

Machine Learning for the Arts  
UCSD SPRING 2019  
**FINAL PROJECT**

## Generating Rivers



Jake Bauer

## DESCRIPTION

In this project I used Carolina Caycedo's *Dammed Landscape* as inspiration to train a Sketch-RNN model to generate sketches of rivers, and then hand-render the sketch to achieve a similar aesthetic. I was disappointed with my results--as I spent days collecting and parsing data and messing around with hyperparameters to achieve an output I was not satisfied with.

### Concept:

This project was largely inspired by Carolina Caycedo's *Dammed Landscapes*, featured in her documentary/experimental film *Yuma: Land of Friends*. Caycedo's work with *Dammed Landscapes* and *Yuma: Land of Friends* critiques mega-development projects by focusing on the damming of the Magdalena River (Yuma) in Colombia. I was inspired by her way of imagining a new landscape through ink on paper. Furthermore, I thought her use of inversion was impacting--in that if used satellite images (a technology developed in part for military surveillance and symbolic of mega-corporation development/control), and uses such to call attention to the sterilizing and depersonalizing nature of a black and white satellite image. My concept for this project was to similarly use machine learning techniques--techniques often associated with capitalist profit-optimization, surveillance, and limited accessibility--as a cooperative entity in imagining the flow of dammed rivers. As far as aesthetically, my goal was to achieve a similar look to Caycedo's *Dammed Landscapes* work.

### Technique:

My technique for this project was to use Sketch-RNN to generate my drawings, and then hand-render these sketched with black ink on top of printed out images of black and white

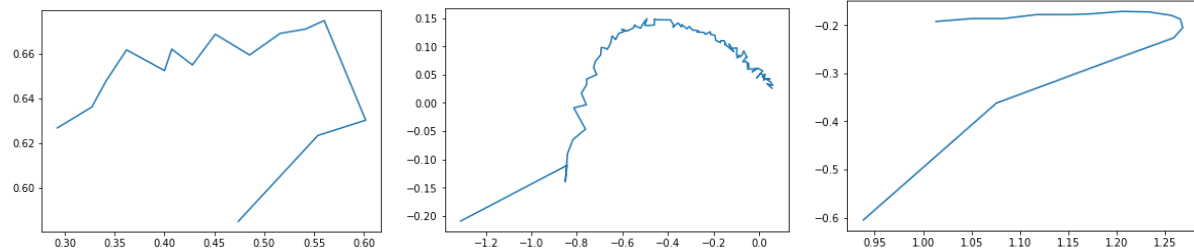
satellite images. Because if unsatisfactory results (see below), I only proceeded to hand-render one of these sketches.

### **Process:**

As I was developing this project, the most time-consuming part was data collection and parsing. I used data from the Natural Earth Datasets, which provided GIS files for the centerlines of all major rivers around the world. Converting this data was a pain, but more frustratingly I thought that my data conversion was wrong--so I re-parsed everything using longitude/latitude instead of normalized vector positions--but then later realized that it was the function used to display the strokes that messed up the output I saw. After correctly representing and sampling the data, I trained Sketch-rnn on it. I trained my model for over 8 hours, and it gave quite similar results to my model with only 30 min of training--so I suspect that I needed a vastly longer training time (maybe on the scale of days) in order to get the output I wanted (I suspect that the preset sketch-rnn models were trained for an extremely long time with much more processing power than was available in Colab).

### **Result:**

Characteristic examples from my final results are below. As you can see, occasionally my model would generate fractal patterns similar to that of rivers, but there was no long term structure. Overall, I was immensely disappointed in the technical result. Aesthetically, I was pleased with how my hand-rendered example turned out though, and think there was potential in the concept behind this project--it just needed to be executed without Sketch-RNN as the model used.



*Results* (from left to right): Low number of strokes, high temperature; high number of strokes, high temperature; high number of strokes, low temperature

### Reflection:

I definitely want to develop more on this project. My current plan is to first try training sketch-rnn for a very very long time. If I still don't see any improvement, then I'll abandon the sketch-rnn idea and use pix2pix or some edge-detection to generate line drawings of rivers directly from a black and white satellite image. Then, I'll feed an incomplete image of this edge through a GAN prediction network to try and complete the river--overlaying it on top of the original satellite image and hand rendering with ink. I think this could work better simply because Sketch-rnn had so many restrictions, but I think it could also fare better as an end-to-end process--there the artist just inputs an image of a dammed river and the machine generates its own imagining and overlays it for the artist to interpret.

**CODE:** <https://github.com/ucsd-ml-arts/ml-art-final-jake-bauer>