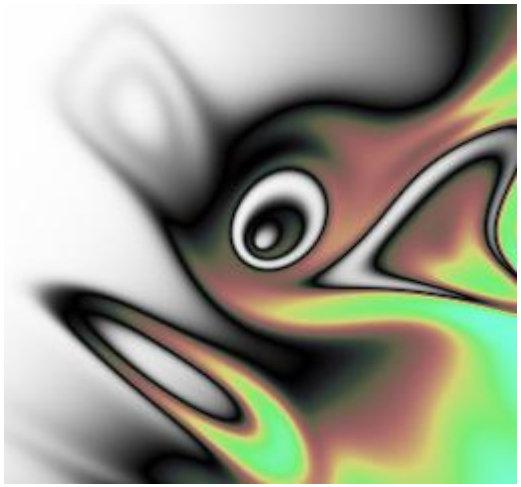
Evolution of Novel Image Deformations

Anthony Wertz and James Schneider

In the beginning...



There was Mr. Quackers (picbreeder)
and it was good

Then

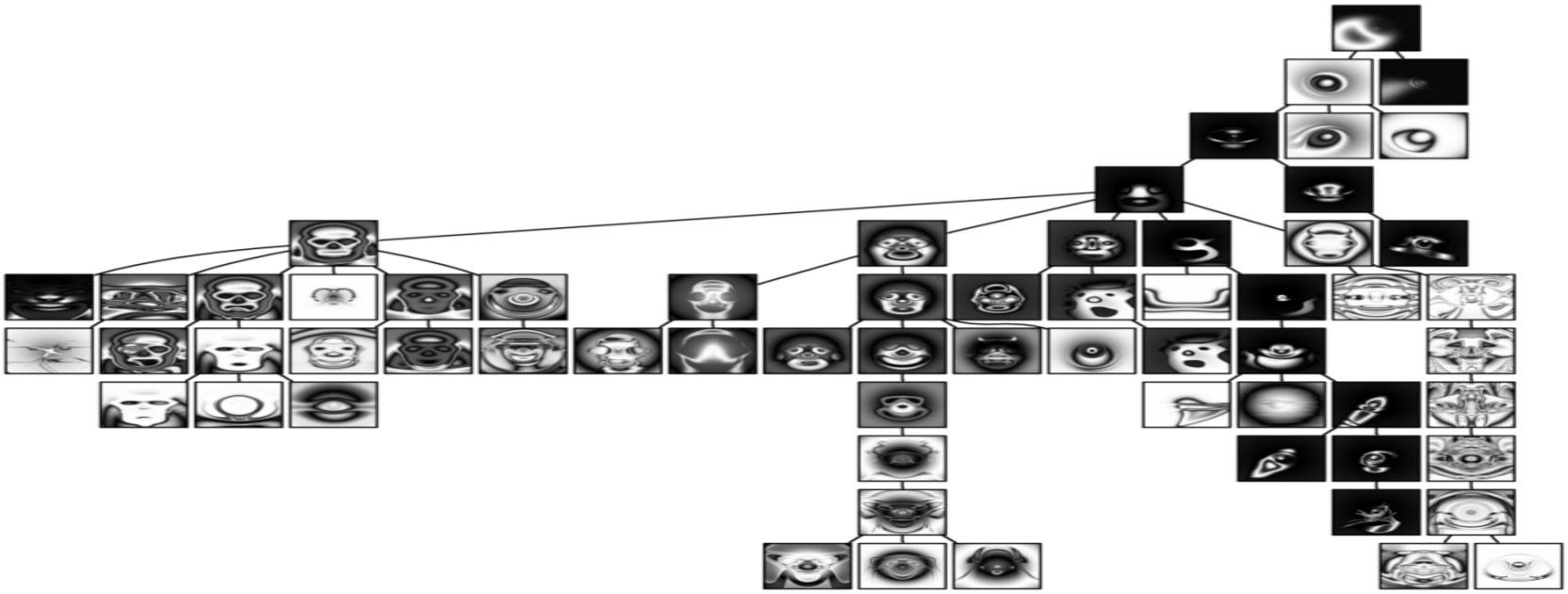
There was a t-shirt evolution
program and it was mysterious

Now

We are proud to present the next
stage in crazy picture evolution

Face deformations

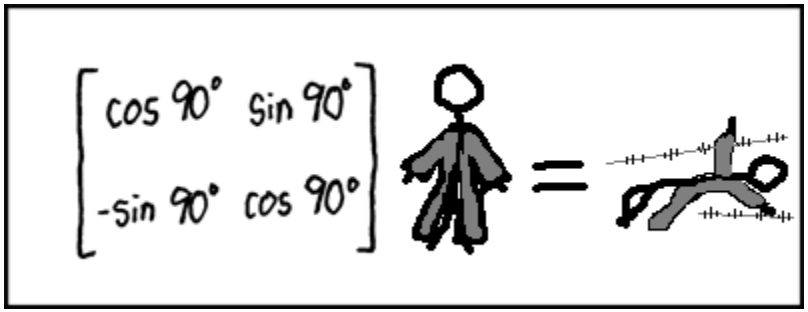
From picbreeder we know that simple functions can be used to evolve and discover new novel transformation techniques



Our idea: what if we applied the same principles just to human faces?

Why?

Since Image transformations are general we know most of the good ones



The question remains are there some good transformations out there that we don't know of?

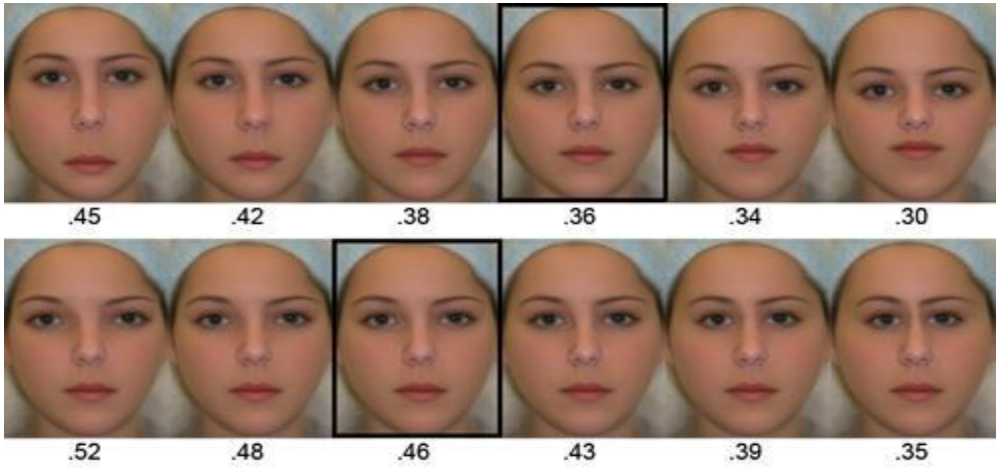
Back of the book dimensionality calculation, assume B&W image and a 256x256 tf matrix

Using $\frac{n!}{(n-r)!}$ there are over 1.65x10^52 different transformations.

Our Observation? - Most of this dimensionality comes from noisy components (random resizing, etc.), allowing only symmetrical components and using the user to evolutionary lines should cut down greatly on this dimensionality

Approach

How would you evolve a unique transformation without sacrificing symmetry?



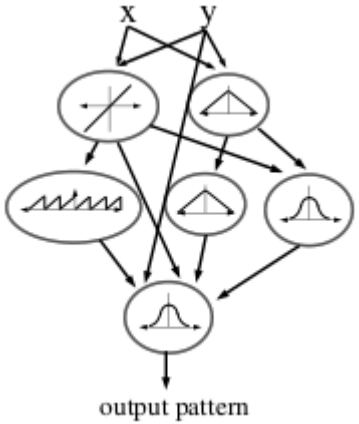
"New Golden ratios for female faces" - Science Daily

Because the search space is so massive NEAT would quickly become bogged down in the weeds. NEAT would have to discover not only what the user likes but also discover the concept of symmetrical faces and facial features. This would increase the search space exponentially and lead to either the algorithm not being able to produce output or the user getting bored quickly of evolving smudges.

Enter the CPNN

Since we know humans like symmetrical faces we can assume those images with large asymmetries will never be picked by the user. This allows us to subtract a very large part of the search space.

CPNN's or compositional pattern-producing neural networks allow us to evolve neural networks with symmetrical activation functions. The CPNN can evolve a set of functions that the user likes and add new nodes that independently apply new functions leading to small incremental changes to the image rather than large jumps.



From "Compositional Pattern Producing Networks: A Novel Abstraction of Development"

Our Architecture

Image Selection
1. Get image from camera
2. Crop Image, and Remove background
3. Convert Image to Gray-scale
4. Pass width and length of image to HyperNEAT

HyperNEAT C++

HyperNEAT Learning
1. Get width and length from picture
2.Similar to XOR experiment, loop through from [-width/2,width/2] and [-length/2,length/2]
3.Set only two inputs and two outputs with a bias
4. Read parameters file (initially this will be normal but eventually it will be able to change it form the GUI)

Python Solver

Python Solver
1. Similar to XOR we will 'evaluate' the experiment in Python
2. Python calls evaluate function, what this does is run the x,y through the network and move the pixel (just gray for now) to a new position
3. Do NOT evaluate fitness until after network is constructed for every member
4. Pass Image Array to Python Interface as a triple array [image_num][x][y]

Python Interface

Python Interface
1. Get Image triple array [image_num][x][y]
2. Display images In a grid with a up arrow and a down arrow below each image
3. User can either up arrow (the image gets a fitness of 10, down arrow image gets a fitness of -10 and a neutral fitness of 0
4. Return double array of [image_num][score]

Python Evaluator

Python Evaluator
1. Receive double array [image_num][score]
2. Set fitness of the members to their scores
3. Send to HyperNEAT C++

HyperNEAT C++

HyperNEAT Learning
1. Create new population
2. Create new species if needed
3. Return new evaluation network

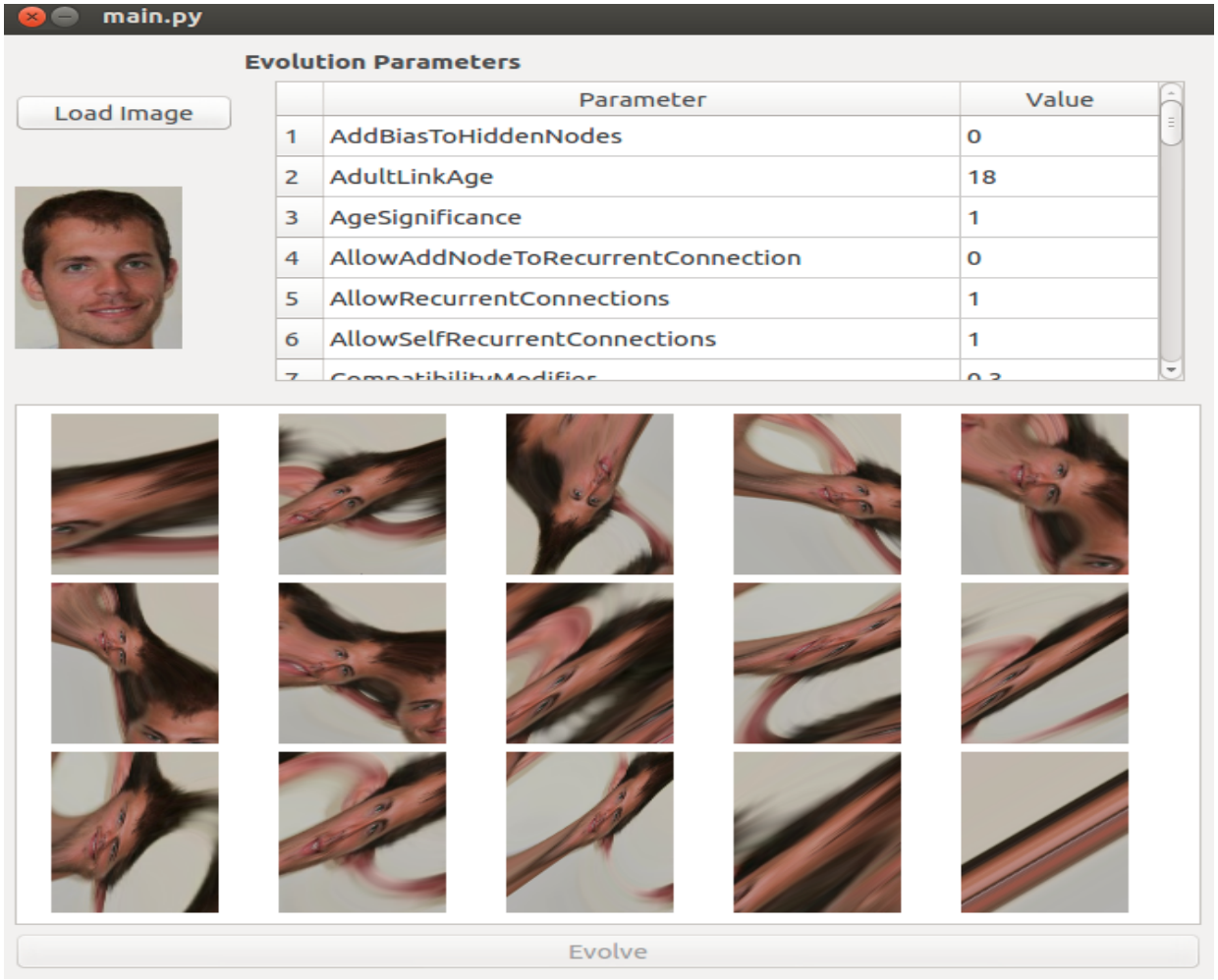
Software used

Jason Gauci's HyperNEAT C++ implementation (heavily modified)

Python 2.7 and a bunch of Python libraries

OpenCV and PyQT for GUI displays

Dev Environment: Ubuntu, VIM, IDLE, Eclipse Cdt, Git



GUI prototype built in pyQT, some basic deformations can be seen in the image

Image of group member Anthony Wertz

A favorite of the algorithm is to output the neck stretching.

Three wise men, this shows that the algorithm can produce symmetry with variation

Algorithm's Favorite output "Elf pez dispenser"

Issues

Why is the elf pez dispenser so popular among the output?

- We found from initial studies that the user of the program will rarely go over 50 generations and that this is due to the algorithm quickly converging.
- This pseudo-random search makes it such that the CPNN's quickly start to converge on an answer as the species start to mimic each other in their genotype but vary in large ways with the phenotypes. Ideally these phenotype variations would be good, but because most of these phenotypes produce such a big distortion effect they are unlikely to be chosen by the human for evolution and left on the cutting room floor thus, the species quickly converge.

How do we solve this issue?

Potential Solution

We must filter!

Two possible solutions are available to us...

Implement “Haar Cascades” to detect facial features. These facial features will be the only ones evolved. The facial features will take the place of the whole image and instead of evolving everything on the image including head position we will just evolve these individual features. Using feature specific information provides our algorithm with apriori knowledge of what humans are interested in thus a sort of filtering effect takes place.

Another solution is to filter based on novelty the champion would be saved, but the rest of the population would be added to a list of known transformations. Then if any future member is near to any member of this list the network is evolved again before being shown to the user.

Potential applications

Iphone/Android app, where you take video or pictures and produce cool effects.

A type of image cryptography, wherein the user is presented with a word and has a specific transformation he/she evolved. This can be like a public key allowing the user to unskew the image and type in the correct input to gain access

Questions?