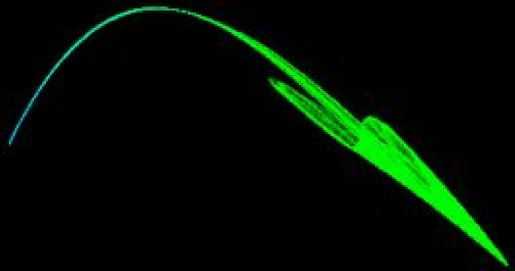
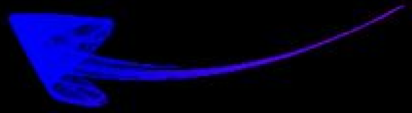


Genetic Evolution of Aesthetically-Pleasing Fractals using Convolutional Neural Networks

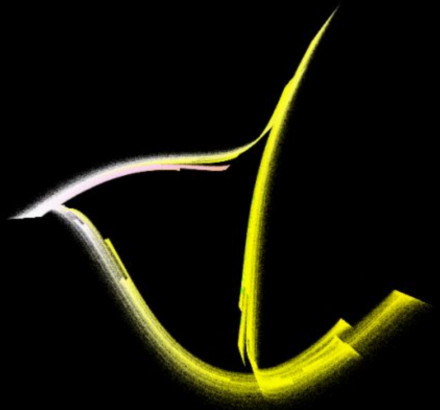
Kevin Yeh

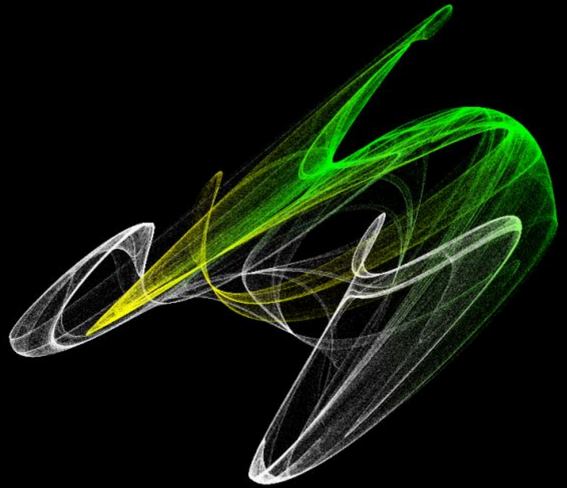


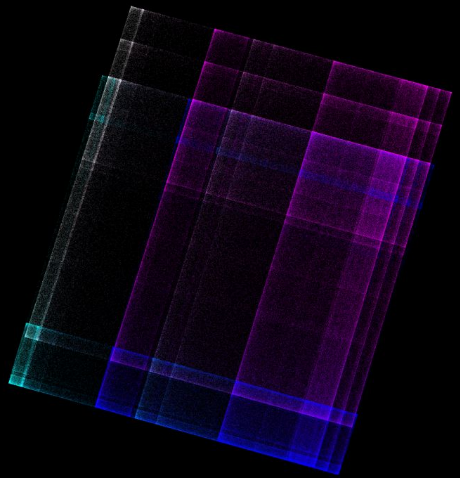
Motivation

Machine Learning and LfD has focused on the automation of control tasks with quantifiable evaluations —

As humans, how do we quantify the personal preference of aesthetics?



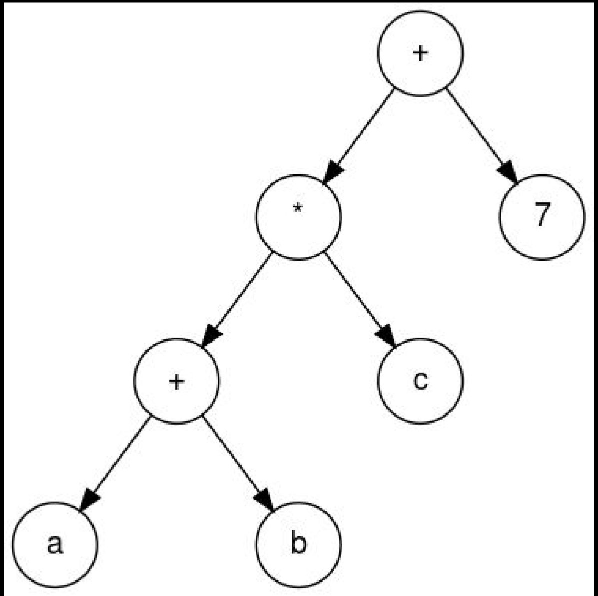




Background: Evolution with Genetic Algorithms

Fractal Representations are defined as a set of recursive mathematical equations outlining a sequence of points in Cartesian space.

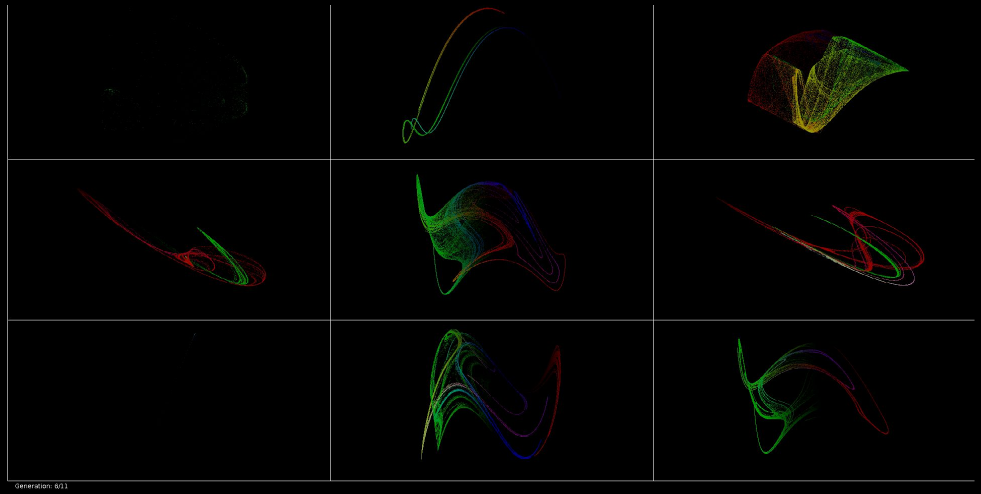
$$x_{n+1} = sin(a * y_n) + c * cos(a * x_n)$$
$$y_{n+1} = sin(b * x_n) + d * cos(b * y_n)$$



Background: Evolution with Genetic Algorithms

Interactive, human-guided evolution:

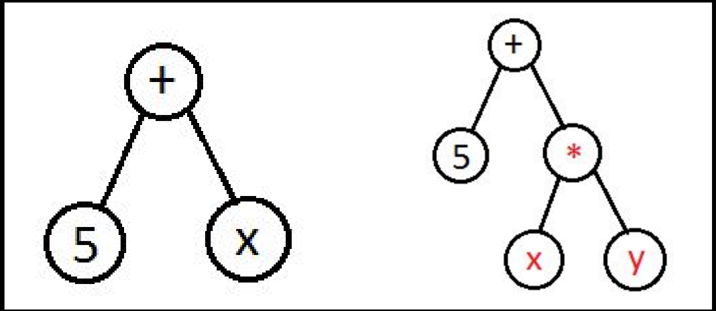
- Each generation consists of 9 fractals
- Allow the user to pick “good” fractals, then genetically combine them to form a new generation.



Background: Evolution with Genetic Algorithms

Genetic Operations:

- Crossover
  - Swap two subtrees from two different fractals.
- Mutation
  - Fuzz the values of the leaves of one fractal.
- Insertion
  - Replace a node with a small, randomly-generated tree.



Learning a Generative Model: CNNs

Build a collection of “good” and “bad” fractals from training.

With enough pos/neg examples, we can take a Convolutional Neural Net and:

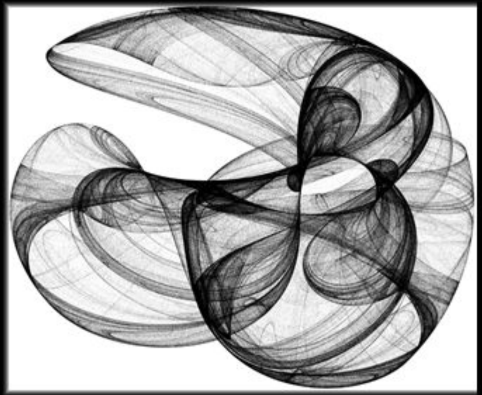
- Train it on ImageNet and other competitive image knowledge bases
- Tune it to learn characteristics of aesthetically-pleasing fractal images

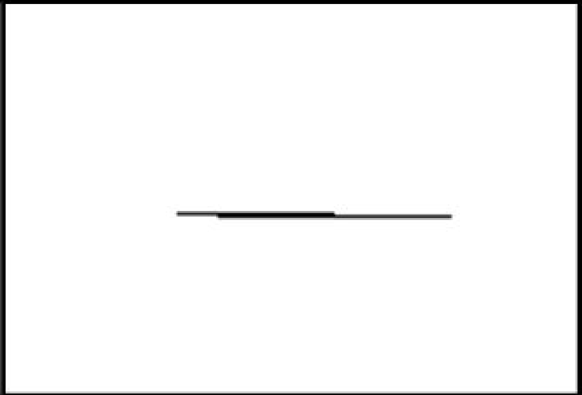
Evolutionary Algorithms + Discriminative Model of Good Fractals
= Generative Model for new, pleasing fractals

Building a Training Set: A Fitness Function

Dense Fractals have both of the following main features:

1. A sufficiently large quantity of distinct points in Cartesian Integer Space.
2. Non-linear figures (e.g. curves or corners)

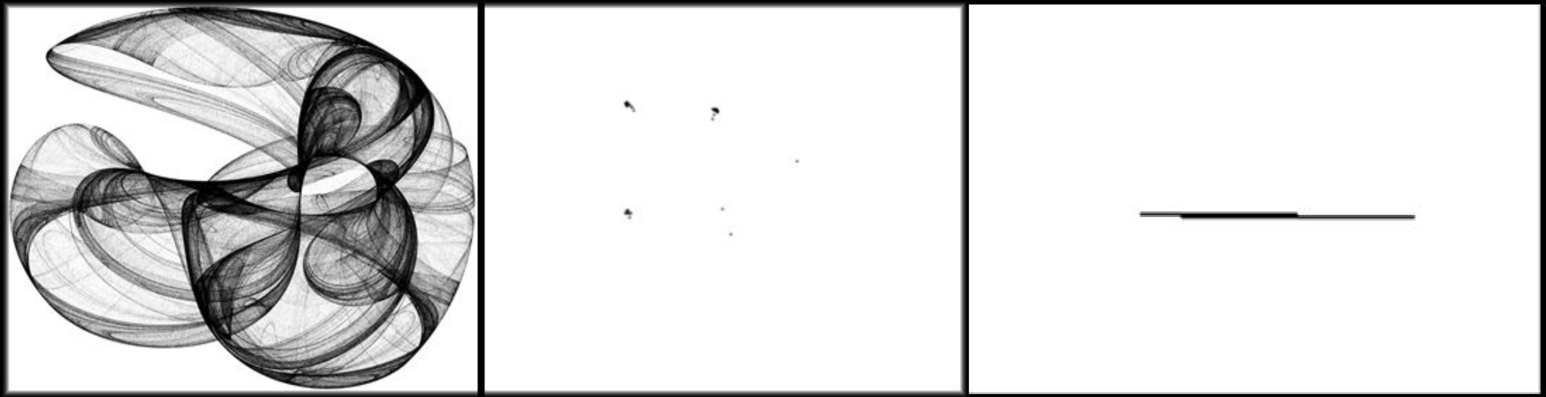




Building a Training Set: A Fitness Function

Computationally cheap metric to measure density?

- File size of saved B/W PNG image.
- PNG encoders compress large regions of one color into a single block.

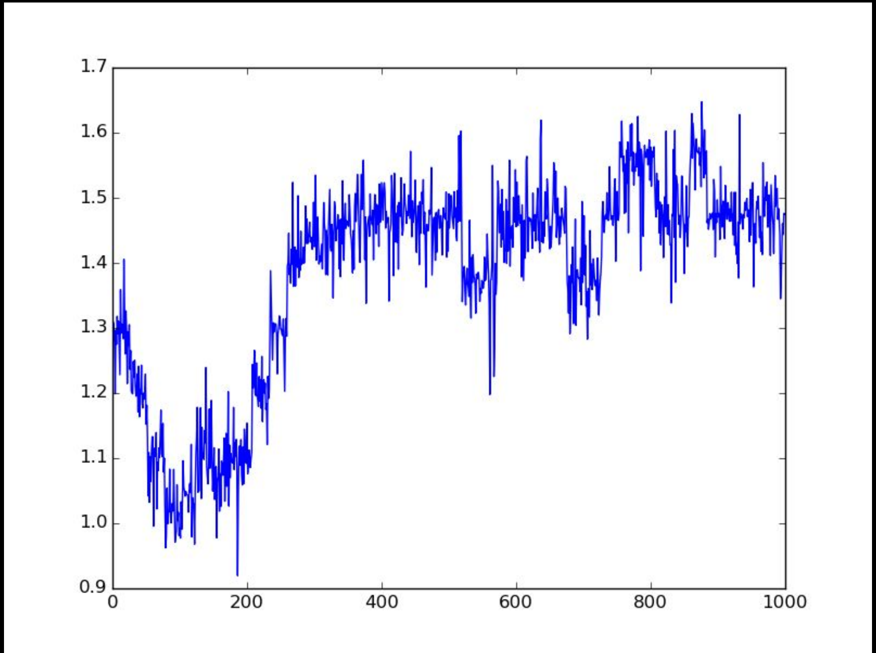


Training a Convolutional Neural Net

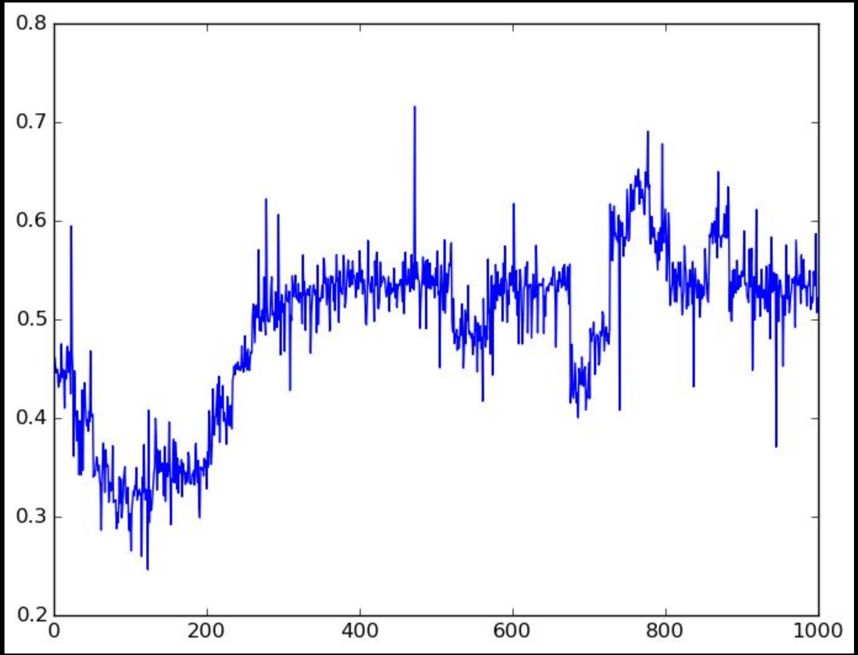
1. Auto-generate 1000 pos/neg training images using file size as an indicator.
    a. > 1.3kb = positive, < 0.7kb = negative.
2. Feed B/W training images into CNN (Clarifai)
3. Auto-generate 1000 new fractals using CNN as a discriminative model.
    a. > 0.6 = positive.

If no positive fractals in a generation, redo the generation.

Observing Results

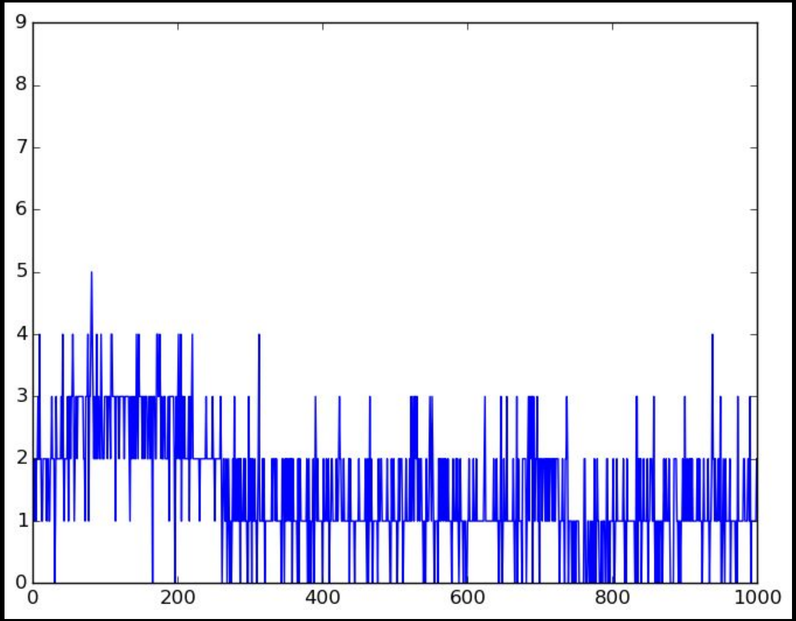


Avg File Size (kB)



Avg Confidence Measure

Observing Results



Number of Sparse Fractals
(file size < 0.7 kB)

Future Work

More Analytics:

- Varying file size cutoffs, confidence cutoffs
- Analyzing file size and confidence trends at a lower level
    - Trends for fractals made from crossover, from mutation, from insertion

Evolution / CNNs with color / RGB equations

Better metrics besides density -- curvature, fairness metric

Subjectivity-based Experiments