Full Citation: Nguyen A, Yosinski J, Clune J (2015) *Innovation Engines: Automated Creativity and Improved Stochastic Optimization via Deep Learning.* Proceedings of the Genetic and Evolutionary Computation Conference.

Innovation Engines: Automated Creativity and Improved Stochastic Optimization via Deep Learning

Anh Nguyen
University of Wyoming
anguyen8@uwyo.edu

Jason Yosinski
Cornell University
yosinski@cs.cornell.edu

Jeff Clune
University of Wyoming
jeffclune@uwyo.edu

ABSTRACT

The Achilles Heel of stochastic optimization algorithms is getting trapped on local optima. Novelty Search avoids this problem by encouraging a search in all interesting directions. That occurs by replacing a performance objective with a reward for novel behaviors, as defined by a human-crafted, and often simple, behavioral distance function. While Novelty Search is a major conceptual breakthrough and outperforms traditional stochastic optimization on certain problems, it is not clear how to apply it to challenging, high-dimensional problems where specifying a useful behavioral distance function is difficult. For example, in the space of images, how do you encourage novelty to produce hawks and heroes instead of endless pixel static? Here we propose a new algorithm, the Innovation Engine, that builds on Novelty Search by replacing the human-crafted behavioral distance with a Deep Neural Network (DNN) that can recognize *interesting* differences between phenotypes. The key insight is that DNNs can recognize similarities and differences between phenotypes at an abstract level, wherein novelty means *interesting* novelty. For example, a novelty pressure in image space does not explore in the low-level pixel space, but instead creates a pressure to create new *types* of images (e.g. churches, mosques, obelisks, etc.). Here we describe the long-term vision for the Innovation Engine algorithm, which involves many technical challenges that remain to be solved. We then implement a simplified version of the algorithm that enables us to explore some of the algorithm's key motivations. Our initial results, in the domain of images, suggest that Innovation Engines could ultimately automate the production of endless streams of interesting solutions in any domain: e.g. producing intelligent software, robot controllers, optimized physical components, and art.

Categories and Subject Descriptors

I.2.6 [**AI**]: Learning—*Connectionism and neural nets*

Keywords

Deep Neural Networks; Deep Learning; MAP-Elites

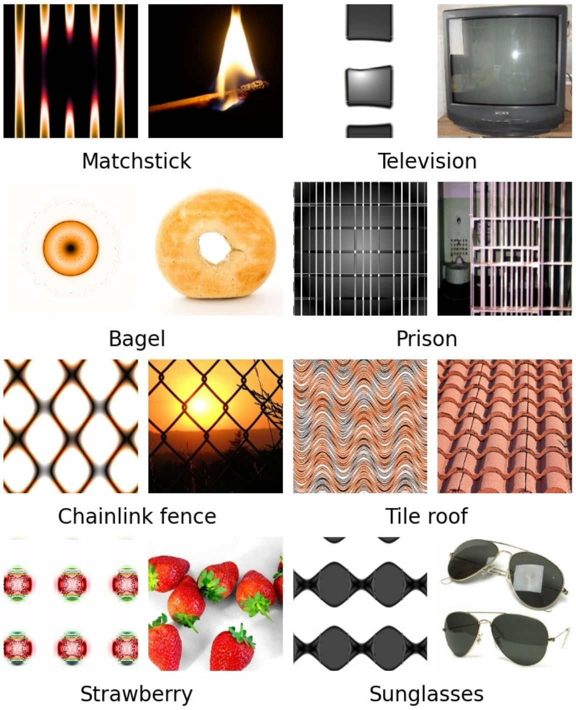

Figure 1: Images produced by an Innovation Engine that look like example target classes. In each pair, an evolved image (left) is shown with a real image (right) from the training set used to train the deep neural network that evaluates evolving images.

1. INTRODUCTION

Stochastic optimization and search algorithms, such as simulated annealing and evolutionary algorithms (EAs), often outperform human engineers in several domains [11]. However, there are other domains in which these algorithms cannot produce effective solutions yet. Their Achilles Heel is the trap of local optima [29], where the objective given to an algorithm (e.g. a fitness function) prevents the search from leaving sub-optimal solutions and reaching better ones. Novelty Search [13, 14] addresses this problem by collecting the stepping stones needed to ultimately lead to an objective instead of directly optimizing towards it. The algorithm

GECCO '15, July 11 - 15, 2015, Madrid, Spain
© 2015 ACM. ISBN 978-1-4503-3472-3/15/07...$15.00
DOI: http://dx.doi.org/10.1145/2739480.2754703

encourages searching in all directions by replacing a performance objective with a reward for novel behaviors, the novelty of which is measured with a distance function in the behavior space [16]. This recent conceptual breakthrough has been shown to outperform traditional stochastic optimization on problems where specifying distances between desired behaviors is easy [13, 14]. Reducing a high-dimensional search space to a low-dimensional one is essential to the success of Novelty Search, because in high-dimensional search spaces there are too many ways to be novel without being interesting [5]. For example, if novelty is measured directly in the high-dimensional space of pixels in a 60,000 pixel image, being different can mean different static patterns, which are not *interestingly different types* of images.

Here we propose a novel algorithm called an *Innovation Engine* that enables searching in high-dimensional spaces for which it is difficult for humans to define what constitutes *interestingly different behaviors*. The key insight is to use a deep neural network (DNN) [2] as the evaluation function to reduce a high-dimensional search space to a low-dimensional search space where novelty means *interesting* novelty. State-of-the-art DNNs have demonstrated impressive and sometimes human-competitive results on many pattern recognition tasks [12, 2]. They see past the myriad pixel differences, such as lighting changes, rotations, zooms, and occlusions, to recognize abstract concepts in images, such as tigers, tables, and turnips. Here we suggest harnessing the power of DNNs to recognize different types of things in the abstract, high-level spaces they can make distinctions in.

A second reason for choosing DNNs is that they work by hierarchically recognizing features. In images, for example, they recognize faces by combining edges into corners, then corners into eyes or noses, and then they combine these features into even higher-level features such as faces [2]. Such a hierarchy of features is beneficial because those features can be produced in different combinations to produce new types of ideas/solutions.

We first describe our long-term, ultimate vision for Innovation Engines that require no labeled data to endlessly innovate in any domain. Because there are many technical hurdles to overcome to reach that vision, we also describe a simpler, version 1.0 Innovation Engine that harnesses labeled data to simulate how the ultimate Innovation Engine might function. While Innovation Engines should work in any domain, we test one in the image generating domain that originally inspired the Novelty Search algorithm and show that it can automatically produce a diversity of interesting images (Fig. 1). We also confirm some expectations regarding why Innovation Engines are expected to work.

2. INNOVATION ENGINES

The Innovation Engine algorithm seeks to abstract the process of curiosity and habituation that occurs in humans. Historically, humans create ideas based on combinations of, or changes to, previous ideas, evaluate whether these ideas are interesting, and retain the interesting ideas to create more advanced ideas (Fig. 2). We propose to automate the entire process by having stochastic optimization (e.g. an evolutionary algorithm) generate new behaviors and a DNN evaluate whether the behaviors are interestingly new. The DNN will then be re-trained to learn all behaviors generated so far and evolution will be asked to produce new behaviors that the network has not seen before. This algorithm should be able to automatically create an endless stream of interesting solutions in any domain, e.g. producing robot controllers, optimized electrical circuits, and even art.

Creating an Innovation Engine requires generating and retaining “stepping stones to everywhere.” The stepping stones on the path to any particular innovation are not known ahead of time [14]. From the stone age, for example, the path to create a telephone did not involve inventing only things that improved long-distance communication, but instead involved accumulating all interesting innovations (Fig. 2). In fact, had human culture been restricted to only producing inventions that improve long-distance communication, it is likely that the telephone would never have been developed. That is because many of the fundamental telephone-enabling inventions were not invented because they contributed to long-distance communication (e.g. wires, electricity, electromagnets, etc) but to completely different goals. The same is true for nearly every significant invention in human history: many of the key enabling technologies were originally invented for other purposes [15]. In art, just as in science, there is a similar accumulation of interesting ideas over time and a pressure to “make something new”, which leads to a steady discovery of new artistic ideas over time [15]. Human culture, therefore, can be seen as an “Innovation Engine” that steadily produces new inventions in many different domains, from math and science to art and engineering.

2.1 The ultimate goal

Our long-term vision is to create an Innovation Engine that does not require labeled data, or perhaps is not even shown data from the natural or man-made world. It would learn to classify the types of things it has produced so far and seeks to produce new types of things. Technically, one way to implement this algorithm is by training generative deep neural network models with unsupervised learning algorithms: these generative models can learn to compress the types of data they have seen before [2, 8]. One could thus measure if a newly generated thing is a new type of thing by how well the generative DNN model can compress it. Evolution will be rewarded for producing things that the DNN cannot compress well, which should endlessly produce novel types of things. If certain types of data (e.g. static) cannot be compressed, those can be deemed uninteresting (unfit).

Imagine such an Innovation Engine in the image domain. A network trained on all images produced so far will attempt to compress each newly generated image, and it will fail more on new types of images. We hypothesize that the DNN will continuously become “bored” with (i.e. highly compress) easily produced classes of images (initially static and solid colors, but soon more complex patterns), which will encourage evolution to generate increasingly complex images in order to produce new types of images. The process thus becomes a coevolutionary *innovation arms race*.

This version of the Innovation Engine is motivated by Schmidhuber’s curiosity work [23] – which emphasizes the production of things that are not compressed yet, but are most easily compressed next – but our work involves modern compressors (state-of-the-art DNNs) and our algorithm does not require the seemingly impossible task of *predicting* which classes of artifacts are highly compressible. Our proposal is similar to [17], but prevents cycling by attempting to produce things different than everything produced so far,

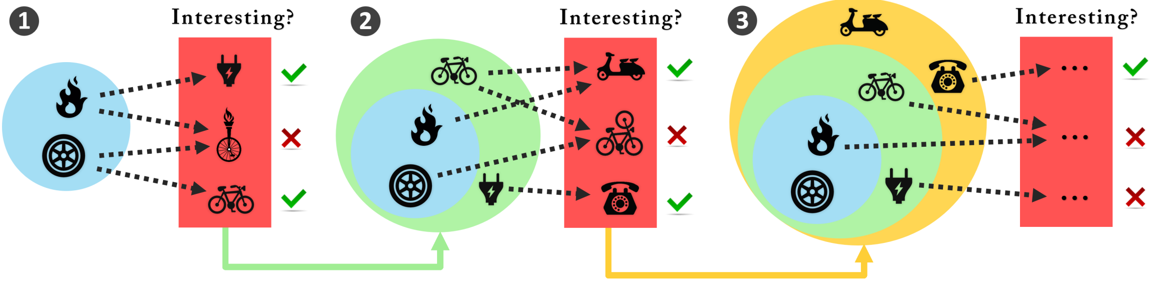

Figure 2: The Innovation Engine: Human culture creates amazing inventions, such as the telephone, by accumulating a multitude of interesting innovations in all directions. These stepping stones are collected, improved and then combined to create new innovations, which in turn, serve as the stepping stones for innovations in later generations. We propose to automate this process by having stochastic optimization (e.g. evolutionary algorithms) generate candidate solutions from the current archive of stepping stones and Deep Neural Networks evaluate whether they are *interestingly new* and should thus be archived.

not just the current population. If it works, this Innovation Engine could produce innovations in the multitude of fields and problem domains that currently benefit from stochastic optimization.

2.2 Version 1.0

Unsupervised learning algorithms for generative models do not yet scale to high dimensional data [3]; for example, they can handle 28 × 28 pixel MNIST images [8] but not 256 × 256 pixel ImageNet images [7]. In this section we describe a simpler Innovation Engine version that can be implemented with currently available algorithms. A key piece of the ultimate Innovation Engine is automatically recognizing new types of classes, which function as newly created niches for evolution to specialize on. We can emulate that endless process of niche creation by simply starting with a lot of niches and letting evolution exploit them all. To do that, we can take advantage of two recent developments in machine learning: (1) the availability of large, supervised datasets, and (2) the ability of modern supervised Deep Learning algorithms to train DNNs to reach near-human-competitive levels in classifying the things in these datasets [8, 12, 2]. We can thus challenge optimization algorithms (e.g. evolution) to produce things that the DNN recognizes as belonging to each class.

Innovation Engines require two key components: (1) a diversity-promoting EA that generates and collects novel behaviors, and (2) a DNN capable of evaluating the behaviors to determine if they are interesting and should be retained. The first criterion could be fulfilled either by Novelty Search or the multi-dimensional archive of phenotypic elites (MAP-Elites) algorithm [19, 6]. We show below that both can work.

3. TEST DOMAIN: GENERATING IMAGES

The test domain for the paper is generating a diverse set of interesting, recognizable images. We chose this domain for four reasons. The first is because an experiment in image generation served as the inspiration for Novelty Search [25]. That experiment occurred on Picbreeder.org, a website that allowed visitors to interactively evolve images [24], resulting in a crowd of humans that evolved a diverse, recognizable set of images. Key enablers of this diversity were [24, 25]: the fact that collectively there was no goal; that individuals periodically had a target image type in mind, creating a local pressure for high-performing (recognizable) images; users were open to the possibility of switching to a new goal if the opportunity presented itself (e.g. if the eyes of a face started to look like the wheels of a car); that users saved any image that they found interesting (usually a new type of image, or an improvement upon a previous type of image) and future users could branch off of any saved stepping stone to create a new image. Critically, all of these elements should also occur in Innovation Engine 1.0; thus one test of that hypothesis is whether Innovation Engine 1.0 can automatically produce a diverse set of images like those generated by humans on Picbreeder. One attempt was made to automatically recreate the diversity of recognizable images produced on Picbreeder, but it produced only abstract patterns [1].

The second motivation for the image-generating domain is that DNNs are nearly human-competitive at recognizing images [12, 10, 28]. The third reason is that DNNs can recognize and sensibly classify the type of images from Picbreeder (Fig. 3), specifically images encoded by compositional pattern producing networks (CPPNs) [27]. We also encode images with CPPNs in our experiments (described below). The fourth reason is because humans are natural pattern recognizers, making us quickly and intuitively able to evaluate the diversity, interestingness, and recognizability of evolved solutions. Additionally, while much of what we learn from this domain comes from subjective results, there is also a quantitative aspect regarding the confidence a DNN ascribes to the generated images. In future work we will test whether the conclusions reached in this mostly subjective domain translate into more exclusively quantitative domains.

To experiment in this domain, we use a modern off-the-shelf DNN trained with 1.3 million images to recognize 1000 different types of objects from the natural world. We then challenge evolution to produce images that the DNN confidently labels as members of each of the 1000 classes. Evolution is therefore challenged to make increasingly recognizable images for all 1000 classes. Generating CPPN-encoded images that are recognizable is challenging [29], making recognizability a notion of performance in this domain. Being recognizable is also related to being interesting, as Picbreeder images that are recognizable are often the most highly rated [24].

4. METHODS

4.1 Deep neural network models

The DNN in our experiments is the well-known convolutional "AlexNet" architecture from [12]. It is trained on the 1.3-million-image 2012 ImageNet dataset [7, 22], and available for download via the Caffe software package [9]. The Caffe-provided AlexNet has small architectural differences from Krizhevsky 2012 [12], but it performs similarly (42.6% top-1 error rate vs. the original 40.7% [12]). For each image, the DNN outputs a post-softmax, 1000-dimensional vector reporting the probability that the image belongs to each ImageNet class. The softmax means that to produce a high confidence value for one class, all the others must be low.

4.2 Generating images with evolution

To simultaneously evolve images that match all 1000 ImageNet classes, we use the new multi-dimensional archive of phenotypic elites (MAP-Elites) algorithm [19, 6]. MAP-Elites keeps a map (archive) of the best individuals found so far for each class. Each iteration, an individual is randomly chosen from the map, mutated, and then it replaces the current champion for any class if it has a higher fitness for that class. Fitness is the DNN's confidence that an image is a member of that class.

We also test another implementation of the Innovation Engine, but with Novelty Search instead of MAP-Elites. Novelty Search encourages organisms to be different from the current population and an archive of previously novel individuals. The behavioral distance between two images is defined as the Euclidean distance between the two 1000-dimensional vectors output by the DNN for each image. Because all of our experiments were performed with the Sferes evolutionary computation framework [20], we set all Novelty Search parameters to those in [18], which was also conducted in Sferes, but followed closely the parameters in [13].

Images are encoded with compositional pattern producing networks (CPPNs) [27], which abstract the expressive power of developmental biology to produce regular patterns (e.g. those with symmetry or repetition). CPPNs encode the complex, regular, recognizable images on Picbreeder.org (e.g. Fig. 3) and the 3D objects on EndlessForms.com [4]. The details of how CPPNs encode images and are evolved have been repeatedly described elsewhere [24, 27]. Briefly, a CPPN is like a neural network, but each node's activation function is one of a set (here: sine, sigmoid, Gaussian and linear). The Cartesian coordinates of each pixel are input into the network and the network's outputs determine the color of that pixel. Importantly, evolved CPPN images can be recognized by the DNN (Fig. 3), showing that evolution can produce CPPN images that both humans and DNNs can recognize.

As is customary [24, 27, 4] we evolve CPPNs with the principles of the NeuroEvolution of Augmenting Topologies (NEAT) algorithm [26], a version of which is provided in Sferes. CPPNs start with no hidden nodes, and add nodes and connections over time, forcing evolution to first search for simple, regular images before increasing complexity [26]. All of our code and parameters are available at http://EvolvingAI.org. Because each run required 128 CPU cores running continuously for ~4 days, our run number is limited.

5. RESULTS

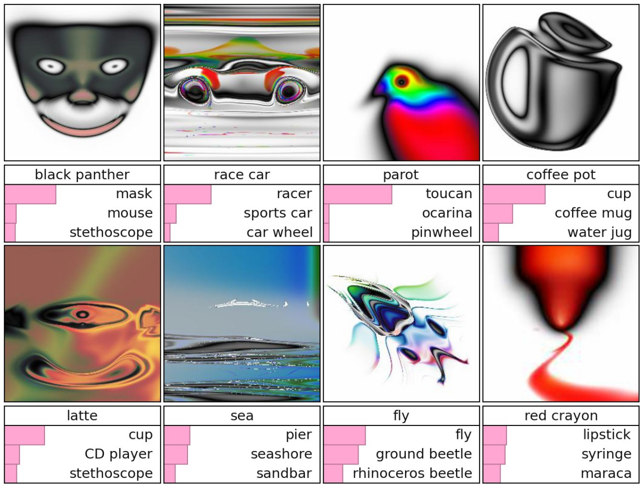

Figure 3: CPPN-encoded images evolved and named (centered text) by Picbreeder.org users. The DNN's top three classifications and associated confidence (size of the pink bar) are shown. The DNN's classifications often relate to the human breeder's label, showing that DNNs can recognize CPPN-encoded, evolved images. Adapted from [21].

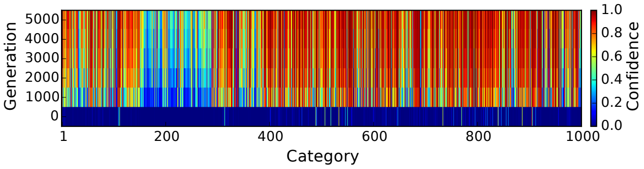

Figure 4: The MAP-Elites evolutionary algorithm produces images that the DNN declares with high confidence to belong to most ImageNet classes. Colors represent median confidence scores from 10 runs.

5.1 Evolving images to match ImageNet

If the Innovation Engine is a promising idea, then Innovation Engine 1.0 in the image domain should produce the following: (1) images that the DNN is confident are class members and (2) a diverse set of interesting images that are (3) recognizable as members of the target class.

In 10 independent MAP-Elites runs, evolution produced high-confidence images in most categories (Fig. 4). It struggles most in classes 156-286, which represent subtly different species of dogs and cats, where it is hard to look like one type without also looking like other types. While the reader must draw their own conclusions, in our opinion the images exhibit a tremendous amount of interesting diversity. Selected examples are in Figs. 5, 1, 7: all 10,000 evolved images are shown at http://EvolvingAI.org. The diversity is especially noteworthy because many images are phylogenetically related, which should curtail diversity.

In many cases the evolved images are recognizable as members of the target class (Fig. 7). This result is remarkable given that it has been shown with the same encoding (CPPN) and evolutionary algorithm (NEAT), that it is impossible to evolve an image to resemble a complex, target image [29]. The lesson from that paper is that if evolution is given a specific objective, such as to evolve a but-

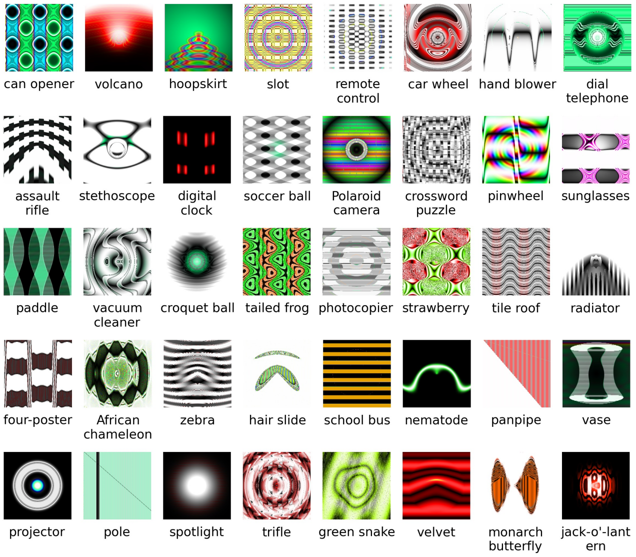

Figure 5: Innovation Engines in the image domain generate a tremendous diversity of interesting images. Shown are images selected to showcase diversity from 10 evolutionary runs. The diversity results from the pressure to match 1000 different ImageNet classes. In this and subsequent figures, the DNN's top label for each evolved image is shown below it.

terfly or skull, that it will not succeed because objective-driven evolution only rewards images that increasingly look like butterflies or skulls, and that CPPN lineages that lead to butterflies or skulls tend to pass through images that look nothing like either. Innovation Engines, like crowds on Picbreeder, simultaneously collect improvements in a large number of objectives. That allows evolutionary lineages to be rewarded for steps that do not resemble butterflies or skulls (provided they resemble something else) and then to be rewarded as butterflies or skulls if they resemble either. Thus, a main result of this paper is that the problem is not being objective-driven, but instead being driven by only a few objectives. The key is to collect "stepping stones in all interesting directions", which can be approximated by simultaneously selecting for a vast number of objectives. Supporting this argument, our algorithm was able to produce many complex structures (Figs. 5, 1, 7) , including some that are similar to butterflies and skulls (Fig. 6). Confirming that these images can be considered art, they were accepted to a selective art competition (35% acceptance rate) and displayed at the University of Wyoming Art Museum.

Some evolved images are not recognizable, but often do contain recognizable features of the target class. For example, in Fig. 5, the *remote control* has a grid of buttons and the *zebra* has black-and-white stripes.

As was recently reported in [21], this algorithm also produces many images that DNNs assign high confidence scores to, but that are totally unrecognizable, even when knowing their class labels (e.g. Fig. 5, *tailed frog* & *soccer ball*). That study emphasized that the existence of such "fooling images" is problematic for anything that relies on DNNs to accurately classify objects, because DNNs sometimes make mistakes. This paper emphasizes the opposite, but not mutually exclusive, perspective: while using DNN as evaluators sometime produces fooling examples, it also sometimes works really well, and can thus automatically drive the evolution of a diverse set of complex, interesting, and sometimes recognizable images. In future work we investigate increasing the percent of evolved images that are recognizable.

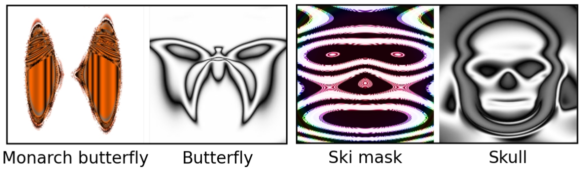

Figure 6: The Innovation Engine 1.0 evolved images that resemble those originally evolved on Picbreeder, but that a previous paper [29] showed were impossible to re-evolve with single-objective, target evolution. ImageNet has a "Monarch butterfly" class; it does not have a "skull" class, but its "Ski mask" class contains the key eyes, nose and mouth features. Shown are images evolved with Innovation Engine 1.0 (*left*) and Picbreeder (*right*).

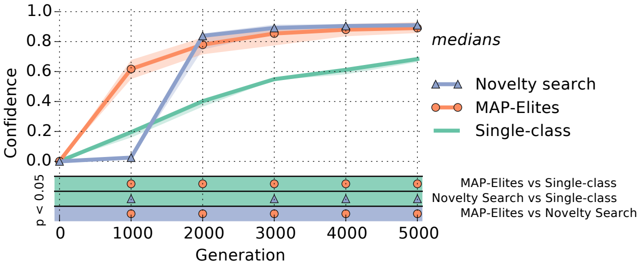

Figure 8: Innovation Engines built with MAP-Elites or Novelty Search perform similarly to each other, and both significantly outperform a single-class evolutionary algorithm. Solid lines show median performance and shaded areas indicate the 95% bootstrapped confidence interval of the median. The bottom three rows show statistical significance.

5.2 Evolving towards multiple objectives

As discussed in the previous section, a key hypothesis underpinning Innovation Engines is that evolving toward a vast number of objectives simultaneously is more effective than evolving toward each objective separately. In this section, we probe that hypothesis directly by comparing how MAP-Elites performs on all objectives vs. how evolution fares when evolving to each single-class objective separately. Because we did not have the computational resources to perform 1000 single-class runs, we randomly selected 100 categories and performed two single-class MAP-Elites runs per category. We compare that data to how the 10 runs of 1000-class MAP-Elites performed on the same 100-class subset.

1000-class MAP-Elites produced images with significantly higher median DNN confidence scores (Fig. 8, 90.3% vs. 68.3%, $p < 0.05$ via Mann-Whitney U test). The theory behind why more objectives helps is because a lineage that is currently the champion in class X may be trapped on a local optima, such that mutations to it will not improve its fitness on that objective (a phenomenon we observe in the

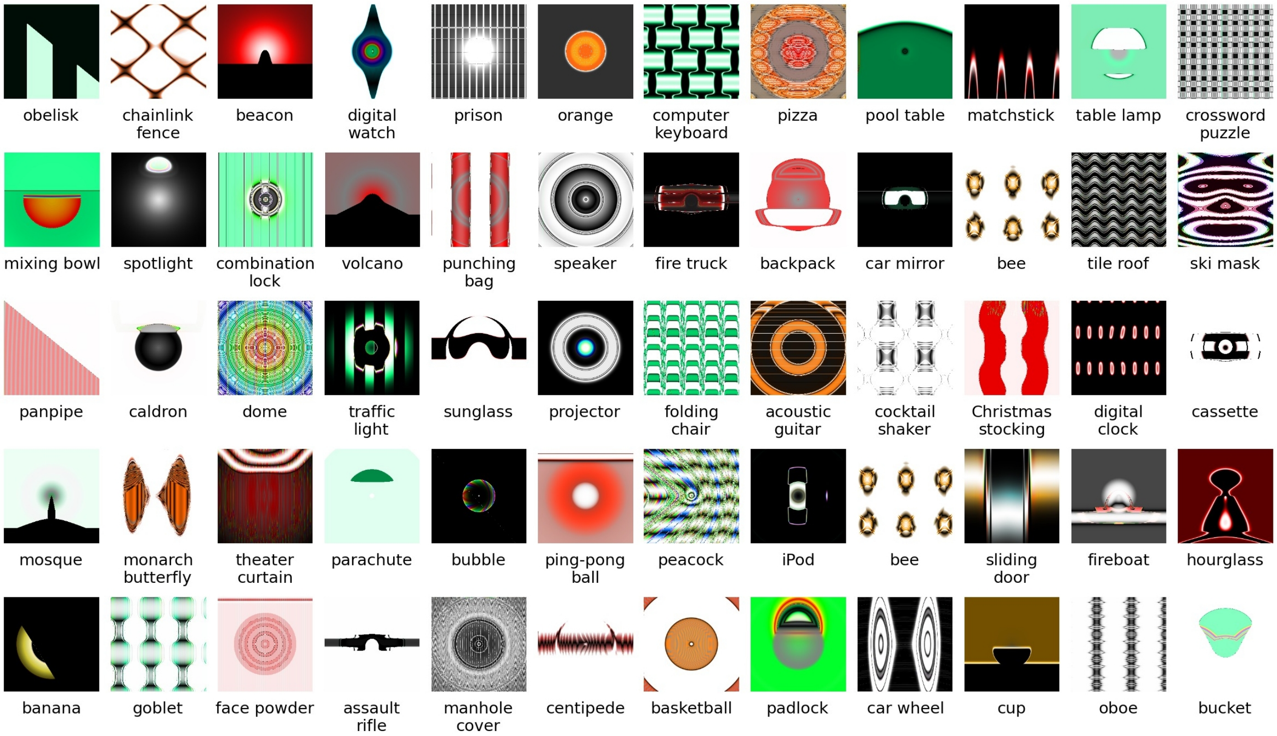

Figure 7: Innovation Engines are capable of producing images that are not only given high confidence scores by a deep neural network, but are also qualitatively interesting and recognizable. To show the most interesting images we observed evolve, we selected images from both the 10 main experiment runs and 10 preliminary experiments with slightly different parameters.

single-class case: Fig. 9 inset). With many objectives, however, a lineage that has been selected for other objectives can mutate to perform better on class X, which occurs frequently with MAP-Elites. For example, on the *water tower* class (Fig. 9 inset), the lineage of images that produce a large, top-lit sphere do not improve for 250 generations, but at generation 1750 a descendant of an organism that was the champion for the *cocker spaniel dog* class (Fig. 9) became a recognizable water tower and was then further refined.

Inspecting the phylogenetic tree of the 1000 images produced by MAP-Elites in one run, we found that the evolutionary path to a final image often went through other classes, a phenomenon we call *goal switching*. For example, the path to a beacon involved stepping stones that were rewarded because they were at one point champions for the *tench*, *abaya*, *megalith*, *clock*, and *cocker spaniel dog* classes (Fig. 9). A different descendent of *abaya* traversed the *stingray* and *boathouse* classes en route to a recognizable *planetarium* (Fig. 9). A related phenomenon occurs on Picbreeder, where the evolutionary path to a target image often involves images that do not resemble the target [24].

We quantitatively measured the number of goal switches per class (the number of times during a run that a new class champion was the offspring of a champion of another class). The code to track this statistic was only available for the last 3 MAP-Elites runs. Each class had a *mean* of 12.5 goal switches, which was 21.7% of the 57.6 mean new champions per class. Thus, a large percentage of improvements in a class came not from refining the current class champion, but from a mutation to a different class champion, helping to explain why Innovation Engines work.

Another expectation, which we indeed observed, is that the evolved images for many semantically related categories are also phylogenetically related. For example, according to WordNet hierarchy [7], *planetarium*, *mosque*, *church*, *obelisk*, *yurt* and *beacon* are subclasses of the *structure* class. The evolved images for these classes often are closely related phylogenetically, which also entails visual similarity (Fig. 9).

If two CPPN genomes produce equivalent behaviors (here, images), it is taken as a sign of increased evolvability if one has fewer nodes and connections [29]. It has been shown that objectives "corrupt" genomes by adding piecewise hacks that lead to small fitness gains, and thus do not find the simple, elegant solutions produced by divergent searches (e.g. novelty search or Picbreeder crowds). If Innovation Engines behave like traditional single- or multi-objective algorithms, one might expect them to produce large CPPN genomes. On the other hand, if Innovation Engines, which are *many*-objective algorithms, are more divergent in nature, they should produce smaller genomes like those reported for Picbreeder [29]. While the comparison is not apples to apples for many reasons, Innovation Engine genomes are actually more compact than those for Picbreeder. The 10,000 MAP-Elites CPPN genomes contain a median of 27 nodes (SD = 5.9) and 37.5 connections (SD= 8.6) vs. the ~7,500 Picbreeder image genomes analyzed in [24], which have 50.3 nodes and 146.7 connections (SD not reported).

5.3 Innovation Engine with Novelty Search

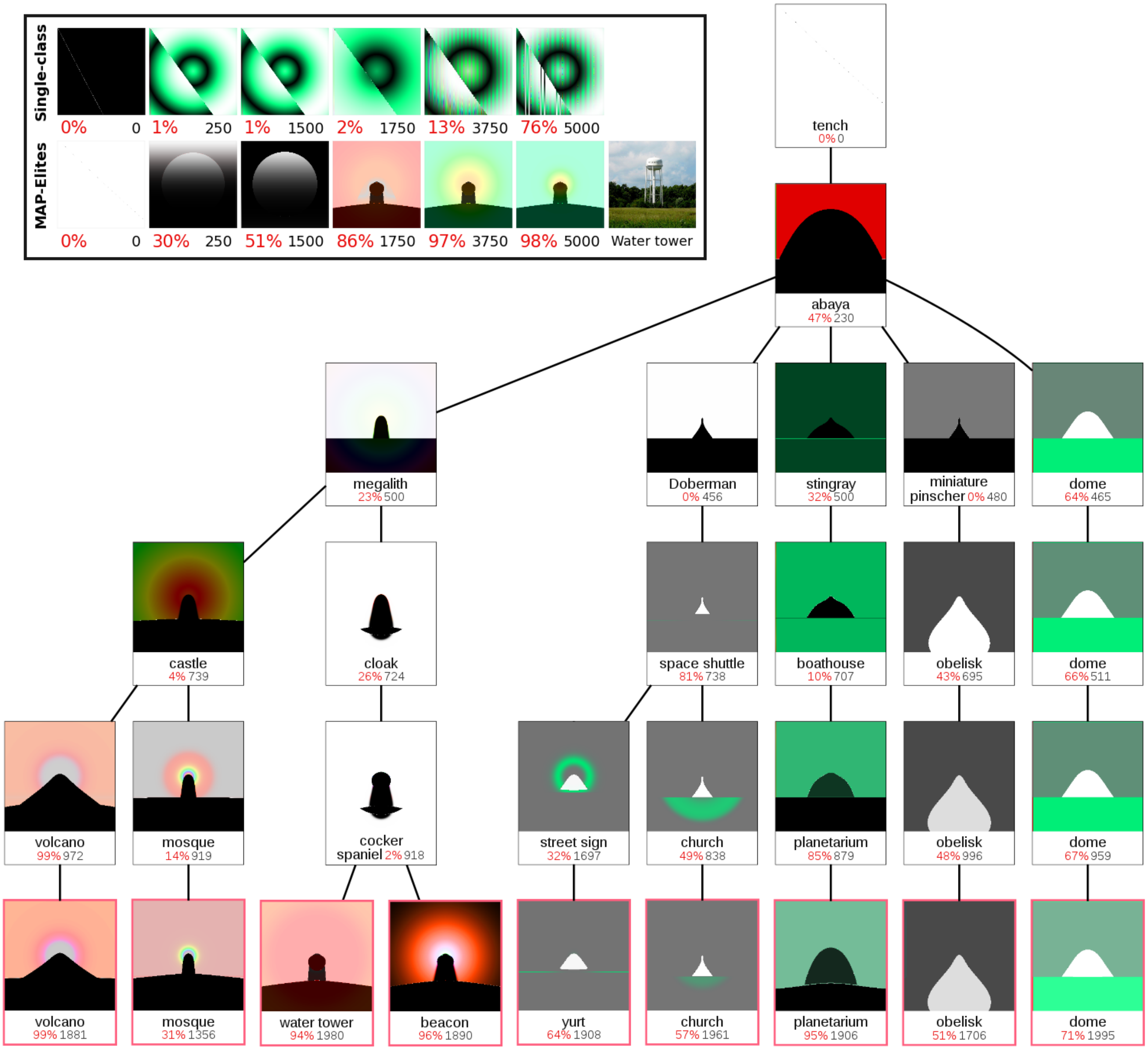

Figure 9: *Inset Panel:* The champions for the water tower class over evolutionary time for a single-class evolutionary algorithm (*top*) and the MAP-Elites variant of the Innovation Engine (*bottom*). Under each evolved image is the percent confidence the DNN has that the image is a water tower (left) and the generation in which the image was created (right). The single-class EA gets trapped on a local optima and refines the same basic idea, resulting in an unrecognizable image with mediocre performance. However, because MAP-Elites simultaneously evolves towards many objectives, often a lineage that is a champion of one class will produce an offspring that becomes a champion of a different class, a phenomenon we call *goal switching*. That occurs here at the 1750th generation, when the offspring of a champion of the cocker spaniel (dog) class (see main panel in this figure) becomes the best water tower produced so far. Its descendants are refined to produce a high-confidence, recognizable image. A water tower image from the training set is provided for reference. *Main Figure:* A phylogenetic tree depicting how lineages evolve and *goal switch* from one class to another in an Innovation Engine (here, version 1.0 with MAP-Elites). Each image is displayed with the class the DNN placed it in, the associated DNN confidence score (*red*), and the generation in which it was created. Connections indicate ancestor-child relationships. One reason Innovation Engines work is because similar types of things (e.g. various building structures) can be produced by phylogenetically related genomes, meaning that the solution to one problem can be repurposed for a similar type of problem. Note the visual similarity between the related solutions. Another reason they work is because the path to a solution often involves a series of things that do not increasingly resemble the final solution (at least, not without the benefit of hindsight). For example, note the many unrelated classes that served as stepping stones to recognizable objects (e.g. the path through cloaks and cocker spaniels to arrive at a beacon).

To support the case that Innovation Engines should work with any diversity-promoting EA combined with a DNN-provided *deep distance function*, we implemented Innovation Engine 1.0 with Novelty Search instead of MAP-Elites. After Novelty Search was afforded the same number of image evaluations, we found the best image it produced for each class according to the DNN. We performed 9 independent runs of Novelty Search. To facilitate comparison to the single-class control, we compare performance on the 100 classes randomly selected for the single-class control (Sec. 5.2). The MAP-Elites vs. Novelty Search comparison on 100 classes is qualitatively the same on all 1000 classes.

As expected, Novelty Search also produced high-confidence images in most classes (Fig. 8). Its median confidence of 91.2% significantly outperforms the 68.3% for the single-class control ($p < 0.05$ via Mann-Whitney U test). While it significantly underperforms MAP-Elites at the 1000th generation, for the 2000th generation and beyond Novelty Search slightly, but significantly outperforms MAP-Elites ($p < 0.05$ via Mann-Whitney U test), although MAP-Elites has a higher final mean (79.5% vs. 74.0%). The images produced by two treatments are qualitatively similar (data not shown). This result confirms that on this domain either MAP-Elites or Novelty Search can serve as the diversity promoting EA.

6. DISCUSSION AND CONCLUSION

This paper provides a first, rough sketch of the Innovation Engine idea. Much work remains to investigate the simple version of it we presented and push toward more ambitious versions. In the short term, we will investigate improving the frequency of recognizable images produced. We will also create Innovation Engines in more quantitative domains. For example, we will pair DNNs trained to recognize different actions in videos (e.g. cartwheels, backflips, handshakes) with evolutionary algorithms to attempt to automatically create robot controllers for thousands of different behaviors.

Our preliminary results have shown that the Innovation Engine concept is worth exploring further. Specifically, we have supported some of its key assumptions: that evolving toward many objectives simultaneously approximates divergent search; that DNNs can provide informative, abstract distance functions in high-dimensional spaces; and that Innovation Engines can generate a large, diverse, interesting set of solutions in a given domain (here images). Innovation Engines will only get better as DNNs are improved, especially when generative DNN models can scale to higher dimensions. Ultimately, Innovation Engines could potentially be applied to the countless number of domains where stochastic optimization is applied. Like human culture, they could eventually enable endless innovation in any domain, from software and science to arithmetic proofs and art.

Acknowledgments: We thank Joost Huizinga, Chris Stanton, Tyler Hughes, Jessica Erickson, & Jean-Baptiste Mouret.

7. REFERENCES

[1] J. E. Auerbach. Automated evolution of interesting images. In *Artificial Life 13*. MIT Press, 2012.

[2] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2009.

[3] Y. Bengio, É. Thibodeau-Laufer, G. Alain, and J. Yosinski. Deep generative stochastic networks trainable by backprop. In *Proc. of the ICML*, 2014.

[4] J. Clune and H. Lipson. Evolving three-dimensional objects with a generative encoding inspired by developmental biology. In *Proc. of the European Conference on Artificial Life*, pages 144–148, 2011.

[5] G. Cuccu and F. Gomez. When novelty is not enough. In *Applications of Evolutionary Computation*. 2011.

[6] A. Cully, J. Clune, and J.-B. Mouret. Robots that can adapt like natural animals. *arXiv preprint arXiv:1407.3501*, 2014.

[7] J. Deng et al. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[8] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006.

[9] Y. Jia et al. Caffe: Convolutional architecture for fast feature embedding. In *Proc. of the International Conference on Multimedia*, pages 675–678, 2014.

[10] A. Karpathy. What I learned from competing against a convnet on ImageNet. http://goo.gl/iqCbC0, 2014.

[11] M. Keane et al. *Genetic programming IV: Routine human-competitive machine intelligence*. 2006.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[13] J. Lehman and K. O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336, 2008.

[14] J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011.

[15] J. Lehman and K. O. Stanley. Novelty search and the problem with objectives. In *Genetic Programming Theory and Practice IX*, pages 37–56. Springer, 2011.

[16] J. Li, J. Storie, and J. Clune. Encouraging creative thinking in robots improves their ability to solve challenging problems. *Algorithms*, 13:14.

[17] A. Liapis, H. P. Martınez, J. Togelius, and G. N. Yannakakis. Transforming exploratory creativity with delenox. In *Proc. of the Fourth International Conference on Computational Creativity*, 2013.

[18] J.-B. Mouret. Novelty-based multiobjectivization. In *New Horizons in Evolutionary Robotics*. Springer, 2011.

[19] J.-B. Mouret and J. Clune. Illuminating search spaces by mapping elites. *arXiv preprint*, 2015.

[20] J.-B. Mouret and S. Doncieux. Sferes v2: Evolvin'in the multi-core world. In *Congress on Evolutionary Computation*, pages 1–8, 2010.

[21] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proc. of the Conference on Computer Vision and Pattern Recognition*, 2015.

[22] O. Russakovsky et al. Imagenet large scale visual recognition challenge. *arXiv:1409.0575*, 2014.

[23] J. Schmidhuber. Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science*, 18(2):173–187, 2006.

[24] J. Secretan et al. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19(3):373–403, 2011.

[25] K. Stanley and J. Lehman. *Why Greatness Cannot Be Planned: The Myth of the Objective*. Springer, 2015.

[26] K. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.

[27] K. O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 2007.

[28] C. Szegedy et al. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[29] B. G. Woolley and K. O. Stanley. On the deleterious effects

of a priori objectives on evolution and representation. In *Proc. Genetic & Evolutionary Computation Conf.*, 2011.