CPPN Based Evolution of Bitmap Facial Distortion

James Schneider
School of Electrical Engineering and
Computer Science
University of Central Florida
Orlando, FL 32816
Email: jschneider@knights.ucf.edu

Anthony Wertz
School of Electrical Engineering and
Computer Science
University of Central Florida
Orlando, FL 32816
Email: awertz@knights.ucf.edu

***Abstract*—This paper seeks to provide an emperical basis to produce novel images from static human faces using NeuroEvolution and Augmenting Toplogies (NEAT) [1] and Compositional Pattern Producing Neural Networks (CPNN) [2]. In this paper new insight is produced into the problems of description of a novelty operator [3] and solution to the problem of user fatigue [4]. Through the NEAT/CPNN process the algorithm has discovered increasingly preplexing and interesting results that may hint at some deeper meaning and understanding of how image processing is done. This paper explores those interesting results and presents interesting avenues for further research.**

I. Introduction

The diversity that appears on planet Earth is broad but reuse is a common trait among all animals. Each face is complex in its own right, but what is the ultimate limit in possible facial configurations? Each facial structure is evolved to attract mates and/or optimize placement of sensors to increase survivability. The possible configurations of faces though is limited by multiple factors such as weight, surface area, etc. these factors place an upper limit on the possible faces that can be seen on the planet Earth. These optimization factors combined with the non-random starting point of currently available facial structures converts an immeasurable problem to one of surprisingly low dimensionality [5].This paper will limit the discussion to Human and Human-like faces in order to constrain the produce empirically verifiable results.

Human like faces are interesting in that they represent the property of repeatability with variation [2]. Although most animal faces do carry these same attributes, its inherently easier for a Human to identify and understand facial features and structures. This process is mostly due to facial recognition circuits buried deep within the Human brain that have evolved for quick unsupervised feature detection and identification of fellow Humans. This understandability of the interestingness of evolved faces directly ties to the ability to quickly verify results obtained by the algorithmic process and is a cornerstone to this paper's research.

The overall goal of this paper is to produce new plausible facial structures and explore how these facial structures evolve.Although, as mentioned earlier the dimensionality for human faces is bounded, but this bound is unexplored. Taking this into consideration this paper will utilize a searching algorithm in order to search for new and novel faces. There are multiple searching algorithms in practice, but most of them require some form of objective function, and it is currently unknown how to objectify interesting faces. Taking this into account this paper will utilize a user guided search where the individual determines what is interesting. This though is not enough as the user may not know what is interesting until it is presented. The algorithm could be a random search, but due to the size of the search space it may take a user years of clicking in order to find interesting results.

To solve these issues the algorithm that was used is Neuroevolution and Augmenting Topologies [1] hereby referred to as NEAT. NEAT allows for directed evolution of neural networks towards a goal using the concepts of speciation (evolving new species), historical markings, and directed complexification. NEAT utilized with Compositional Pattern Producing Neural Networks [2] and interactive evolution in order to produce more lifelike and non-objective driven images. This paper will seek to provide evidence that not only can interesting images be found by Humans but these can be found with great efficiency, and that in the process of exploration a deeper process starts to unveil itself.

This paper will provide evidence that evolution of images need not only a directed search but an understanding of the domain in order to correctly capture interesting facial structures. Exploration of the process and results of the evolution will take place in this paper including examination of interesting results gleamed from this exploration process. These interesting results hint at some deeper understanding of not only what Humans find interesting but how Human interaction changes the exploration/exploitation paradigm. This paper along with these theoretical underpinnings explores a new image evolution architecture that produces more varied results and allows for faster exploitation of those figures and features found in the exploration process.

II. Background

Neural Networks first appeared in literature as early as the 1960's, but until NEAT was introduced a user had to manually estimate the number of nodes needed to solve the proper and how these nodes were connected. In addition the network required retraining every time new data was presented and failed to properly react to a changing objective function. These issues were solved with the introduction of NEAT in 2002 [1]. NEAT evolves not only the weights of a network but

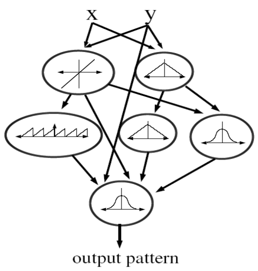


Fig. 1. CPNN Graph Abstraction

A graph abstraction of a compositional pattern producing neural network. The CPNN abstraction can be applied to a neural network type structure as the graph depicts [1]

the placement of linkages and nodes inside of the network. This process is governed by a memorization genetic algorithm that seeks to optimize a user's fitness function. The major advantage of this system is the protection of innovation via speciation. Speciation allows novel structures to survive long enough to eventually find a more optimal solution to the user's problem. This is important to the facial deformation problem as novel facial structures would be lost in any algorithm that vigoursly explores the search space. Introducing the concept of speciation along with fitness sharing smooths out search space and allows semi-guided search to take place. The biggest issue with NEAT is that it still requires some user input in the form of an objective. This domain input is needed in order for NEAT to decide if a structure is more correct than another, but this function describing images is currently not known.

Due to the limitations of the NEAT process a second abstraction must be introduced in order to properly solve the facial deformation problem. CPNNs or Compositional Pattern Producing Networks are indirect encoded developmental networks that describe in a minimal set how something is created [2]. CPNNs eliminate two limitations the first being a uniform activation function, and the second is sampling among a continuum rather than discrete points. Because CPNNs describe how a function evolves rather than how to evaluate a function CPNNs represent a different structure then that of artificial neural networks but one that can utilize the same structure of the networks II. The combination of NEAT with the CPNN abstraction allows for greater ability to come up with a generational model for how a function develops [2].This process also allows for simplification of the fitness sharing metric involved in speciation of the neural networks.In-lieu of a domain restricted optimization function the distance between the two networks can be used to properly speciate the evolved networks [2]. Another fortuitous advantage of the algorithm is due to the domain of images, as most images change in resolution and density more input neurons are not required as would be under traditional neural network architectures. The CPNN ignores the resolution of the image and can generalize how the image is produced thus saving valuable computational time.In addition, the process of including symmetrical activation functions with a generative process allows exclusion of large amounts of the search space allowing more efficient search for interesting artifacts. To this end the CPNN/NEAT architecture allows interesting facial deformations to be discovered by the user in a timely and rigorous manner.

There still exists a issue to implementation of the algorithm and that is the objective function.Interestingness of a Human face is difficult to express in mathematical terms. Due to this limitation evolving interesting deformations with CPNN/NEAT is impossible. There exists another abstraction that of Interactive Evolutionary Computation or IEC, IEC supplements a mathematical objective function for one driven by a Human operator.Instead of performing a straight optimization route on images, Humans perform a meta-heuristic like search of the novel spaces which encodes the optimal function this performs a search for interesting or novel entires [3] [7]. The CPNN/NEAT architecture has already been applied successfully to this domain with the inclusion of IEC [4]. PicBreeder allows for anyone with an Internet connection to evolve images from simple functions into genetic art. The users of the site effectively perform a modified Gibbs sampling algorithm on the search space of interesting images. This gives the PicBreeder program insight into the higher dimensional fitness function will still allowing CPNN/NEAT to evolve the underlaying image representation. PicBreeder becomes a lesson in how to efficiently search through possible images, and its process becomes a cornerstone of the facial deformation discovery process.

III. Approach

The CPNN/NEAT architecture is at the forefront of the facial deformation discovery process. The architecture takes advantage of developed software suites most notably HyperNEAT C++ [8]. HyperNEAT C++ is a complete version of NEAT, CPNN/NEAT and HyperNEAT that was suited to prove the validity of the HyperNEAT platform. Extensions to this original program had to be made in order to tailor it to the field of facial deformations. HyperNEAT C++ was extended with the Python programming language in order to create easier accessibility to the underlying algorithmic process. The architecture presented in this paper is simple and allows for quick evaluation of image novelty by the user. The simple architecture 3(a) allows for recursive processing of information in order to process not only static images but videos and allows expansion to include more complex deformations including non Human faces.

The major issues with the architecture became obvious when running, the most glaring issue was that of user fatigue. Cited in many major publications user fatigue is the anomaly that occurs when users are presented with a long search process [4] [7]. Due to the constructive nature of genetic art in that each part or symmetry is discovered in an orderly process and innovative structures are seldom seen by the user. Users frequently expect the innovative images to be produced immediately, but this is inhibited by the expansive search space and no knowledge of what constitutes an mathematically

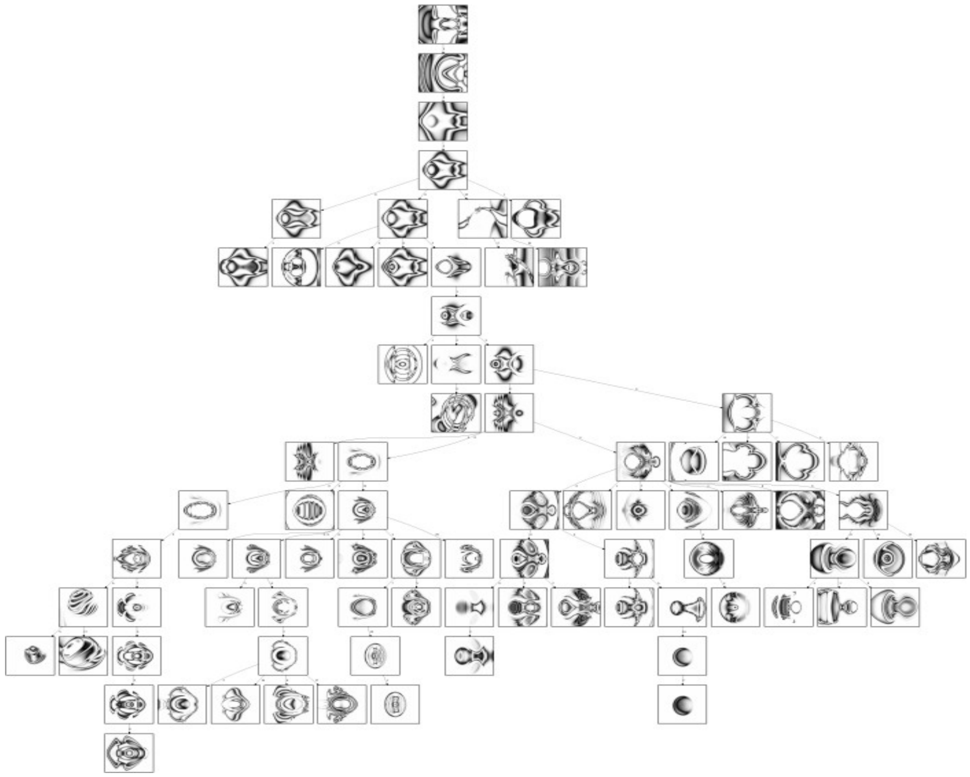
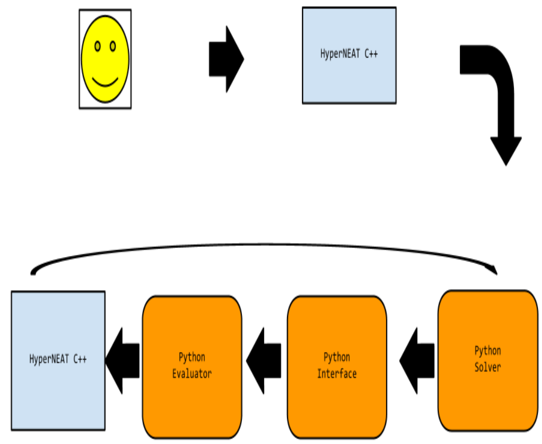


Fig. 2. PicBreeder
A graph abstraction of a compositional pattern producing neural network. The CPNN abstraction can be applied to a neural network type structure as the graph depicts [6]

interesting image. This problem is further exacerbated by the presence of noise in the genotype to phenotype representation. This noise takes the form of large changes to the underlying neural network structure producing relatively small changes in the images. These small changes take the form of rotational and translational image transformations and represent the phenotype of the large change in the genotype. It was discovered that these jumps occur more at lower complexity levels than at larger levels and occur with a relative frequency. The combination of user fatigue, nonlinear mapping, and low complexity leads to uninteresting images being produced by the algorithmic process. It is theorized that these issues would be solved given more users with programs such as PicBreeder, as the process can be parallelized and utilize user histories to solve not only the fatigue issue but also the complexity issues [4].

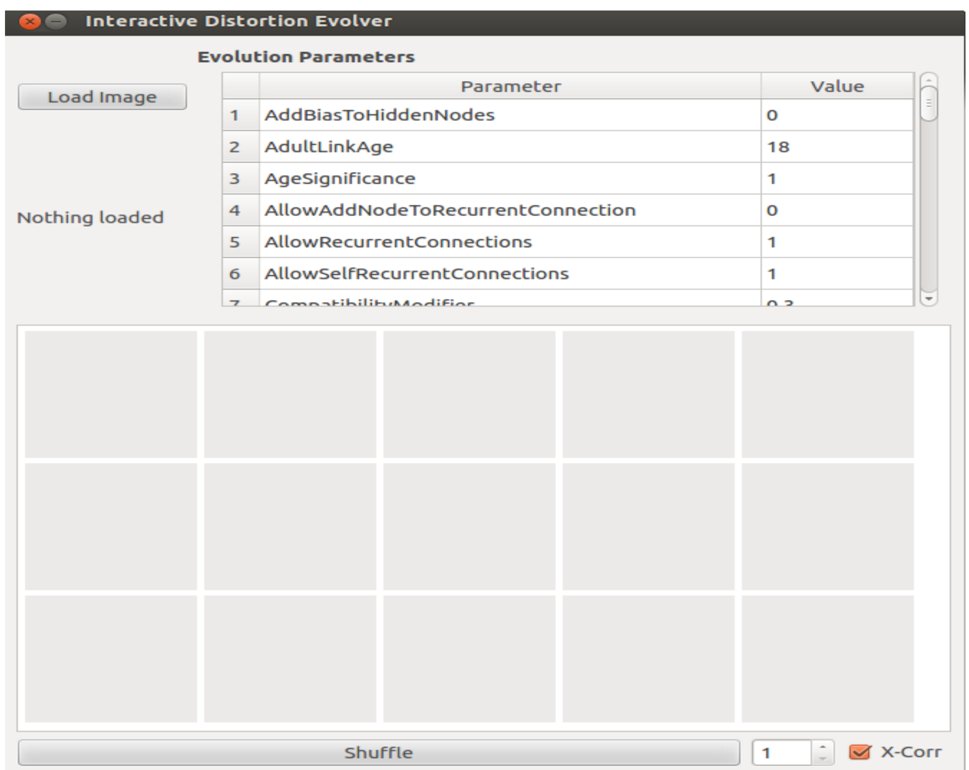
Unable to acquire more users these issues had to be addressed in order for new novel structures to be discovered by the algorithmic process. Using the insight that the novel structures represent something that has not been discovered before a novelty like search algorithm can be implemented [3]. The biggest hurdle is to discover a function of novelty of an image, this was discovered by accident through observation. Using a image histogram a simple entropy calculation can be performed. This histogram is then averaged over the mean and standard deviation of all images in the historical novelty record to discover the offset the image represents. A simple threshold of a correlation matrix is then used to determine if the image is novel or not and whether to present this image to the user. This threshold is troublesome as small changes in it can create large differences in the number of novel images selected to be presented to the user. To further solve this problem a W3C luminance calculation is performed which transforms the red,green,blue image which varies from [0,255] to luminosity (contrast) values between [1,21] this smaller variation value allows less emphasis to be placed on threshold placement [9].Image entropic value estimation though does not solve the rotation issues present, but through empirical analysis it lessens the impact, due to usage of an offset rather then a direct value evaluation. Novelty search in combination with image entropy metric allows a solution to user fatigue to become present as non-novel images are greatly reduced. Novelty search solves the user fatigue issue but does not solve the complexification issue that remains.

Even with the addition of novelty search the initial search process starts out very slow as the network builds complexity. To solve this a shuffle and randomization ability was created to boost the algorithm to a higher dimensional space quickly without user interaction. A Gaussian distribution is used with which new random evaluation values are given to novel images in the search space. The Gaussian distribution was found to be key as the algorithm would become confused if on one



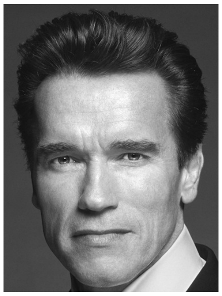
Deformations Architecture.png Deformations Architecture.png

(a)



(b)

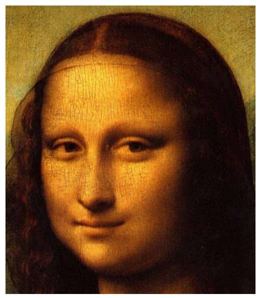
Fig. 3. Facial deformations architecture and user interface for interactive evolution.



(a)

(b)

(c)



(d)

(e)

Fig. 4. Face images used in the experimentation.

generation an image became very highly valued and the next was considered junk due to the randomness. Having a smooth random fitness function for the population members allows each species to equally have a chance to become complex while stopping confusion as the fitness gradually levels off with time rather than be subject to computational randomness. The complexification of these initial functions allows for user fatigue to be further curtailed and the novel facial deformations to be discovered by the user.

IV. Experiments

The experiments run utilized five different face images: three quite different real faces, one artist's rendering of a face, and one unrealistic cartoon character's. The majority of these faces are quite recognizable to an English audience, as is probably apparent in figures 4(a) through 4(e).

The diverse image set was chosen specifically to observe the effects of single distortions across a variety of faces. In many cases the same effect is achieved, though in some cases some quite diverse effects are seen (especially when dealing with the oddly proportioned cartoon figure). Of course the same mathematical function is being performed but the proportions of the face along with its structure have an impact on what the human observers interpret to have happened to the face.

Generally for each image the interactive evolutionary process is conducted by the tool operators (independently). The different faces are evolved individually to determine what effects a distortion chosen for one image has on the rest of the set. Throughout the evolution there were a few tools at the disposal of the operators. The most intuitive feature allows the operator to select the favored image or images to be evolved, which is followed by a single generation evolutionary step. This is a fairly slow heuristic compared with those used in many other evolutionary applications but it allows a tailored approach to the process.

In some instances the population might be highly homogeneous with only slight distortions between images. If this is not desirable, the cross correlation boost method can be used which will, over some user-specified number of generations, select images from the population which are not highly correlated with others in the selection. For the correlation the brightness map of each image is calculated and for each image it is determined whether it will be added one to the selection. Any given image is only added if there is a large amount of dissimilarity between it and the rest of the elements in the selection which is determined by a two dimensional correlation with each image already in the selection. This does a reasonable job of adding some additional diversity in the set. In cases where the distortions are changing quite slowly this can be helpful in acting as a spring-board for larger changes.

One pitfall of the cross-correlation approach is that it's quite slow, primarily because the output images have to be calculated for each generation. This process is sped up considerably as the evolutionary application uses scaled down versions of the original image, a process made possible by the use of a normalized image coordinate space, but the correlation and output calculation especially still take a noticeable amount of time to compute. The cross correlation method is useful to gain some diversity, but if a boost in complexity is desired the uncorrelated boost feature can be used which does not correlate any images, nor does it ever even calculate the output of the networks for all but the last set to be displayed to the operator. In this approach, members of the population are randomly assigned a fitness. The benefit here is that a large number of generations can be executed extremely which can be useful in order to gain complexity (or potentially to remove complexity, depending on how the evolutionary parameters are configured). The pitfall is that the fitness of the individuals is determined randomly so they're not always desirable. Cross correlation does a better job at adding momentum to the changes in a homogeneous set while uncorrelated boosting shuffles the phenotype space.

As alluded to, the operator is able to modify the NEAT parameters at any time to affect the evolutionary process. If more complexity is desired, for example, the node or link creation parameters may be increased. If less complexity is desired, the same parameters may be decreased along with the increase of the node and link destruction parameters. The experiments run for this project included a variety of the methods described above.

Once a favorable distortion has been evolved, the distortion is stored then applied across all images in the set for comparison.

V. Results

As mentioned in the previous section a strictly interactive evolutionary approach was not adhered to: there were a few tools at the disposal of the operators. However the initial results somewhat necessitated these changes, an observation that will be described shortly. Figures 5 to 8 show four examples of interesting distortion generated in the evolutionary process.

Figures 5 and 6 are examples of interesting distortions attained prior to the implementation of the boosting process. The first, the “big eye” distortion, creates an odd elongation of the eye. This is one of the cases mentioned previously where the results, while performing the same operation, yield different artifacts per each image. Notice on figures 5(b), 5(c), and 5(e) it would appear that each individual’s left eye is elongated, though on 5(d) and 5(f) it would appear to be the right. Just given the different head geometry and orientation the essence of what distortion is taking place is fundamentally altered. This is something which could perhaps be fixed by first warping the image into a predefined coordinate space which will be discussed later.

Figures 6(b) to 6(f), victims of the “multi-eye“ distortion, are very interesting for a few reasons. First, in general it’s a quite odd warping that has occurred. The face of the individuals seems repeated and warped in a very fluid manner yielding nicely connected mouths and sets of eyes. It’s oddly comforting to see the features flow together to form a face that could be somewhat realistic, though far-fetched, with it’s flowing hairlines and its oddly shaped but not completely disjoint features. But what seems most interesting here is the warping to the cartoon character 6(f). It wasn’t surprising to see the combination of multiple faces in the same weaving fashion observed in the other images. What was surprising is the seeming lack of blending and well defined features. On the lower three-eyed face, for example, it was not at all expected that the eyes would be three independent clearly defined eyes as opposed to a blur as in the other faces. This is a feature consistent across all distortions of the cartoon which seems to be primarily due to the limited palette used in the image so very little blurring actually occurs. This has two interesting applications: first, the face evolution can probably be used with great success to form new cartoon character morphs because definition is maintained and limited blurring occurs; second, a realistically proportioned cartoon of a face can be used to see with greater ease exactly what the distortion is accomplishing, which might be useful in the search for generic interesting distortions.

The last two distortions seemed fairly interesting. The issue prior to incorporating the boosting though was that complexity didn’t grow very well without a large number of generations. Even with a number of generations many interesting networks had little or no hidden nodes (the previous distortion networks figures 5(a) and 6(a) had none). As long as interesting distortions come out this isn’t a huge issue but the lack of complexity was yielding a constrained set of distortions that reappeared on a run to run basis. The first resort was to meddle with the complexifying evolution parameters (node and link creation) but this resulted in quite a bit of parameter ”thrashing“ as the parameters would need to be modified sufficiently high or low to provoke or inhibit complexification. The boosting method seemed to offer a good approach allowing rapid complexification when needed and fine tuned adjustments when it wasn’t needed.

The distortions represented in figures 7 and 8 are more complex examples. By comparing their networks (figures 7(a) and 8(a)) with those of the last two examples (figures 5(a) and 6(a)) the complexity difference is obvious. It’s also obvious when evaluating the distortions that arise as a result.

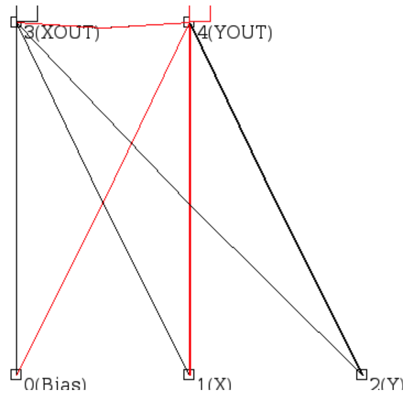
Figure 7, the ”Gaussian V“, has a quite large network for a distortion that seems fairly fluent and not terribly difficult to explain. For the real faces and the painting it pretty consistently spits out a two headed amalgam backed by a very interesting ”V“ shaped texture. Again, what’s really interesting though is the cartoon figure where the distortion, due potentially to the lack of refined detail and the exaggerated features, generates something more a kin to a new creature than the old one somewhat warped. The cartoon has turned in to some sort of bug or alien creature. This is a perception that’s completely different than what’s gathered from the same distortion applied to the other images.

The final distortion depicted in figure 8 is a fairly simple network that creates from an image of a face a linear texture using the colors extracted from the original image. Some how this network seems to have grabbed a single column from the original image, flipped the upper and lower halves, stretched one and compressed the other, then tiled it across the image. This is done with two sine nodes and one linear node. It’s a pretty interesting outcome that doesn’t seem at all intuitive given the network. It does serve to demonstrate the inherent difficulty in understanding CPPNs based purely on their makeup, even simple CPPNs with three hidden nodes.

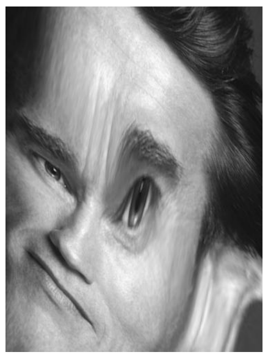
VI. Discussion

The experimentation and results yielded some useful findings regarding the work and its usefulness.

First, sending in raw face images alone without any uniformity in or designation of features (such as the head dimensions, eye sizes and locations, et cetera) will not necessarily yield similar results across multiple faces. There were some examples presented showing different eyes elongated for example. Given the networks are configured to flow from the output destination to the source pixel location there is no coherent way to indicate the location of features since those are not known until the deformation is performed. One possibility to look at is reversing the flow of information so that the source



(a)



(b)

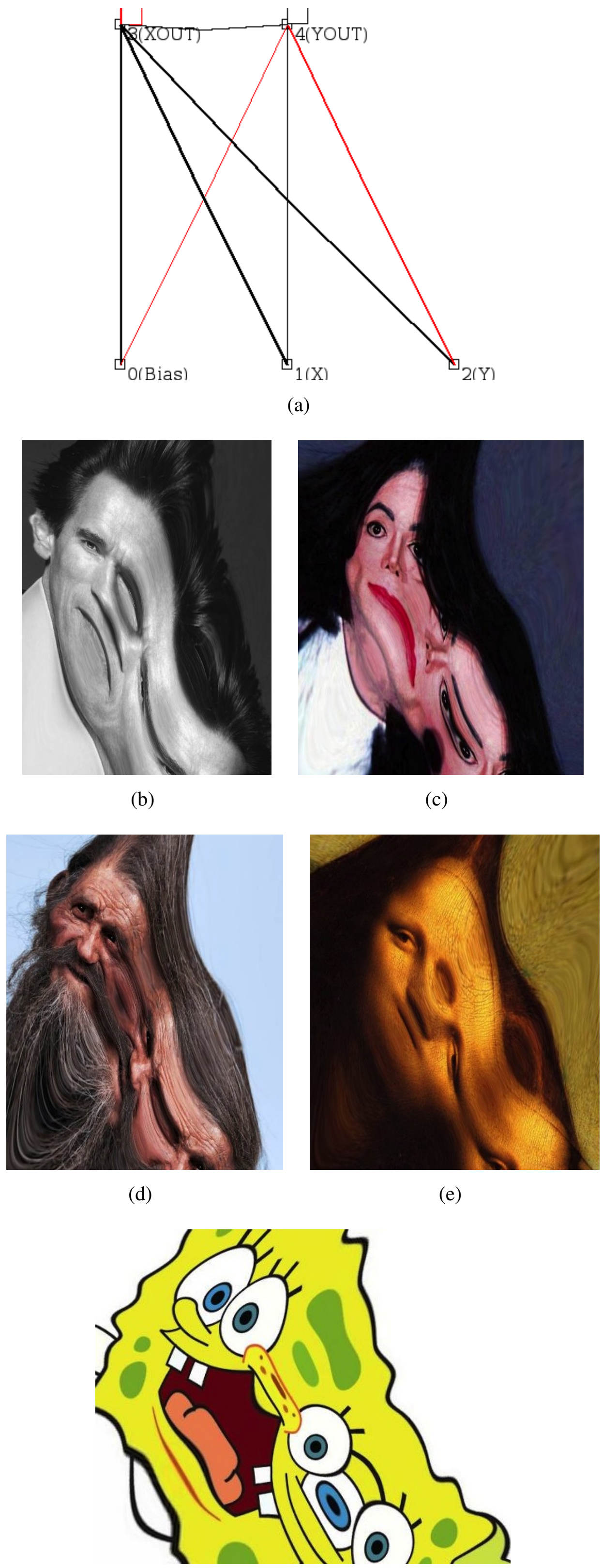
(c)

(d)

(e)

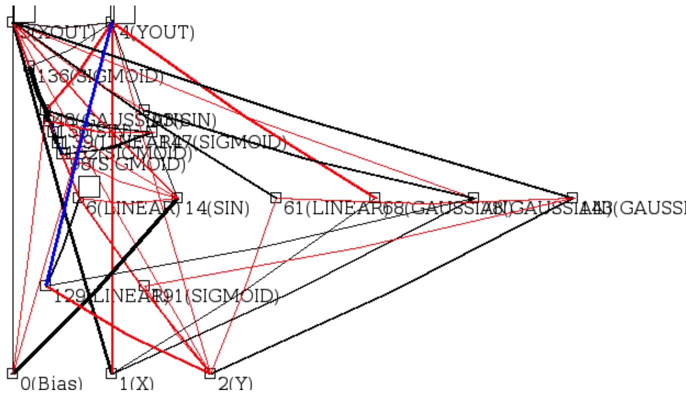
(f)

Fig. 5. Big-eye distortion: 5(a) the network and 5(b)-5(f) the distorted face set.

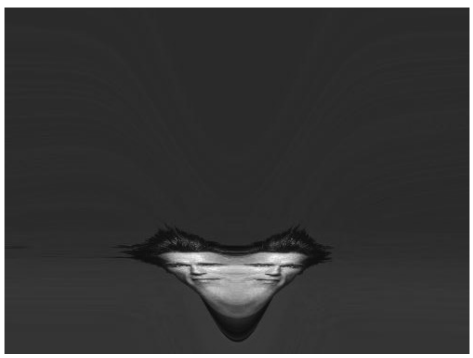


(a) (b) (c) (d) (e) (f)

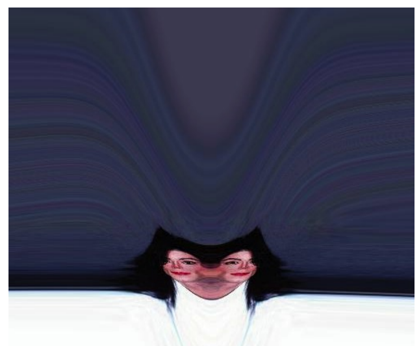
Fig. 6. Multi-eye distortion: 6(a) the network and 6(b)-6(f) the distorted face set.



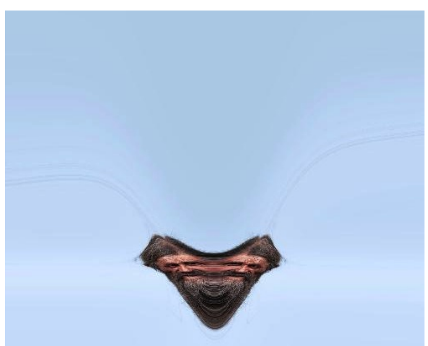
(a)



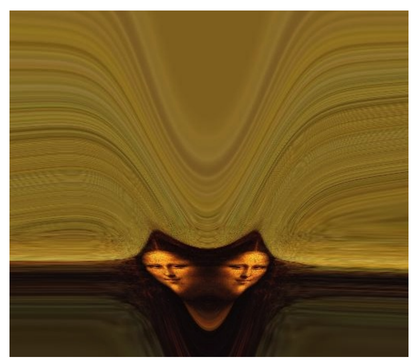
(b)



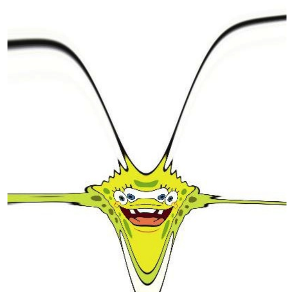
(c)



(d)

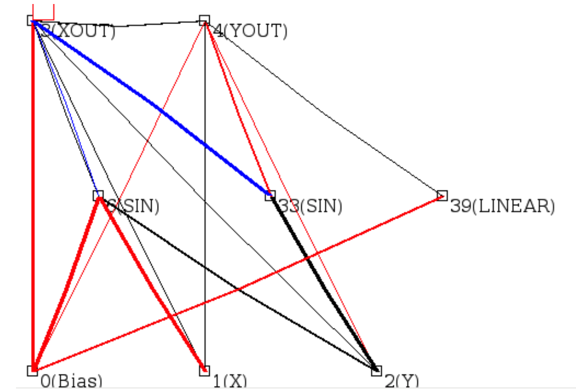


(e)

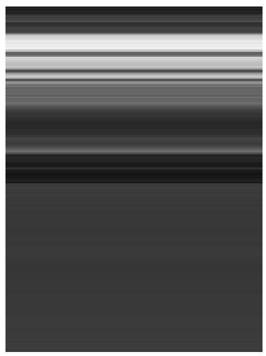


(f)

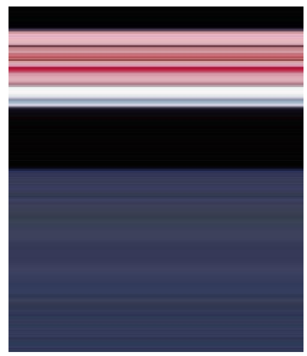
Fig. 7. Gaussian V distortion: 7(a) the network and 7(b)-7(f) the distorted face set.



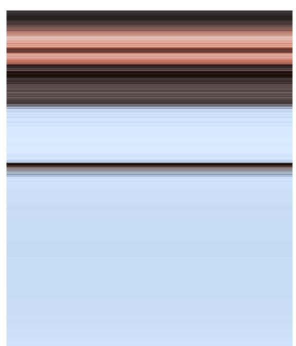
(a)



(b)

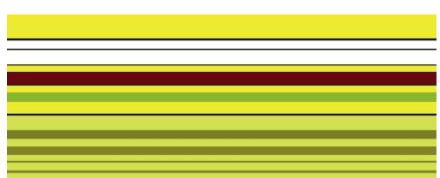


(c)



(d)

(e)



(f)

Fig. 8. Color extraction distortion: 8(a) the network and 8(b)-8(f) the distorted face set.

locations are input and the destination locations are output. This may cause some issues with sparsely populated image generation which, depending on the distortion, may be difficult or even impossible to fix with blending and interpolation techniques. Another possibly more practical method might be to detect the features in the face image then allow warping to be done on the features individually and recombined to form a face. This has the disadvantage of compartmentalizing individual components though.

Second, interesting results can be obtained with very little complexity but the results tend to converge to similar designs. Even a few hidden nodes can greatly increase the diversity in the results. But to get past this initial no diversity hump the boosting effort seemed fairly useful. Another option might have been to simply start the network off with a few hidden nodes but that approach requires some assumptions in determining what might compose a desirable network, a task that's fairly complicated even with few nodes. For example, is figure 8 desirable? One may suppose it depends largely on the application; for general face deformations it's probably not very interesting. It's difficult to deduce beforehand that two sine nodes and one linear won't make something interesting and it's difficult to deduce that they won't ever create something interesting. The other aspect of boosting that is of interest is the part played by the cross correlation. It seemed to perform well at discriminating close to duplicates but they had to be very close for it to be useful. This may very well not be the best similarity indication available. Even if it were there would need to be large speed improvements made before it could be considered practical. This may very well come out of simply moving it away from Python into C or C++.

Third, the technique as it is currently implemented may have limited practical use on real images or detailed drawings but it seems to have some clear applications in less detailed use cases. The cartoons generated, for instance, show some potential of the system to evolve cartoons and, if expanded into a three dimensional space, perhaps also structures. This could be an interesting addition if trying to generate unique user-generated data (in games, for example).

VII. Future Work

Curing cancer, world domination, etc

VIII. Conclusion

Fin.

References

[1] K. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.

[2] K. Stanley, "Compositional pattern producing networks: A novel abstraction of development," *Genetic Programming and Evolvable Machines*, vol. 8, no. 2, pp. 131–162, 2007.

[3] J. Lehman and K. Stanley, "Efficiently evolving programs through the search for novelty," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. ACM, 2010, pp. 837–844.

[4] J. Secretan, N. Beato, D. D Ambrosio, A. Rodriguez, A. Campbell, and K. Stanley, "Picbreeder: evolving pictures collaboratively online," in *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. ACM, 2008, pp. 1759–1768.

[5] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *JOSA A*, vol. 4, no. 3, pp. 519–524, 1987.

[6] "Picbreeder website based on work by dr. kenneth o. stanley," http://picbreeder.org/behind.php, accessed: 12/01/2012.

[7] Y. Li, C. Hu, and X. Yao, "Innovative batik design with an interactive evolutionary art system," *Journal of Computer Science and Technology*, vol. 24, no. 6, pp. 1035–1047, 2009.

[8] "Hyperneat c++ source code," https://github.com/MisterTea/HyperNEAT, accessed: 09/10/2012.

[9] "W3c luminance calculation," http://www.w3.org/WAI/GL/wiki/Relative_luminance, accessed: 12/01/2012.