CSCI-731 Project: Style Transfer for a Segmented Video

Paul Galatic

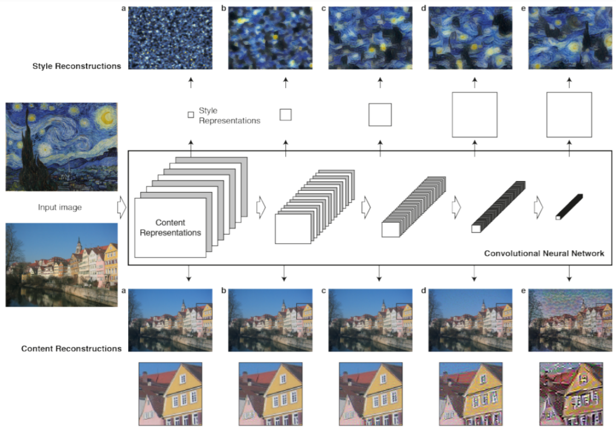

Fig. 1: CNN described by Gatys et. al

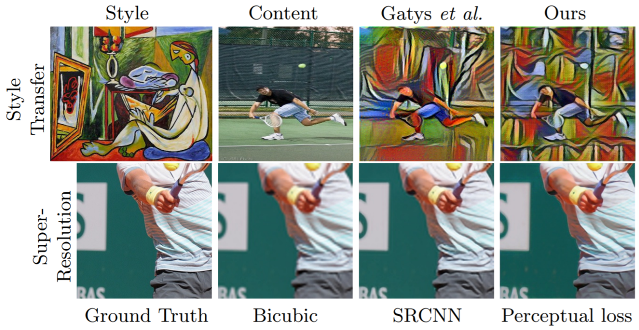

Fig. 2: Johnson et. al comparing their stylization algorithm to [3] and demonstrating the effectiveness of perceptual loss when upscaling images in comparison to conventional methods.

***Abstract*— Neural Style Transfer is the process of transforming one image into the artistic style of another image. Image segmentation generally refers to the process of separating an image into the uninteresting ”background” and interesting ”foreground”. This report analyzes the possibility of combining these two approaches for the purposes of artwork by applying neural style transfer to the background while utilizing advanced image segmentation to preserve the foreground. The algorithms utilized will be Fast Artistic Videos [1] and Reference-Guided Mask Propagation [2].**

I. INTRODUCTION

This report examines the full history of neural style transfer, and also covers the most popular object segmentation techniques for video. The unification of these fields of research will have interesting practical applications for artists in the near future, and also sheds light on the nature of art and human perception.

II. BACKGROUND

A. A Neural Algorithm of Artistic Style

Neural style transfer was pioneered by Gatys et. al [3] and relied on a convolutional neural network (CNN). Their process was simple, and it required two images: A style image and a plain image. The style image would be deconstructed into a ”style representation” to capture the ”texture” that would then be used as a component of the cost function. ”The images are synthesised by finding an image that simultaneously matches the content representation of the photograph and the style representation of the respective piece of art” (p. 4). In other words, low-frequency features of the plain image are replaced with the texture of the style image. This approach had several drawbacks, most notably that it required individual training over each plain image, severely limiting its throughput.

B. Perceptual Losses for Real-Time Style Transfer and Super-Resolution

A caveat of [3] was the fact that they trained on an inefficient form of loss that, while effective, was needlessly time-consuming. Johnson et. al introduced two substantial improvements: The use of feed-forward networks, and upscaling based on perceptual loss [4]. The first is relatively intuitive, but the second warrants further explanation.

Perceptual loss measures what one might intuitively call the ”visual difference” between two photographs, and for computer vision it is an extremely powerful measurement when analyzing images designed for human consumption. Perceptual loss is based on differences in high-level features of images. The primary advantage of utilizing perceptual loss and feed-forward networks is not the quality of the output, which is similar to [3], but rather in the speed it takes to generate output—that is to say, real-time.

C. A Learned Representation for Artistic Style

The next major advancement in this area was the development of a neural network that could train on a set of styles and then utilize those styles as filters, which could then be applied to any plain image, created by Dumoulin et. al [5]. This algorithm is much more practical, as it can transfer the styles of images in real time. However, it can only apply styles that it had time during training to learn.

D. Exploring the structure of a real-time, arbitrary neural artistic stylization network

The next question was whether or not it was possible to have a neural network learn the *essence* of stylization, as

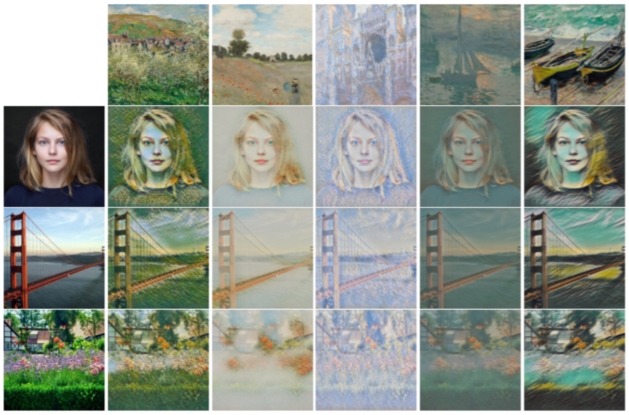

Fig. 3: Exmaples of style images (top) being applied to plain images (left), as demonstrated by Dumoulin et. al.

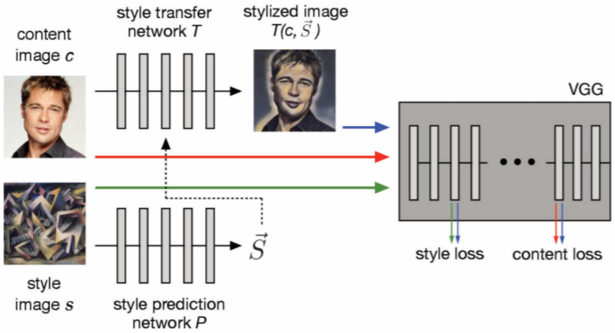
content image c
style transfer network T
stylized image $T(c, \vec{S})$
VGG
style image s
style prediction network P
$\vec{S}$
style loss
content loss

Fig. 4: The network structure created by Ghiasi et. al. The style image is fed into a style prediction network, the output of which is fed into a stylization network along with the plain image. Stylized images are evaluated based on content loss and style loss (high-frequency features and low-frequency features, respectively).

opposed to learning how to apply a specific style (or set of styles). Ghiasi et. al created just this: A neural network that could take an arbitrary style image and arbitrary plain image and combine them instantly, regardless of whether it had seen the style image during training or not [6]. This method relies on a substantial dataset of both training images from ImageNet and style images from a dataset called Painter By Numbers[1], a dataset of paintings. While the application of arbitrary styles may not be applicable to video style transfer due to the fact that the style of a video is typically consistent, this step is worth noting as a milestone in the history of neural style transfer.

E. Style Transfer for Video

The naive example of style transfer for video was possible ever since [4]; this is simply the application of the same style to every frame of a video and stitching together the results. The weakness of this approach is that there is no pressure to keep features consistent between frames. As a result, the colors of the resulting stylized video can appear to flicker wildly, which is not necessarily in the interests of the user.

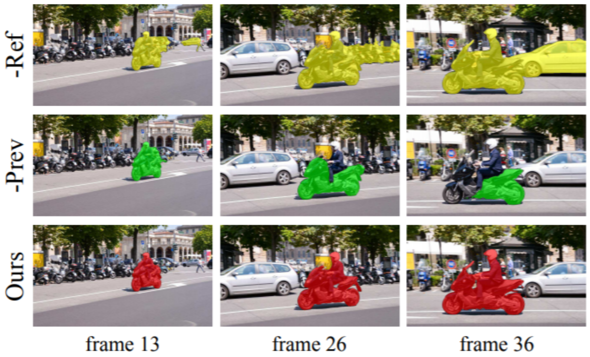

Fig. 5: Examples of the segmentations provided by both the prototypes and the finished product of the algorithm developed by Oh et. al.

Their solution to frame-by-frame inconsistency is to add a temporal constraint to the objective function, penalizing substantial discrepancies between the low-frequency features of two frames. In this way, the stylization is kept more consistent. This penalty considers optical flow, compensating for movement in the plain video. However, this process, as it is based on [3], is extremely slow, and is not suitable for real-time stylization.

F. Artistic Style Transfer for Videos and Spherical Images

Ruder shortly returned to the scene with a dramatic improvement to his algorithm, combining his technique with the feed-forward stylization of [4] to create a fast feed-forward network that, in ideal conditions, can process video frames every second, rather than once every few minutes. This allows for even minutes-long videos to be stylized in a reasonable amount of time. This new version of their older algorithm utilizes the optical flow between frames of the source video to intelligently penalize the neural network during training time for stylizations that randomly distort or flicker between frames. They also use long-term motion estimates to improve texture consistency over time—that way, if an object is included over the course of a few frames, it will maintain its previous style when shown again. They also fixed an issue with the previous algorithm that created artifacts along the boundaries of frames by increasing the comprehensiveness of their algorithm's stylization attempts, incorporating multiple forward and backward passes. Finally, they demonstrated the possibilities for stylizing spherical images that are projected onto a cube map.

G. Fast Video Object Segmentation by Reference-Guided Mask Propagation

As I am primarily interested in the applications of neural style transfer, the reader will be spared a detour into the storied history of object segmentation. Suffice to say that one of the latest and greatest algorithms for object segmentation in video is by Oh et. al, a semi-supervised approach that combines fast runtime with highly accurate segmentation [2]. Their algorithm utilizes the previous frame's segmentation as

[1] https://www.kaggle.com/c/painter-by-numbers

an recurrent portion of their network, helping to predict an appropriate segmentation of the next frame. This implementation is impressive and wholly suitable for the purposes of this report [2].

III. Approach Used

Using Reference-Guided Mask Propagation (RGMP) and Fast Artistic Videos (FAV) in combination is a complex task, as they are not designed to work together. The basic contraption I am trying to design is simple:

1) Choose a video, a target, and a style. The target must be visible in the first frame of the video, and the video should not have jump cuts. Ideally the video will be short in length and have a high frame rate.
2) Manually create a mask of the target in the first frame using the watershed algorithm.
3) Use RGMP to turn this single mask into a mask over the target in all the frames of the video. Use these masks to create versions of each frame where all but the target is transparent.
4) Create a stylized version of the video with FAV.
5) Paste the partially transparent frames from RGMP onto the stylized frames from FAV.
6) Stitch these frames together into a video. Since audio isn't important to any of these tasks, it can be left out if desired.

RGMP's code base is messy and its installation is brutally complex. Fortunately, the majority of the legwork for that portion of this project was completed as part of other coursework. I tidied up the code and demonstrated that the algorithm, with some caveats, could work to generate a sequential mask. This mask can then be used to segment out the relevant portion of the image. However, care must be taken with respect to the source material; RGMP does not perform well over all inputs, and small errors accumulate and compound over time.

In my private experimentation, I have had much better luck with FAV, though its installation and activation procedures are similarly arduous. A few sample videos are included for the viewing pleasure of the reader. Unfortunately, my limited computational resources place constraints on my ability to stylize high-resolution images—the program runs out of available memory and terminates on any input greater in size than 240p—and so my experiment will have to take this into account.

IV. Experiment

My intention is to apply the process discussed in the previous section to the video 'shark.mp4', which can be found in the 'samples/stylization' folder.

V. Discussion and Analysis

To be determined. I have included works of stylization that I have completed in the past in the same folder as 'shark.mp4', which serve as proof that I understand how to utilize FAV. Similarly, my previous work with RGMP is in the 'samples/segmenation' folder, though my results there leave a lot more to be desired. Taming RGMP will be the biggest challenge of this project.

VI. Results

To be determined, though this section will involve a lot of videos and the frames of said videos. I predict that RGMP will mess up a lot and segment random parts of the video instead of/in addition to the target, which will be slightly disappointing, but should still generate an interesting and entertaining result. As mentioned above, preliminary results can be found in the 'samples' folder.

VII. Conclusions

To be determined.

References

[1] M. Ruder, A. Dosovitskiy, and T. Brox. Artistic style transfer for videos and spherical images. *International Journal of Computer Vision*, 126(11):1199–1219, Nov 2018. online first.

[2] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[3] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

[4] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.

[5] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *CoRR*, abs/1610.07629, 2016.

[6] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *CoRR*, abs/1705.06830, 2017.

[2] Implementation details at https://github.com/seoungwugoh/RGMP