Deep Interactive Evolutionary Computation

Philip Bontrager
philipjb@nyu.edu

Wending Lin
wendinglin96@gmail.com

Sebastian Risi
sebastian.risi@gmail.com

Julian Togelius
julian@togelius.com

Abstract

This short paper outlines our ongoing work in combining generative adversarial networks (GANs) with interactive evolution. While GANs can be trained to produce lifelike images, they sample randomly from the learned distribution, providing no control over the results. Interactive evolution has been used to create various artifacts such as images, music and 3D objects, but relies on having a good evolvable representation of the target domain. We use the trained generator part of a GAN as genotype-to-phenotype mapping, allowing for controllable high-quality image generation.

1 Introduction

Interactive evolutionary computation (IEC) is a powerful approach for general mixed-initiative design [8]. In IEC, an evolutionary algorithm uses one or several human beings as the fitness function(s). The algorithm allows people to explore a large range of ideas, while a person allows the algorithm to evaluate qualitative properties. Picbreeder, EndlessForms and MaestroGenesis are a few examples of recent systems that use IEC to allow people to create images, shapes and music respectively [7, 3, 5].

While these works are impressive, they don't allow people to express a specific idea that they start with in their mind. The complexity of the image representations and the limited time of a human evaluator makes it difficult to search for a specific design. In this paper we present Deep Interactive Evolutionary Computation (Deep IEC). Deep IEC uses a deep neural network (DNN) to constrain the search space of the evolutionary task. A DNN is pre-trained on a particular domain of data to be able to generate similar data. The person and the evolutionary algorithm then search the DNN for a design of interest. This allows Deep IEC to quickly produce highly detailed content.

2 Content Domain

We decided to produce 2d images as there is a lot of recent advances in using DNNs to generate images. We looked for datasets that represent domains that are interesting to design for. In our initial setup, we use three datasets: CelebA face dataset [6], UT Zappos50K shoes [9], and 3D Chairs images [1]. A separate DNN is trained for each domain.

3 Architecture

The algorithm consists of two main parts; the DNN and the IEC algorithm. The DNN can in theory be any generator network that translates a small set of latent variables into an image. The evolutionary algorithm is then evolving the latent variables and the person is selecting their favorite images.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

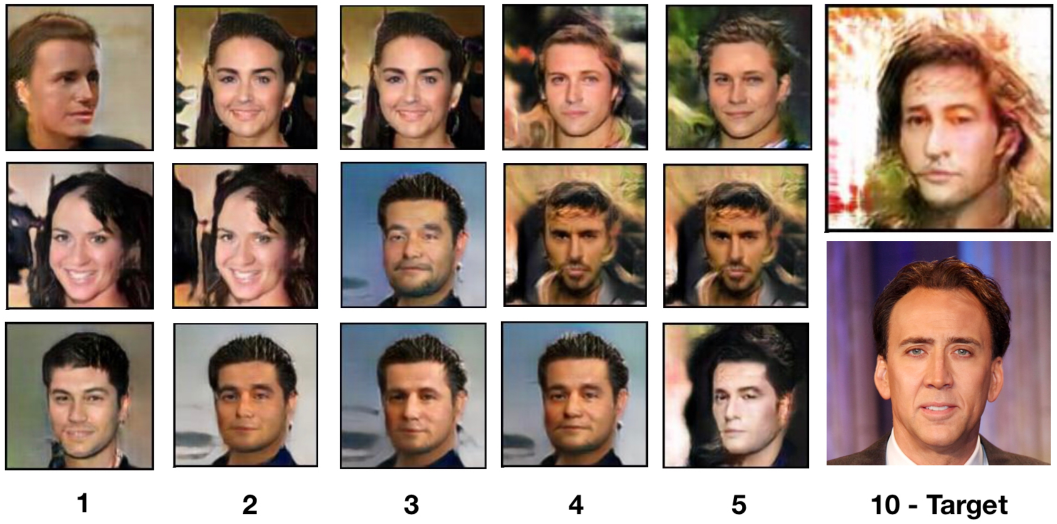

Figure 1: This demonstrates evolving an image to look like Nicolas Cage. Each column contains the selected images for the given generation. While three images were selected each time here, that is not necessary. The last column shows the best image after 10 generations.

Deep To generate our content we used a Generative Adversarial Network (GAN). A GAN consists of two networks, a generator network and a discriminator, pitted against each other. The discriminator learns to discriminate between images in a dataset and the output of the generator. The generator learns from the discriminator how to fool it, indirectly learning to mimic the dataset. The recent WGAN-GP variant has shown success in generating relatively large images [4].

Our network has 20 latent variables and outputs a 128×128 square image. In a recent work, Berthelot et. al. found that the number of latent variables did not have a big impact on image diversity [2]. Keep the number of latent variables low reduces the number of variables to evolve.

Interactive As Deep IEC is an interactive system, it requires an interface. The interface starts by letting the user select from a list of the available domains. The user is then presented with a single generation of images. The user selects their preferred images and presses next. They can also select how large the mutation is via a slider. This process is repeated until a satisfactory image is found.

Evolutionary Computation To focus on the result of this new combination, we implement a simple evolutionary algorithm. One of our variation operators is crossover, meaning we take pairs of selected images and randomly select from their corresponding latent variables to make a new set of latent variables. While the latent variables to the GAN we used are not independent variables, in our observations they did show some independent characteristics. This warranted using crossover to randomly select features from each parent, albeit not in a predictable or consistent way.

The population size is 20. For each generation the selected images are brought forward. Two additional images are randomly generated to inject diversity into the population. To rebuild the rest of the population, pairs of the original selection are sampled with replacement for crossover. Every member of the population has a 50% chance of mutation which is a Gaussian noise with a standard deviation equal to the users setting, between 0 and 1.

4 Demonstration

In figure 1, the progression of evolution can be seen. For each generation, images that contain desirable properties are selected, even if they look nothing like the intended goal. Over several iterations, the traits of the selected images come together. Faces are demonstrated for analysis to show the system on a domain where we are very perceptive. It is very easy to apply this system to other domains given enough data. This systems allows the user to control the direction of the design, but can also be used in an exploratory fashion where the user just follow their favorite suggestions.

References

[1] Mathieu Aubry, Daniel Maturana, Alexei Efros, Bryan Russell, and Josef Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*, 2014.

[2] David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[3] Jeff Clune and Hod Lipson. Evolving three-dimensional objects with a generative encoding inspired by developmental biology. In *ECAL*, pages 141–148, 2011.

[4] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

[5] Amy K Hoover, Paul A Szerlip, and Kenneth O Stanley. Interactively evolving harmonies through functional scaffolding. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 387–394. ACM, 2011.

[6] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[7] Jimmy Secretan, Nicholas Beato, David B D Ambrosio, Adelein Rodriguez, Adam Campbell, and Kenneth O Stanley. Picbreeder: evolving pictures collaboratively online. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1759–1768. ACM, 2008.

[8] Hideyuki Takagi. Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296, 2001.

[9] A. Yu and K. Grauman. Fine-Grained Visual Comparisons with Local Learning. In *Computer Vision and Pattern Recognition (CVPR)*, June 2014.