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