Peter Victoratos

CSC 411

April 17, 2018

**Milestone 1: Project Plan [10 pts] (Due: Tue 4/17)**

**Problem:**

The problem I am proposing to solve is to speed up the video editing process of style transfer. Although programs like Final Cut Pro and Adobe Premiere have a vast library of edits and special effects ready to be applied to any footage, they cannot do everything that a filmmaker would want. This holds especially true when considering the time component of editing (which is usually pretty long to begin with). The most essential part of editing through a video processing software is the keypoint. Keypoints are distinct times over an event (or video timeline) in which an effect turns on, off, or is altered in someway. This works great for changing audio volume across a video, ken burns-style movements and for most special effects in general. Keypoints are extremely useful however there are some times where lining up each specific frame to its respective keypoint could take extremely long. A prime example of this, my project’s focus, would be style transfer. This is when an artistic style is taken from one peice of art and applied to some other object (a video frame in this case). Having an effect like this can probably be done using key points but everything would be done by hand. Thanks to Python we can use feature detectors, recursive calls and the overall better processing speed of a computer, rather than just using our eyes. From a content creators perspective, being able to write a script will save me time and money as well. It can also be reapplied to any combination of paintings and input videos (if it works perfectly of course).

**Solution:**

To create a program that performs a style transfer on video segments, I will have to construct a Convolutional Neural Network. This network will create the different convolutions so I can get all of the features in the image. Beyond this, I could calculate a *Gram-matrix* to get the output tensors from the style-layers. These map the different brush strokes and colors so it can be later applied to our output image. I will then either create or use a pre-trained neural network that optimizes the loss function (the difference between features in the pre and post images) to stylize the image. Once this is functioning, I can experiment with having an input video instead of just one image and having the output video source from a live webcam.

**Method:**

This project will first read in the three images of paint styles that we want the stylizations to cycle through. It will then construct a file (maybe a gif or mov) that fades from one painting to another (by making the first image less opaque and the second image more opaque at an even rate). This file will then be analyzed by the detectors in real time, adjusting the overlay effect on our unedited input video (or the webcam). The program will return the newly edited video (or gif).

Should be able to find three high quality images from the web based on what artist you selected, array slices them to be the same size so the filter isn’t messed up.

A mockup of what I would like my ideal output to be is below:



**Data/Experiments:** I will create three categories of painting style to choose from (based on the artist). The artists I will choose are Picasso (cubism), Wyndham Lewis (vorticist), Piet Mondrian (De Stijl) Each artist will either have 10 images off of google or if I can find a dataset with more I’ll use that.

•I’ll have to calculate time to make sure that the program runs on time with the rotating photos

•Accuracy will be measured on my algorithm that collects my images

•I will also perform tests on how accurate the stylization is with different types of scenes, ones with bad lighting/motion/etc.

**Low Target:** Functioning Style Transfer algorithm that can transfer a stylze from an image to another image

**Medium Target:** Style Transfer can cycle through different paintings and apply this to the output image

**High Target:** Style Transfer works well on most types of images including live video from the video cam. Another cool experiment would be able to find a way to have the style change with an action of the subject in frame.

**Breakdown**

* Background reading of the topic - 3 hours
* Find the most ideal dataset of photos - 1 hour
* Create the program that will either get input paintings - 3 hours
* Train the models - 2 hours
* Find the best feature detector and adjust it so it works accurately - 3 hours
* Create Style Transfer algorithm from one photo to another - 5 hours
* Improve Style algorithm to work for videos - 2 hour
* Cycle through the chosen artist and produce a gif/video to be scanned - 2 hour
* Improve the stylization algorithm to accept input videos (rather than images) - 4 hours
* Compile data and write report - 3 hours
* Sync style change with object action - 4 hours

Total Time: **32 hours**

<https://medium.com/data-science-group-iitr/artistic-style-transfer-with-convolutional-neural-network-7ce2476039fd> Here’s the link I found this project at

<https://gist.github.com/genekogan/d61c8010d470e1dbe15d>

<http://genekogan.com/works/style-transfer/>