CSCI-731 Project: Style Transfer for a Segmented Video

Paul Galatic

***Abstract*— Neural Style Transfer is the process of transforming one image into the artistic style of another image. The slow and rigid networks of Gatys et. al [1], which needed to train on every single image, were eventually replaced with a faster, yet still limited network that could process in real time, but only with one style per neural network. This algorithm was later improved to be practical on even high-resolution images, lowering stylization time from minutes to fractions of a second [2] It was not long before a neural network was trained to juggle several styles at once by reducing styles to points in an embedding space [3], and later a fully robust algorithm was developed that could apply an arbitrary style to an arbitrary image in real time [4]. These advances make real-time style transfer for video possible, though a naive implementation would suffer from a lack of image stability between frames. An early stability enhancement, while effective, could not be applied in real time, as it greatly lengthened processing time [5]. This stability enhancement was quickly improved [6]. This report aims to combine neural style transfer for video with object segmentation for video.**

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

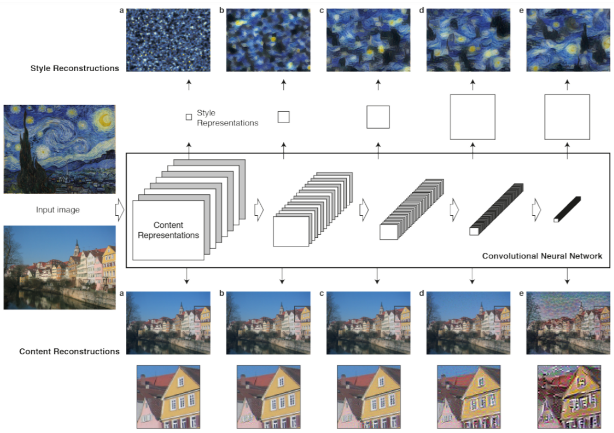

Fig. 1: CNN described by Gatys et. al

Neural style transfer was pioneered by Gatys et. al [1] 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 [1] was the fact that they trained on an inefficient form of loss that, while effective, was needless. Its network structure is also needlessly complex. Johnson et. al introduced two substantial improvements: The use of feed-forward networks, and upscaling based on perceptual loss [2]. The first is relatively intuitive, but the second warrants further explanation.

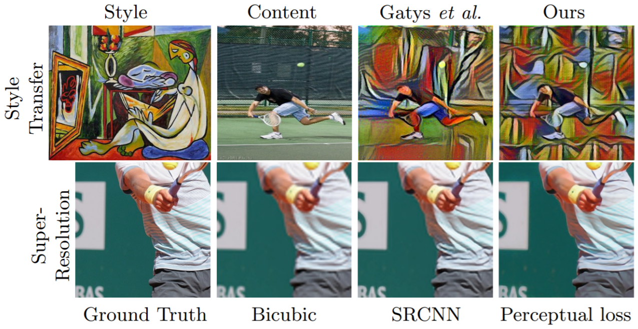

Fig. 2: Johnson et. al comparing their stylization algorithm to [1] and demonstrating the effectiveness of perceptual loss when upscaling images in comparison to conventional methods.

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 [1], 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 [3]. 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.

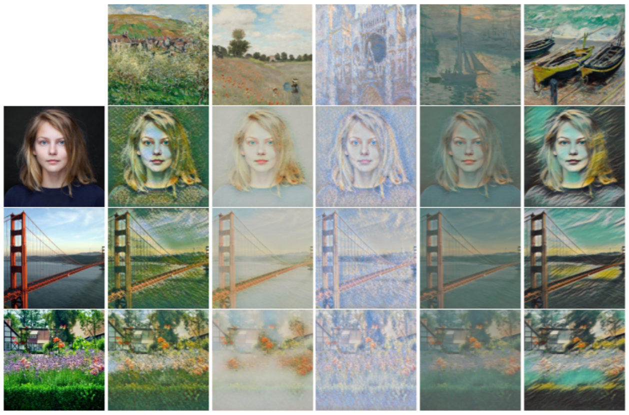

Fig. 3: Exmaples of style images (top) being applied to plain images (left), as demonstrated by Dumoulin et. al.

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

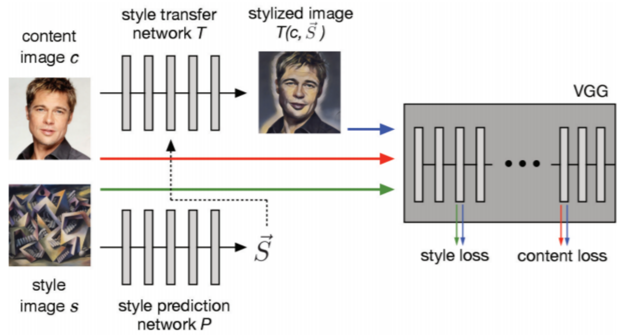

Fig. 4: The network structure created by Ghiasi et. al. The style image is fed into a style prediction network, the output of which is fed into a stylization network along with the plain image. Stylized images are evaluated based on content loss and style loss (high-frequency features and low-frequency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

The next question was whether or not it was possible to have a neural network learn the *essence* of stylization, as 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 [4]. 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 [2]; 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. 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 [1], is extremely slow, and is not suitable for real-time stylization.

F. Stabilizing neural style-transfer for video

These problems were tackled by Rainy et. al [6]. They describe combining the work of [2] and [5] to create a fast feed-forward network that can process video in real time. To solve the problem of flickering, they apply stabilization to the objective function during the initial training process, rather than at runtime. This stabilization loss is achieved by passing two images to the stylization kernel: the original plain image, and a slightly noisy version of this plain image. The loss provides an incentive to minimize the differences between the resulting images. The intuition is that most changes in video frames are quite small, so small differences in inputs should produce the same stylization. Rainy et. al utilize this to great effect, and it is on their work on which this report is based[2].

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

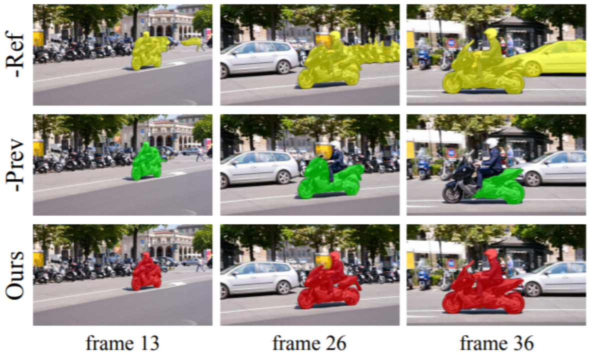

Fig. 5: Examples of the segmentations provided by both the prototypes and the finished product of the algorithm developed by Oh et. al.

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

[2] Implementation details at: https://github.com/ElementAI/chainer-fast-neuralstyle/tree/stable-style

As this report is 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 [7]. Their algorithm utilizes the previous frame's segmentation as 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 [3].

REFERENCES

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

[2] 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.

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

[4] 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.

[5] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. *CoRR*, abs/1604.08610, 2016.

[6] Jeffrey Rainy and Archy de Berker. Stabilizing neural style-transfer for video. *Medium*, 2018.

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