5.1 Results

Figure 2: **West Elm styled in Eames**

Figure 3: *C* Img

Figure 4: *C* mask

Figure 5: *S* Img

Figure 6: *S* mask

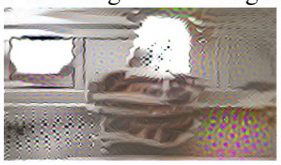

Figure 7: *GS, SLG: 1*

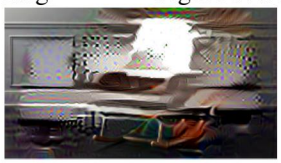

Figure 8: *BS, SLG: 1*

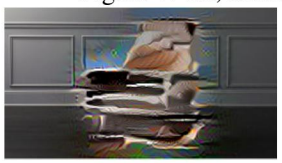

Figure 9: *TS, SLG: 1*

Figure 10: *TS, SLG: 2*

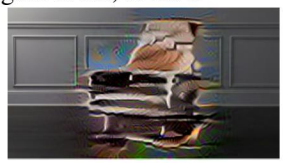

Figure 11: *TS, SLG: 3*

Figure 12: **Ikea styled in West Elm**

Figure 13: *C* Img

Figure 14: *C* mask

Figure 15: *S* Img

Figure 16: *S* mask

Figure 17: *TS, SLG: 1*

Figure 18: *TS, SLG: 2*

Figure 19: *TS, SLG: 3*

[2]Results here use a 0.1 learning rate. We tested with 1, 0.1, and 0.01 and found that 0.1 performed well relative to 1 and 0.01. We additional use a standard iteration rate of 300 iterations, as this produced reasonable results without excessive run time. Additionally, we use an Adam optimizer.

### 5.2 Discussion and Analysis

We now analyze the output. First, note that generic transfer (*GS*) and baseline transfer (*BS*) produce very poor results (see Fig. 7,8). We see that in with BS we style the entire content image, as evidenced by constant perturbations throughout the image. Next, we note that in both BS and GS we see large patches of white. This is likely because we take style from the *entire* style image in GS and the *content* chair's mask in BS, which both include the white background of the Eames chair.

Next, we see that the quality of the segmentation can have an impact on the quality of the transfer. Specifically, we see that the segmentation mask of the first content chair (Fig. 4) includes the area between its legs. As a result, we see that styling is in fact applied to these regions in our output image (see Fig. 9,10,11). We can also compare the results of using different layers at which we compare style loss. Specifically, we note that calculating style loss only at earlier layers leads to much more modest adjustment to style and content than when later layers are used. Note that in both examples (Fig 9,10), we style have minimal "warping" around the edges of the chair and the styling is very close in tone and texture to the content image. When we compare this to the result of using layers 1 and 3, we note that later layers likely cause more significant styling changes in the output layer, especially when we compare how similar the outputs of the two are and how drastic they are relative to group 2 outputs. An interesting result of this may be that we should invest further time and experimentation in how we project the masks as they are projected further down the network. We currently use re-sampling to resize the images, however this may produce very distorted projections as the image gets projected further into the network, leading to unreliable mapping of activations later in the network to pixels in the input image.

## 6 Conclusion and Future Work

Ultimately, we were able to implement the work outlined by Gatys et al. to accomplish transfering style from the region containing the chair in the style image to the region containing the chair in the content image. However, as evidenced by our results, these generate images do not represent fully-formed chairs. Rather, they represent an initial step in transfering the texture/color of the style chair onto the content chair. In order to generate more reasonable results, we would like to further investigate the use of measuring style loss at earlier layers in the network. Additionally, we would like to investigate the result of initializing our images with *noise* rather than the content image in order to less stringently bias the output towards the form of the content chair. We would specifically like to experiment with initializing with random noise *as well as* running our model for more iterations. This will hopefully allow us to generate chairs that are potentially more novel than the results we have produced. Finally, we would like to experiment with providing some form of "slack" to the mask of the content image. That is, we would like to be able to generate chairs that do not simply take the structure of the content chair and then have the texture of the style chair put on top of it. Rather, we would like to generate novel *structures*, which requires that we be able to draw outside the lines of the content chair's segmentation mask. We would like to test each of our hyperparameters more exhaustively, including learning rate and number of iterations. Lastly, we would like to extend our experiments by applying *multiple* masks to each chair, such that we could segment the wooden base and the fabric of the seat and apply separate styles to each. We could segment both the style and content chair and style different regions of the content chair with styles from different regions of the style chair, which we currently don't do, as evidenced by the wooden color showing up in our chair's fabric in image 17, 18, and 19, for example.

## 7 Contributions

As this was a one-person project, I did all of the original work described in this report. The repository that I forked my work from implemented the initial architecture of the CNN I use, as well as the ability to take in a mask and apply it to both the content and style images. However, it did not support the ability to segment the both the style and content chairs with different masks nor the associated normalization. Additionally, I implemented the code to segment the images, as well as the script to run multiple experiments across a grid of hyperparamaters. Further details can be found in the readme.

## References


[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *CoRR*, vol. abs/1508.06576, 2015.

[2] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, "Controlling perceptual factors in neural style transfer," *CoRR*, vol. abs/1611.07865, 2016.

[3] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut": Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, pp. 309–314, Aug. 2004.

[4] K. Yang, J. Lee, and J. Wang, "Semantic segmented style transfer," 2017.

[5] C. Smith, "neural-style-tf." `https://github.com/cysmith/neural-style-tf`, 2016.