Machine Learning for the Arts
UCSD SPRING 2019
Final Project

Hypnotic Bloom

Brian Sebastian

Description:

For this project, I decided to utilize Google DeepDream on kaleidoscope videos to produce interesting effects. By utilizing OpenCV to split the video into frames, I was able to apply DeepDream onto each frame and then reassemble the frames back into a video. I experimented with different ways of applying DeepDream to the frames by running each frame through the algorithm multiple times, averaging the inputs to DeepDream to create less noisy videos, and used different layers of the pretrained network to see different effects. I also experimented with playing the videos at different speeds to which one was most appealing. In the end, I was able to produce a video that turned all of the geometric shapes of the kaleidoscope video into images that look somewhat like flowers. The final video is only 30 seconds long and at a resolution of 360p due to the time it took to run each frame through DeepDream. Given more time, I would like to create a longer video at higher resolution that would go through multiple color schemes.

Concept:

I was inspired to do this project after seeing the trippy effects that DeepDream had on still images. After seeing this, I became interested in seeing how DeepDream could be used to morph frames of video into one another. When considering what type of video would work best to see this effect, the first ideas I came up with were to use either kaleidoscope videos or videos of zooming in on fractals. I initially decided to use these kinds of videos because I find them interesting to watch and see how the image morphs with time. With this, I also thought that it would be interesting to experiment with the frame rates of these kinds of videos and see how this would affect the noticeability of the effects of using DeepDream. Due to the high run times involved from the number of frames needed to use with DeepDream, I decided to only use kaleidoscope videos. I chose kaleidoscope videos over fractal videos because I like the symmetry that they have and I thought that it would make it easier to appreciate entire frames and notice the subtle changes introduced by DeepDream.

Technique:

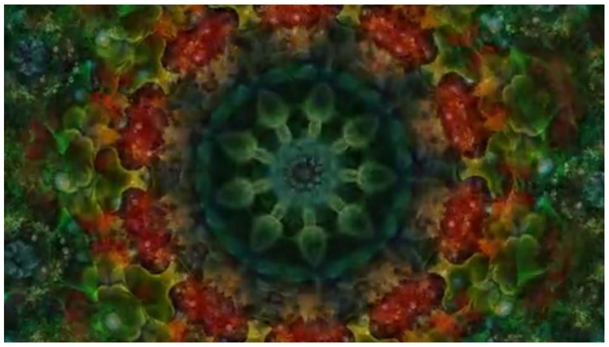
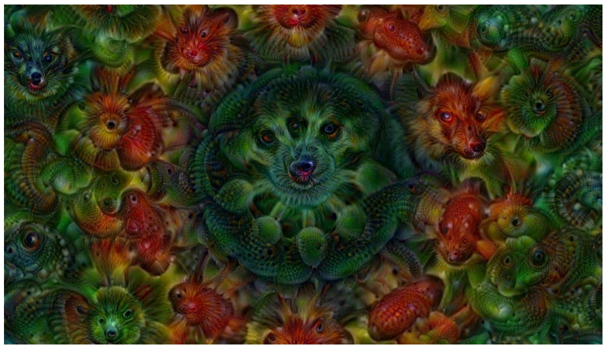
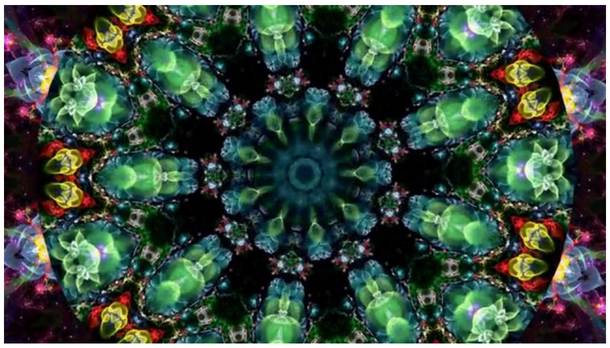
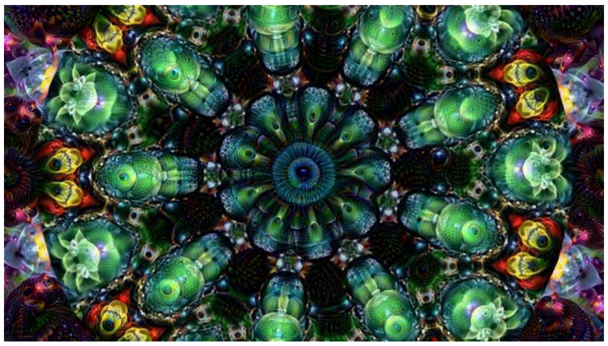
I began by finding a kaleidoscope video that I wanted to use. I imported the video using OpenCV and split it into frames. I also watched the video multiple times to pick which segments of the video I wanted to test on. Given the time it takes for DeepDream to run on each frame, I had to be somewhat selective and choose the segments that I thought would be most interesting since it would not be reasonable to run the entire video through.

In terms of how I used DeepDream on each frame of the video, I decided to just use the pretrained network to perform image generation. I was interested in training a network so that I could have better control over the effects produced in the images. However, I decided that it would be more worthwhile to focus on testing different inputs and ways that DeepDream could be applied to the images.

Process:

For my first attempt, I decided to use the full network because I liked the effects that it had in the example jupyter notebook. I was hoping that it would be able to create images of animals that would fade in and out as the kaleidoscope video changed. I initially chose some segments of the videos with interesting color combinations and ran a couple frames through the algorithm to see how it would affect them. Below are some examples of individual frames before and after running through the algorithm:

Before: After:

Although the differences were somewhat subtle, I thought that they were interesting enough to create a video with them. I also wanted to have the video zoom in and out over the duration of the video, so I wrote a short script that using OpenCV to accomplish this. The script essentially uses the full frames for the first quarter of the video, continuously zooms in over the second quarter, stays zoomed in at 2x for the third quarter, and zooms continuously zooms back out to the full frame over the last quarter of the video. Another goal I had in mind was to create a video that could repeat without too abrupt of a transition between the ending and beginning. To accomplish this, I decided to have the video slowly fade in for the first couple seconds and fade out over the last couple seconds. I wrote a short script to accomplish this, where I defined the interval over which the video should fade in or out and all of the color channels are scaled down depending on the time within this interval. The video from this first attempt is listed as "deepdream_kaleidoscope20.mp4" in this Github repository. To create this 36 second long video I had to run DeepDream on 720 frames with a resolution of 1280p, which took around 13 hours.

I was unhappy with this first attempt as many of the subtle differences became unnoticeable with frame rates higher than 15 fps. Frame rates lower than this were unsatisfactory as the video took too long to change to remain interesting. One of the main problems was that the stochastic nature of the effects caused by DeepDream caused each frame to look substantially different from the ones before and after it. When compiled into the video, these differences look like random noise instead of the morphing effect I was hoping to achieve.

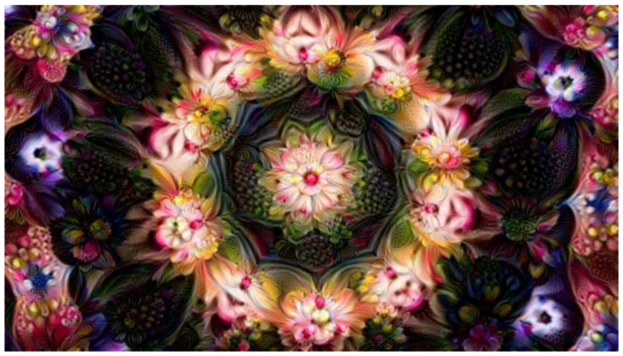
To make each frame look like it morphs into the one after it, I applied a weighted average between the current frame and the previously outputted frame before running each one through DeepDream. At first I used an equal weighting for each frame, but this caused the effects of DeepDream to overtake the kaleidoscope video and after a while the influence of the kaleidoscope frames could no longer be seen. The video essentially began to look as if DeepDream was continuously applied to the same image. To fix this, I changed the weighting so the current frame would have a weighting of 0.75 and the previous frame would have a weighting of 0.25. This created the desired morphing effect so I decided to move on to making the final video.

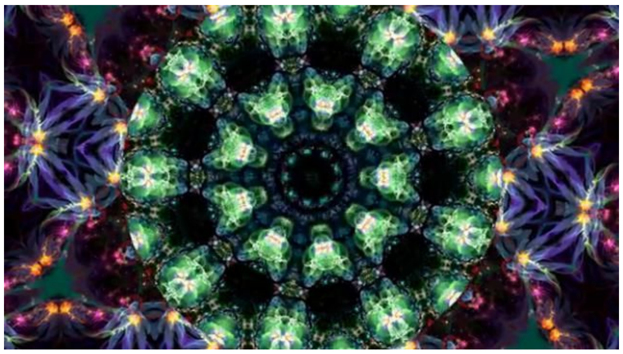
To reduce the time it would take the program to run, I decided to reduce the resolution of the input video to 360p. Before running the program to create a video, I did some more testing with different layers of the network to see their effect. I tried a couple and decided to use one that softened all the straight edges of the shapes in the kaleidoscope video. I also tried running a frame through the algorithm multiple times to exaggerate the effects it would have. I decided that running each frame through twice was a good compromise between the effects produced and the total runtime.

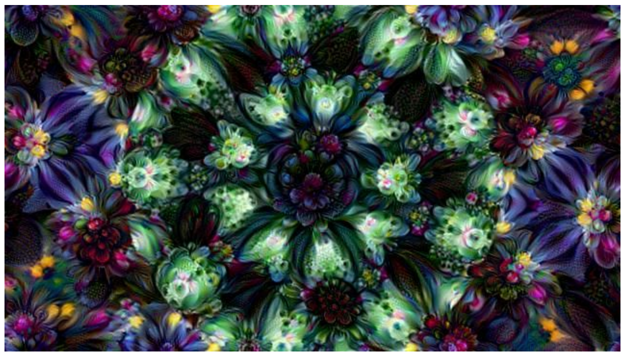
Result:

The final video produced is 30 seconds long and contains 450 different frames. The video fades in and out over an interval of 3 seconds, and I believe Iwas able to accomplish my goal of creating a video that can be repeated without an abrupt transition between the beginning and end. The video also has the same zoom in and out pattern as previously described and I think it adds an interesting dimension to the video. I was really happy with the flowery effects it had on all the shapes of the kaleidoscope video and am satisfied with the output overall. The final video is titled “Hypnotic Bloom” and below are some examples of frames before and after processing:

Before: After:







Reflection:

Overall, I am happy with the final output of this project. This project was very experimental in nature and am both satisfied with the output and surprised with how well it worked. To improve upon this project, I would like to use a higher resolution input video that is longer and goes through more patterns. Given more time, I would also ike to experiment with using fractal videos as inputs.

CODE: https://github.com/ucsd-ml-arts/ml-art-final-brian-sebastian