Semantic Segmented Style Transfer

Kevin Yang* Jihyeon Lee* Julia Wang*
Stanford University Stanford University Stanford University
kyang6 jlee24 jwang22

Abstract

Inspired by previous style transfer techniques that can capture the style of a particular artist in a content image of a real world scene, we developed a method that uses segmentation on the content image and content-based selection of style images from animated films to preserve greater semantic meaning in the output image. To overcome limitations of object localization in animation and lack of more nuanced categories, we focused on segmenting content images into a building, the foreground, and its landscape, the background. We use the GrabCut algorithm to produce segmentation masks, and using a content loss function, we select two particular frames that most closely resemble the foreground or the background. Finally, we transfer the style based on the Neural Algorithm for Artistic Style [2] using the segmentation masks and separate style images. Our results show that this method maintains more structural information in the foreground than vanilla techniques, while still successfully transferring the art style of a particular animation studio through texture and color, especially in the background.

1. Introduction

In the growing field of the intersection of computer science and art, the question of whether artificial intelligence can produce art has been very much of interest. Style transfer has recently taken the artificial intelligence field by storm, as one of the top trends in 2016 [1]. The paper "A Neural Algorithm of Artistic Style" [2] has been implemented many times, and these implementations have to ability to transfer style of a certain painting, such as Van Gogh's *Starry Night* or Edvard Munch's *The Scream* to a photograph.

In the animation industry, creating the artwork for animated films is a painstaking, time-intensive process [6]. It is important for the films to maintain a consistent art style across all frames. This can be especially difficult when rendering detailed backgrounds, such as countryside landscapes, cityscapes, and elaborate buildings.

While style transfer works well for highly textured paintings with distinct color palettes, it performs significantly worse with much subtler artistic styles, such as the animation style of Studio Ghibli, a popular Japanese animation studio that has produced famous works such as *My Neighbor Totoro* (1988) and *Spirited Away* (2001). Furthermore, style transfer is especially difficult for photographs with rigid lines, such as buildings.

Current implementations of style transfer have many limitations, including in transferring the style of animation to detailed landscapes and buildings. Often, the semantic meaning of a content image is lost because the objects blend together, losing structural, color, and other visual information. In this paper, we investigate approaches to overcome these limitations by segmenting input images and identifying multiple style images to best maintain the semantics of the content image. We transfer style in a semantically segmented way across images with the ultimate goal of capturing the essence of a certain film or studio.

2. Related Work

The first style transfer using neural networks was proposed in *A Neural Algorithm of Artistic Style* by Gatys et al. [2] which performs gradient ascent on a white noise image, minimizing two losses: content loss against an input photo and style loss against a style image. The style loss function is taken at multiple layers in the network, and characterized by the mean squared loss between the Gram matrices of the input image and the artwork.

Since then, style transfer has been a widely explored topic, and many have worked towards improvements on the algorithm. Patch-based methods have been developed using CNN features, such as in the Neural Doodles project [7], which used manually authored pixel labels as semantic annotations, achieving higher quality results than vanilla style transfer. Additionally, a recent paper by Gatys, et al. [3] analyzed different ways in controlling color, spatial location, and scale. They used guided Gram matrices and guided sums in controlling spatial location. Results showed notable improvements in maintaining semantic meaning in regards to foreground and background in the content image.

In the same vein, as the research done by Gatys, et al., Li, et al. combined Markov Random Fields with Convolutional Neural Networks [18] to more realistically transfer semantically similar parts of a style image over to a content image. Their initial results are very promising.

Little work has been done on style transfer specifically for animated styles, such as Studio Ghibli. However, an open source version of Studio Ghibli's Toonz software, dubbed OpenToonz, was recently released [4]. OpenToonz advertises the ability to perform anime-like style transfer, although they do not reveal the method they use.

3. Technical Approach

3.1 Dataset

We collected frames from various Studio Ghibli films, including *My Neighbor Totoro*, *Spirited Away*, *Kiki's Delivery Service*, and *Howl's Moving Castle* for our style images. For our content images, we collected pictures of various California landscapes and buildings on Stanford campus from Google Image search.

3.2 Baseline

For our baseline, we applied simple style transfer on content images of Stanford campus and style images from Studio Ghibli films.

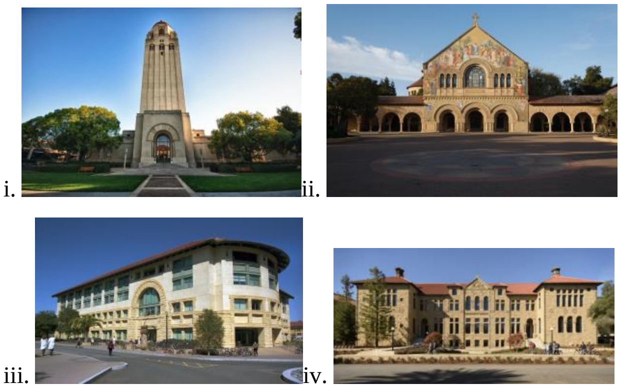
3.3 Segmentation Approaches

We originally planned to utilize object detection and localization to transfer style from object to object across images. However, we found that nuanced segmentation is difficult even with more recent, advanced techniques; for example, Liang, et al. [8] found that the CRF-RNN model had the greatest performance in terms of accuracy among FCN, DeepLab, and DeconvNet, so we tested segmentation on Zheng et al.'s implementation of CRF-RNN [9]. Unfortunately, because of the limited categories, the results were poor for our landscape-type images, which did not have clearly defined objects like bicycles, boats, or buses. Thus, we instead determined that our semantics would be differentiating from buildings in the foreground to landscape in the background. Below we show our results after experimentation with simpler methods to most closely segment the foreground from the background.

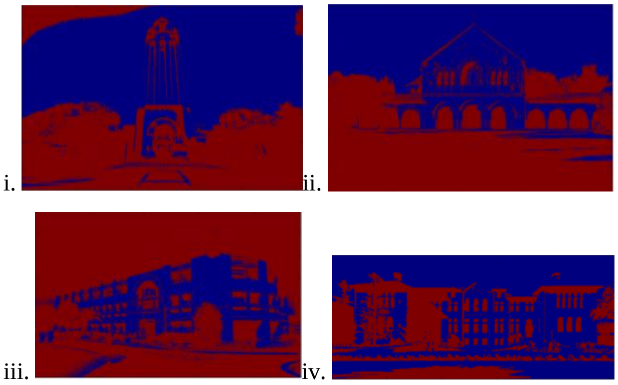
A. Binary segmentation using Otsu thresholding

Otsu's method, named after Nobuyuki Otsu, automatically performs clustering-based image thresholding to reduce a grayscale image to a binary image [8]. The algorithm assumes that there are two classes of pixels following a bi-modal histogram and then calculates the optimal threshold such that their inter-class variance is maximal. We converted our input image to grayscale and then performed Otsu's method through the scikit-image package [11] but found that only in the Gates image (Fig. 1(b)(ii)) the building had been clearly distinguished from the background.

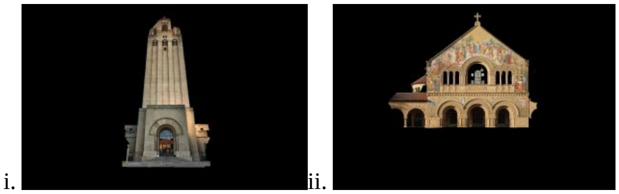
(a) Content Images



(b) Segmentation results using Otsu thresholding



(c) Results using the GrabCut algorithm



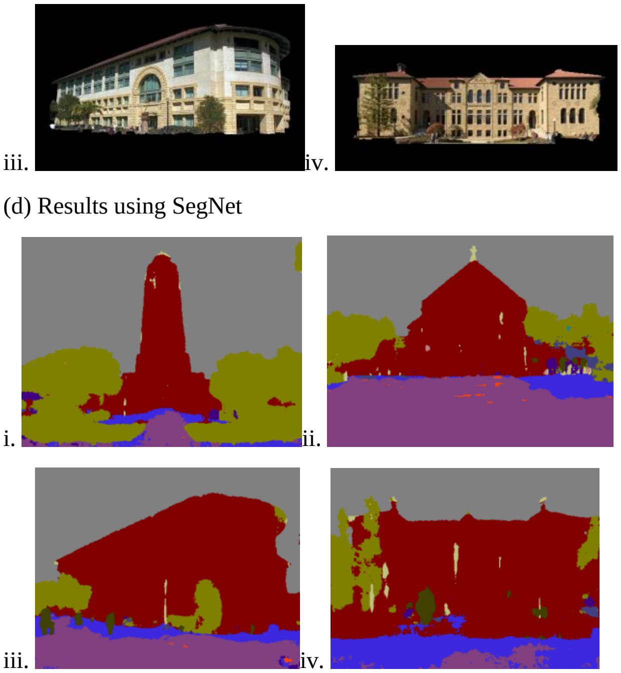


Figure 1. Segmentation results. **(a)** Content images: Hoover Tower, Memorial Church, Gates, and the Chemistry buildings respectively. **(b)** Results from the scikit-image package of Otsu's method. The blue represents the foreground and red the background. **(c)** Results from the OpenCV package of the GrabCut algorithm. The part of the building shown has been classified as foreground. **(d)** Results from SegNet. The red areas have been categorized as "building," light green as "tree," blue as "pavement," pink as "road," and gray as "sky."

B. Foreground extraction using GrabCut

GrabCut, developed by Rother, et al., is another method of foreground extraction [12]. The GrabCut algorithm uses a Gaussian Mixture Model (GMM) to model the foreground and background. GMM learns a new pixel distribution by labelling unknown pixels as either probable foreground or probable background depending on its relation with other pixels in terms of color statistics, similar to clustering. A graph is produced from the distribution, where the nodes are pixels, every foreground pixel is connected to a source node, and every background pixel is connected to a sink node. The weights of the edges are defined by pixel similarity and a mincut algorithm is used to segment the graph, continuing until the classification converges.

An additional feature of GrabCut is its interactivity. The user can specify a bounding box, where the pixels within the box are unknown and those outside are hard-labeled as background. We used the OpenCV package for GrabCut [13] and specified a bounding box roughly surrounding the building portion of a given content image (Fig. 1(c)).

C. Segmentation using SegNet

SegNet is a deep encoder-decoder architecture for multi-class pixelwise segmentation, created by the Computer Vision and Robotics Group at the University of Cambridge, UK [14]. The model consists of a sequence of encoders, non-linear processing layers, and a corresponding set of decoders followed by a pixelwise classifier. Each encoder is composed of one or more convolutional layers with batch normalization and ReLU non-linearity, followed by maxpooling and sub-sampling. By maxpooling indices in decoders to perform upsampling of low resolution feature maps, the model retains high frequency details in the segmented images. The entire architecture can be trained end-to-end using stochastic gradient descent.

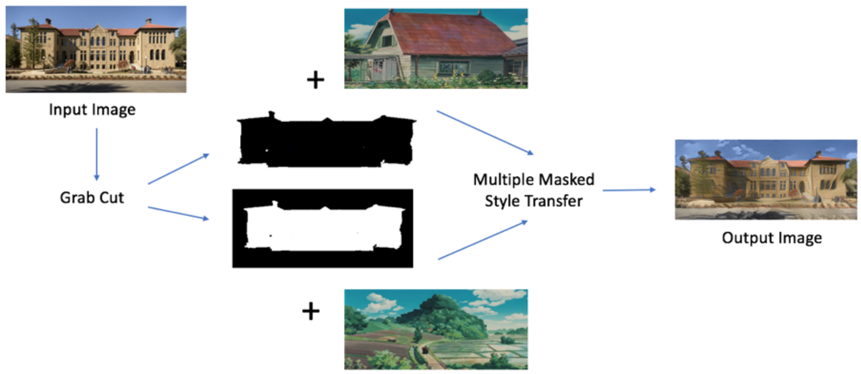


Figure 2. Model Pipeline. After the image is segmented into foreground and background, the best style images are chosen. All are fed into the multiple masked style transfer network to create the output image.

In our case, we used the SegNet demo [15] to produce results (Fig. 1(c)). The model was very good at distinguishing the building, road/pavement, trees, and sky, which are the main object categories we wanted to use in pixelwise classification. However, after comparison with more traditional computer vision techniques like GrabCut we decided to use masks from the GrabCut algorithm instead. The main two reasons for this decision were: the masks from GrabCut had less noise, which can be seen as the yellow pixels within the red buildings in every sample mask (Fig. 1), and the edges of buildings were sharper which is especially apparent in the top roof of the third building (Fig. 1(iii)). We expect to use more nuanced segmentation like SegNet or DeepMask [16] and SharpMask [17] in the future in order to overcome these shortcomings.

3.4 Finding the Best Frames from Animation

After segmenting the input photo, we obtain two photos, one of the foreground and one of the background. The rest of the image is filled in with the mean value of the photo. We then pick an image for the style image for each. For each frame of the movies, we calculate the content loss between the frame and the background. We take the frame with the minimum loss as the style image for background. We repeat the process for the foreground. The content loss, also known as the mean squared error (MSE) is defined as:

$$Loss = \sum_{i,j} (P_{ij} - M_{ij})^2$$

In the end we have two style images, one to be used on the foreground and one on the background.

3.5 Masked Style Transfer

For style transfer, we modified an existing implementation of the neural algorithm of artistic style [19], which uses a pre-trained VGG-16 network. We used two loss functions, one of the content and one of the style. The content loss function taken at one level in the network, and is characterized by the mean square loss of the representations of the input image and the photograph at those levels. The style loss function is taken at multiple layers in the network, and characterized by the mean squared loss between the Gram matrices of the input image and the artwork. The total loss is then a combination of these two loss functions. We then perform gradient ascent on a white noise image to minimize the total loss [2]. We modified this algorithm to be able to take in two masks of the content image and two style images, transferring the style selectively from each image to its corresponding content image mask.

For both baseline and Semantic Segmentation we opted to keep the original color of the content image in order to decrease the number of independent variables when comparing results. Furthermore, across all tests in baseline and Semantic Segmentation we ran 800 iterations using an L-BFGS optimizer with a learning rate of 4e-4.

4. Results

4.1 Baseline

The images for baseline were able to preserve some content from the input image and some style from the style images. However, there were several issues. The building edges became distorted due to the sharp corners present in the style image. Furthermore, without masking off the buildings from the rest of the image the background started hallucinating sharp edges, as seen in the sky in the baseline results (Fig. 3(a i)) and (Fig. 3(a iv)).

4.2 Masked Style Transfer with Handpicked Style Images

The results from Masked Style Transfer with handpicked style images yielded the best results. We can now see a clear distinction between the two styles transferred to each part of the content image. For example, in Fig. 3(b i) and Fig. 3(b iv) we can see that the sky does not have any sharp straight edges; rather the sky is much more semantically similar to the sky in Studio Ghibli animations with puffy and circular white clouds. Moreover, in Fig. 3(b i) we can see that the trees kept more semantic meaning than compared to baseline, where the trees became distorted. The buildings also showed improvement from baseline, with less distortion while still maintaining some animation-like style. However, there was still some distortion present, as seen in Fig. 3(b iv).

4.2 Masked Style Transfer with Automated Style Images

The results from Masked Style Transfer with automated style images were not as successful than when compared to Masked Style Transfer with handpicked style images. However, many issues that were faced in baseline were solved, such as the sharp edges in the sky. Looking at Fig. 3(c i) and Fig. 3(c iv) we can see that the mask that was generated successfully isolated the sky in the content image from the foreground style.

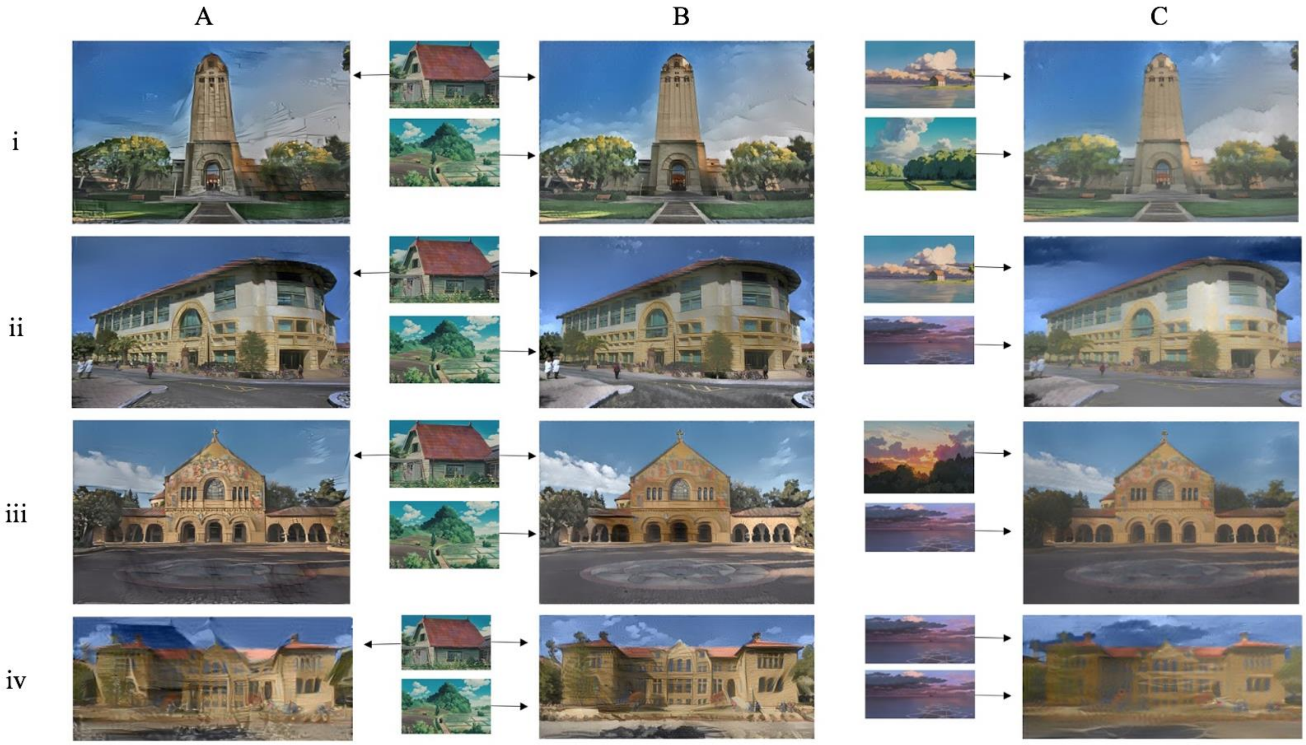


Figure 3. Baseline (A) results, Masked Style Transfer with handpicked style images (B), and Masked Style Transfer with automated style images (C) for Hoover Tower (i), Gates (ii), Memorial Church (iii), and Chemistry Buildings (iv).

One feature the masked style transfer with automated style images produced was that the images produced were more muted, with less contrast in the shadows, most notably seen in Fig 3(c i) compared to Fig 3 (a i). This was likely a result of the color palette and contrast of the style images selected. This can be interpreted as either an improvement or decline from baseline, as the muted colors is often characteristic of the style of the Studio Ghibli films.

5. Conclusion and Future Work

We presented a method of style transfer that uses segmentation of the content image and content-based selection of style images from animation to preserve greater semantic meaning in the output image. Our goal was to capture the essence of a certain animation studio, and we initially planned on detecting, localizing, and transferring style from objects in style images from films to corresponding objects in the content image. However, we ran into challenges, such as poor object detection for animated images and limited categories for objects to begin with. We determined that we would focus on identifying buildings in the foreground and the rest of the landscape in the background. After testing a variety of different pixelwise segmentation techniques we selected GrabCut because of its lack of noise and relative computational simplicity. In addition, instead of choosing style images arbitrarily, we used a content loss function to select particular frames from a given animated film; one style image for the foreground and one for the background, chosen because of their semantic similarity to the content image. With the content image, its segmentation masks, and style images as input, we use a pre-trained VGG-16 network with two loss functions as described in [2], one for content and one for style, to perform segmented style transfer.

We found that this method successfully preserves structural information especially in the foreground, while still being able to successfully transfer the Studio Ghibli art style of painted texture and color for objects in both the foreground and background. Furthermore, we found added benefits in sharper building edges with this method when compared to baseline. Ultimately, while we do not think we have completely achieved our goal of capturing the "essence" of a given animation, we believe there are still important applications of being able to apply style in a semantically-sensitive way as well as applying different styles to different areas or only some areas of an image.

In the future, we hope to use more nuanced semantic segmentation techniques, such as DeepMask [16] and SharpMask [17], for more precise object-to-object style transfer to preserve the semantics of a given content image at an even greater degree. We also hope to experiment with other more complex content loss functions to select frames from animations that are semantically similar to the content image more carefully; for example, we could consider other features such as spatial or structural information. We would also like to be able to transfer to more than just two segments, such as foreground and background, but rather to semantically segmented components, such as building, tree, sky, etc. Finally, we would also like to see if our method generalizes to other animation studios beyond Studio Ghibli or even focus on a certain “mood” or atmosphere of a particular part of a film and explore what aspects of style must be transferred to evoke that mood.

References

[1] Karpathy, Andrej. “A peek at trends in machine learning.” https://medium.com/@karpathy/a-peek-at-trends-in-machine-learning-ab8a1085a106

[2] L. A. Gatys, A. S. Ecker, and M. Bethge. A Neural Algorithm of Artistic Style. arXiv: 1508.0657, 2015.

[3] L. A. Gatys, A. S. Ecker, and M. Bethge, A. Hertzmann, E. Shechtman. Controlling Perceptual Factors in Neural Style Transfer. arXiv: 1611.07865, 2017.

[4] Open Toonz, DWANGO Co. LTD.
https://opentoonz.github.io/e/

[5] E. Shelhamer, J. Long, T. Darrell. Fully Convolutional Networks for Semantic Segmentation. arXiv: 1605.06211, 2016.

[6] “Studio Ghibli Creation Process.” *The Legacy of Hayao Miyazaki*, 2017.

[7] A. J. Champandard. Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artworks. arXiv: 1603.01768, 2016.

[8] X. Liang, B. Zhuo, P. Li, L. He. CNN based texture synthesize with Semantic segment. arXiv:1605.04731, 2016.

[9] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, P. H. S. Torr. Conditional Random Fields as Recurrent Neural Networks. arXiv:1502.03240, 2015.

[10] Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". *IEEE Trans. Sys., Man., Cyber*. **9** (1): 62–66.

[11] Otsu thresholding through the scikit-image package. http://www.scipy-lectures.org/packages/scikit-image/#binary-segmentation-foreground-background

[12] C. Rother , V. Kolmogorov, A. Blake, "GrabCut": interactive foreground extraction using iterated graph cuts, ACM Transactions on Graphics (TOG), v.23 n.3. doi:10.1145/1015706.1015720, 2004.

[13] Interactive foreground extraction using GrabCut algorithm through the OpenCV package. http://docs.opencv.org/trunk/d8/d83/tutorial_py_grabcut.html

[14] Vijay Badrinarayanan, Ankur Handa and Roberto Cipolla "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling." arXiv:1505.07293, 2015.

[15] SegNet Demo.
http://mi.eng.cam.ac.uk/projects/segnet/demo.php

[16] P. Pinheiro, R. Collobert, P. Dollar. Learning to Segment Object Candidates. arXiv: 1506.06204, 2015.

[17] P. Pinheiro, T. Lin, R. Collobert, P. Dollar. Learning to Refine Object Segments. arXiv: 1603.08695, 2016.

[18] C. Li, M. Wand. Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis. arXiv: 1601.04589, 2016.

[19] TensorFlow (Python API) implementation of Neural Style. https://github.com/cysmith/neural-style-tf