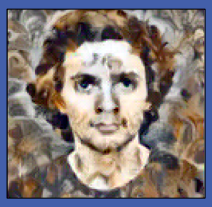
Neural Style Transfer for Videos

Kyle Meredith and Stefan Himmel

Middlebury College CS701



MOTIVATION

This project arose from two primary motivations: to learn about machine learning, specifically Convolutional Neural Networks (CNNs), and to apply our knowledge to create novel artistic output. We gravitated towards the field of Neural Style Transfer (NST), since it sits at the intersection of technical implementation and compelling creative output.

BACKGROUND

Neural Style Transfer (NST) is a popular application of machine learning that transfers the style of a style image to a content image while maintaining the important features of the content (see Figure 1). There are two main techniques to achieve this effect: the optimization technique, which allows arbitrary style images but stylizes slowly, and the feedforward method, which stylizes in real-time but must be pre-trained for one particular style image.[3] In recent years, both of these techniques have been extended to apply image styles to video sequences, which is the focus of this project.

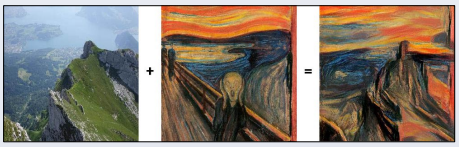


Figure 1.[8]

In all of these cases, neural style transfer involves a convolutional neural network. A CNN is a particular type of neural network architecture that is used to extract features of images—commonly used for image classification (exemplified in Figure 2). It forwards a three-channel image through a series of layers that perform various linear algebra transformations, the most important of which is *convolution* with a filter. The output of these convolution layers is a series of *activation maps*, which reflect specific features of the input image (e.g. vertical edges or faces).

NST techniques achieve stylization by using CNNs that have been pre-trained for image classification. In order to stylize an image, two loss functions are introduced: *content loss* and *style loss*. These loss functions involve three inputs: the content image, the style image, and the generated image, which is the stylized result of the NST process. At a high level, the content loss calculates the mean-squared error between the content image and generated image, while the style loss calculates the mean-squared error between stylistic features of the style image and generated image. By minimizing both of these loss functions, the generated image will share the main stylistic features of the style image while retaining the important features of the content image.

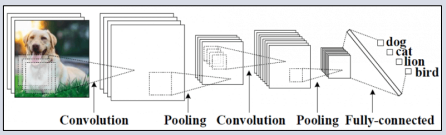


Figure 2.[9]

PROBLEM STATEMENT

For our project, we targeted two specific applications of neural style transfer. The first target was to implement a feedforward NST system that can stylize videos in real-time, based on a paper by Huang et al. titled "Real-Time Neural Style Transfer for Videos."[4] We refer to this goal as *implementation*.

The second target was to explore new possibilities within the field of NST, developing a system that allows novel support for *style videos*. This system takes as input a content video and style video, splits each video into their frames, then uses an optimization NST system by Johnson et al.[5] to apply the style of consecutive frames to corresponding frames in the content video. We will refer to this goal as *invention*.

METHODS

Implementation

We implemented our system for real-time style transfer for videos using Pytorch, a modern deep learning framework for Python. The architecture is based on the paper "Real-Time Neural Style Transfer for Videos" by Huang et al.[4], and it consists of two primary elements: a hybrid loss network and a stylizing network (see Figure 3).

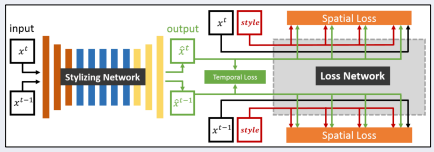


Figure 3.[4]

The hybrid loss network utilizes the VGG19 convolutional neural network, an industry standard architecture that is pre-trained for image classification and commonly used for NST. The hybrid loss consists of two constituent losses: the spatial loss and the temporal loss. The spatial loss is the weighted combination of three components: content loss and style loss, which are typical to NST systems, and an added TV regularizer that enforces smoothness between pixels. The novel temporal loss function calculates the mean-squared error between adjacent generated frames based on the optical flow between them. By minimizing this loss, the stylizing network creates smooth transitions between frames.

The stylizing network is a custom feedforward CNN specified in section 3.1 of Huang et al.[4] This network takes as input a content frame and, according to its internal parameters that have been tuned by the hybrid loss network, outputs the corresponding stylized generated frame. When the stylization network is sufficiently trained, indicated by the convergence of the hybrid loss to a small value, it can stylize single frames of a video fast enough to create NST videos in real-time.

Invention

In pursuit of our second goal of invention, we created a Python script that performs neural style transfer with style videos. To achieve this functionality, we began with a pre-made NST system for images by Johnson et al.[5], which was written in Lua using the Torch deep learning framework. Our program takes as input a content video and style video, then splits them into frames using OpenCV. It loops through the frames of the content video, feeding each of them to Johnson's NST system along with the frame of the style video at the corresponding index (see Figure 4). Ultimately, this process produces a sequence of generated frames with a dynamic style, which the script combines into an output video using ffmpeg. Generating a 426 x 240 output video takes roughly 2 minutes per frame on the Gattaca GPU.

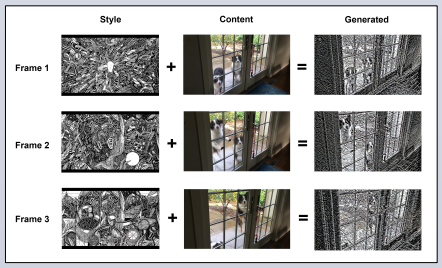


Figure 4.[2]

RESULTS

For our implementation system, we were unable to fully replicate the desired results within the semester. As illustrated in Figure 5, the stylized frames produced by our final model resemble the content image, but they do not share the intended primary features of the style image. For comparison, we have included the ideal stylized output from this combination of content and style, generated using a pre-made NST system by Ruder et al.[7]

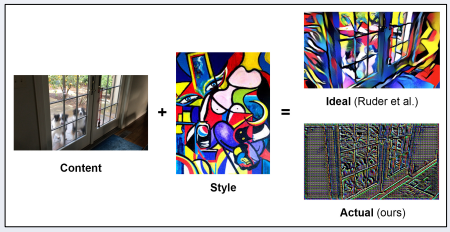


Figure 5.[6]

Our invention system for NST with style videos performs exactly as planned. The final system takes in a content video and style video and successfully generates an output video with a dynamic style (see Figure 6). As expected with this method, the results are quite flickery, since our system does not incorporate temporal loss.

Figure 6.

CONCLUSION

In conclusion, we met both of our original goals for the project. By implementing a nearly-complete real-time NST system for videos, we learned a great deal about convolutional neural networks and their creative applications. Similarly, by implementing a prototype for NST with style videos, we satisfied our original hopes of creating novel artistic output.

If we decide to continue work on this project in the future, there are several clear next steps. For the implementation side of the project, we would first fix the spatial loss to achieve accurate stylization. Then we would shift our focus to integrate the temporal loss function that we began coding.

For the invention side, there is significant room for experimentation and improvement. We would introduce new stylization options like style weight and maintaining original colors, as well as procedural parameters to modify the rate and dynamics of the video stylization. Finally, we would try to minimize flickering by introducing a temporal loss term. This could be done by modifying Manuel Ruder's system for video NST, which includes a temporal loss module.[7] This approach was our initial plan, but the repository was developed with Torch in Lua—we did not have time to delve into this unfamiliar environment.

REFERENCES

[1] G. Carlsson. Using topological data analysis to understand the behavior of convolutional neural networks. https://www.ayasdi.com/blog/artificial-intelligence/using-topological-data-analysis-understand-behavior-convolutional-neural-networks/, 2018.

[2] D. Deacon. DDWIWDD (Dan Deacon When I Was Done Dying), https://www.youtube.com/watch?v=tujquvbj4re, 2015

[3] S. Desai. Neural Artistic Style Transfer: A Comprehensive Look. https://medium.com/artists-and-machine-intelligence/neural-artistic-style-transfer-a-comprehensive-look-f54d8649c199, 2017.

[4] H. Huang, H. Wang, W. Luo, L. Ma, W. Jiang,X. Zhu, Z. Li, and W. Liu. Real-Time Neural Style Transfer for Videos. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 7044–7052, July 2017. doi:10.1109/CVPR.2017.745.

[5] J. Johnson. Neural-style. https://github.com/jcjohnson/neural-style, 2017.

[6] P. Picasso. Untitled. http://wallpaper.istriku.site/picasso-famous-paintings-worth/, 1960.

[7] M. Ruder, A. Dosovitskiy, and T. Brox. Artistic Style Transfer for Videos and Spherical Images. CoRR, abs/1708.04538, 2017. URL:http://arxiv.org/abs/1708.04538, arXiv:1708.04538.

[8] M. Singh. Artistic style transfer with convolutional neural network. https://medium.com/data-science-group-iitr/artistic-style-transfer-with-convolutional-neural-network-7ce2476039fd, 2017.

[9] M. D. Zeiler and R. Fergus. Visualizing andunderstanding convolutional networks.Lecture Notesin Computer Science, pages 818–833, 2014. URL:http://dx.doi.org/10.1007/978-3-319-10590-153,doi:10.1007/978-3-319-10590-1_53.