

The latest news from Research at Google

---

## Inceptionism: Going Deeper into Neural Networks

Wednesday, June 17, 2015

Posted by Alexander Mordvintsev, Software Engineer, Christopher Olah, Software Engineering Intern and Mike Tyka, Software Engineer

Update - 13/07/2015

*Images in this blog post are licensed by Google Inc. under a [Creative Commons Attribution 4.0 International License](#).*

*However, images based on places by [MIT Computer Science and AI Laboratory](#) require additional permissions from MIT for use.*

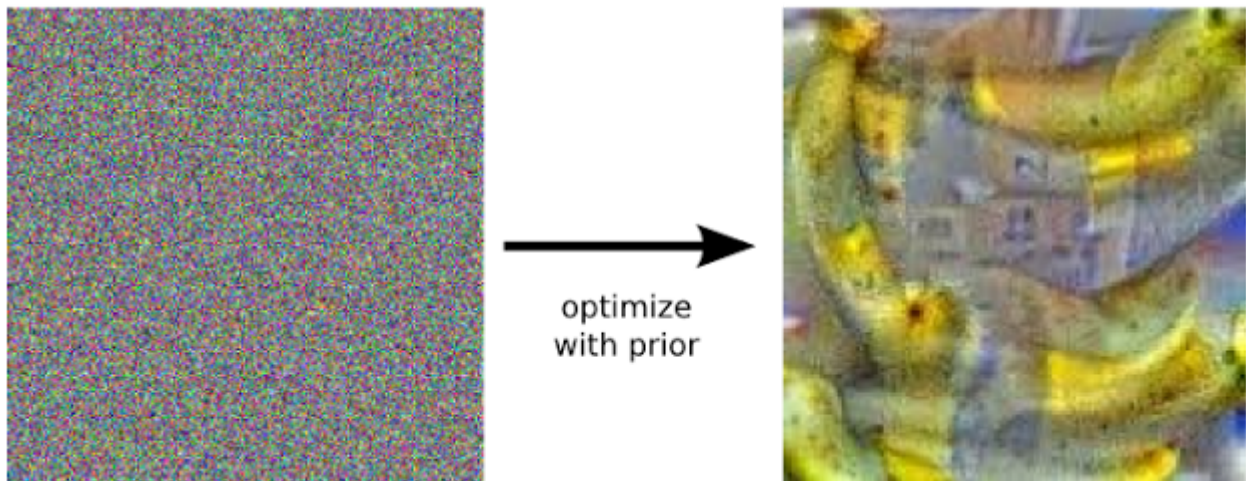
[Artificial Neural Networks](#) have spurred remarkable recent progress in [image classification](#) and [speech recognition](#). But even though these are very useful tools based on well-known mathematical methods, we actually understand surprisingly little of why certain models work and others don't. So let's take a look at some simple techniques for peeking inside these networks.

We train an artificial neural network by showing it millions of training examples and [gradually adjusting the network parameters](#) until it gives the classifications we want. The network typically consists of 10-30 stacked layers of artificial neurons. Each image is fed into the input layer, which then talks to the next layer, until eventually the "output" layer is reached. The network's "answer" comes from this final output layer.

One of the challenges of neural networks is understanding what exactly goes on at each layer. We know that after training, each layer progressively extracts higher and higher-level features of the image, until the final layer essentially makes a decision on what the image shows. For example, the first layer maybe looks for edges or corners. Intermediate layers interpret the basic features to look for overall shapes or components, like a door or a leaf. The final few layers assemble those into complete interpretations—these neurons activate in response to very complex things such as entire buildings or trees.

One way to visualize what goes on is to turn the network upside down and ask it to enhance an input image in such a way as to elicit a particular interpretation. Say you want to know what sort of image would result in "Banana." Start with an image full of random noise, then gradually tweak the image towards what the neural net considers a banana (see related work in [\[1\]](#), [\[2\]](#), [\[3\]](#),

[4]). By itself, that doesn't work very well, but it does if we impose a prior constraint that the image should have similar statistics to natural images, such as neighboring pixels needing to be correlated.



So here's one surprise: neural networks that were trained to discriminate between different kinds of images have quite a bit of the information needed to generate

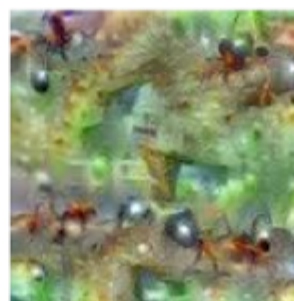
images too. Check out some more examples across different classes:



Hartebeest



Measuring Cup



Ant



Starfish



Anemone Fish



Banana



Parachute



Screw

Why is this important? Well, we train networks by simply showing them many examples of what we want them to learn, hoping they extract the essence of the matter at hand (e.g., a fork needs a handle and 2-4 tines), and learn to ignore what doesn't matter (a fork can be any shape, size, color or orientation). But how do you check that the network has correctly learned the right features? It can help to visualize the network's representation of a fork.

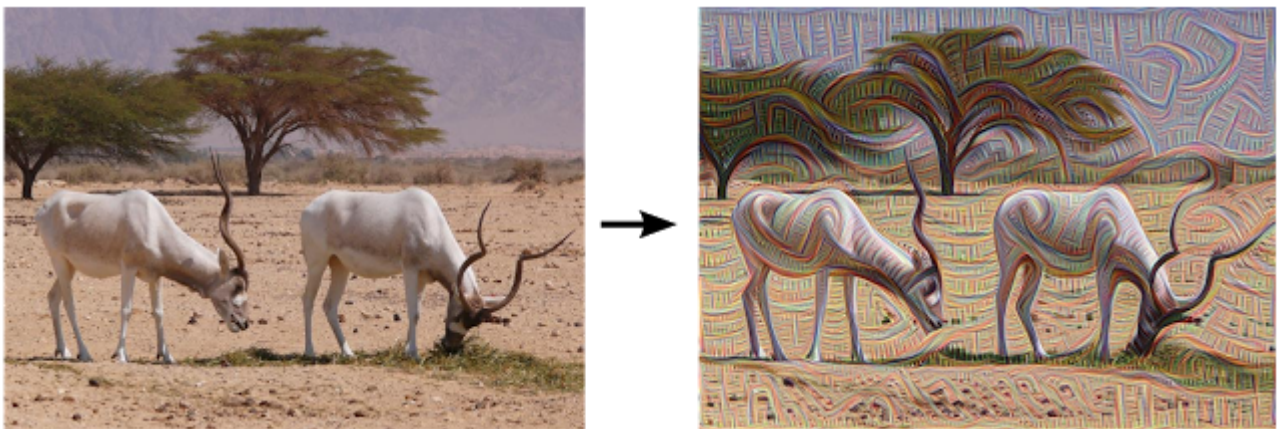
Indeed, in some cases, this reveals that the neural net isn't quite looking for the thing we thought it was. For example, here's what one neural net we designed thought dumbbells looked like:





There are dumbbells in there alright, but it seems no picture of a dumbbell is complete without a muscular weightlifter there to lift them. In this case, the network failed to completely distill the essence of a dumbbell. Maybe it's never been shown a dumbbell without an arm holding it. Visualization can help us correct these kinds of training mishaps.

Instead of exactly prescribing which feature we want the network to amplify, we can also let the network make that decision. In this case we simply feed the network an arbitrary image or photo and let the network analyze the picture. We then pick a layer and ask the network to enhance whatever it detected. Each layer of the network deals with features at a different level of abstraction, so the complexity of features we generate depends on which layer we choose to enhance. For example, lower layers tend to produce strokes or simple ornament-like patterns, because those layers are sensitive to basic features such as edges and their orientations.

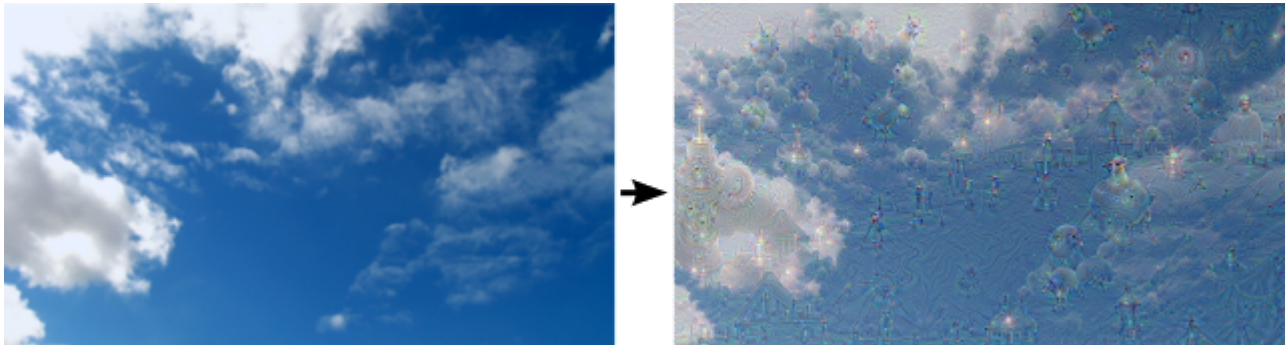


Left: Original photo by [Zachi Evenor](#). Right: processed by Günther Noack, Software Engineer



Left: Original painting by [Georges Seurat](#). Right: processed images by Matthew McNaughton, Software Engineer

If we choose higher-level layers, which identify more sophisticated features in images, complex features or even whole objects tend to emerge. Again, we just start with an existing image and give it to our neural net. We ask the network: "Whatever you see there, I want more of it!" This creates a feedback loop: if a cloud looks a little bit like a bird, the network will make it look more like a bird. This in turn will make the network recognize the bird even more strongly on the next pass and so forth, until a highly detailed bird appears, seemingly out of nowhere.



The results are intriguing—even a relatively simple neural network can be used to over-interpret an image, just like as children we enjoyed watching clouds and interpreting the random shapes. This network was trained mostly on images of animals, so naturally it tends to interpret shapes as animals. But because the data is stored at such a high abstraction, the results are an interesting remix of these learned features.



"Admiral Dog!"

"The Pig-Snail"

"The Camel-Bird"

"The Dog-Fish"

Of course, we can do more than cloud watching with this technique. We can apply it to any kind of image. The results vary quite a bit with the kind of image, because the features that are entered bias the network towards certain interpretations. For example, horizon lines tend to get filled with towers and pagodas. Rocks and trees turn into buildings. Birds and insects appear in images of leaves.

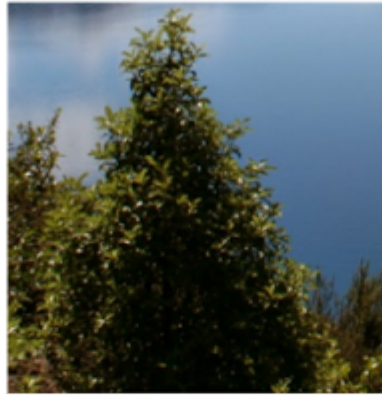




Horizon



Towers & Pagodas



Trees



Buildings



Leaves



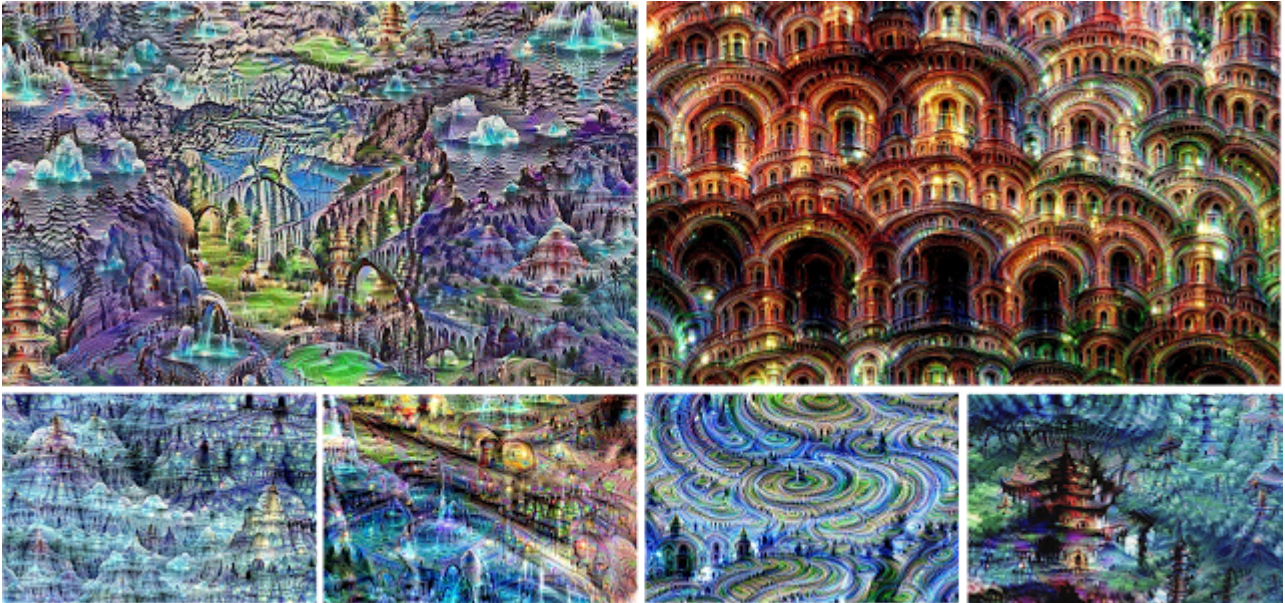
Birds & Insects

*The original image influences what kind of objects form in the processed image.*

This technique gives us a qualitative sense of the level of abstraction that a particular layer has achieved in its understanding of images. We call this technique “Inceptionism” in reference to the [neural net architecture](#) used. See our [Inceptionism gallery](#) for more pairs of images and their processed results, plus some cool video animations.

### **We must go deeper: Iterations**

If we apply the algorithm iteratively on its own outputs and apply some zooming after each iteration, we get an endless stream of new impressions, exploring the set of things the network knows about. We can even start this process from a random-noise image, so that the result becomes purely the result of the neural network, as seen in the following images:



Neural net “dreams”— generated purely from random noise, using a network trained on places by [MIT Computer Science and AI Laboratory](#). See our [Inceptionism gallery](#) for hi-res versions of the images above and more (Images marked “Places205-GoogLeNet” were made using this network).

The techniques presented here help us understand and visualize how neural networks are able to carry out difficult classification tasks, improve network architecture, and check what the network has learned during training. It also makes us wonder whether neural networks could become a tool for artists—a new way to remix visual concepts—or perhaps even shed a little light on the roots of the creative process in general.



31 comments

Google+



Add a comment as 严宇宇

Top comments ▾



**Jefferson Allan** 1 month ago - Shared publicly

Quite incredible. More images like the impala please! I think they're beautiful. They would make great screensavers if they were full res.

+2 +1



**Multistream** 5 months ago - Shared publicly



4 months ago - Shared publicly

On reddit you said, that it is possible to work with music (sound). But how?

+4  · Reply



**Shannon Ramos** shared this via Google+ 4 months ago - Shared publicly

 · Reply



**Siddha Ganju** shared this via Google+ 5 months ago - Shared publicly

+1  · Reply



**John Horton** 6 months ago - Shared publicly

Can I use this with my own videos, and if so how would I do that?

 · Reply



**Alessandro Braga** 5 months ago

<https://github.com/graphific/DeepDreamVideo>



**John Horton** 5 months ago

thank you



**Nacia Luo** 1 day ago - Shared publicly

It's quite uncomfortable to look at some of these pictures.

 · Reply



**Shauna Eggers** 5 days ago - Shared publicly

They do look wonderful! Esp the "buildings" dreams.

 · Reply



**Manikanta Maddula** 1 week ago - Shared publicly

Incredible inference. Truly amazing

 · Reply



**Kim Colbourne** 1 month ago - Shared publicly

Reminds me of Australian First Nation artwork. [http://art-educ4kids.weebly.com/uploads/8/9/6/9/8969100/8364444\\_orig.jpg?0](http://art-educ4kids.weebly.com/uploads/8/9/6/9/8969100/8364444_orig.jpg?0)

 · Reply



**Mobi Hunterz** 6 months ago - Shared publicly

Love to integrate this speech recognition.

+2  · Reply



**sandip patil** shared this via Google+ 1 month ago - Shared publicly

+1 · Reply



**Quinton Miszczenko** 2 months ago - Shared publicly

Shared on my facebook page very interesting

+1



**Richard Goh** 3 months ago (edited) - Shared publicly

Thanks. Very interesting. Worked on neural nets for many years before.

How do we know this(below)? Is it true for general network architecture or there needs to be some constrains for this to be true. For as I understand it, neural network backpropagation training is a mere mathematical curve fitting, there is no reason to presume it has "levels" of abstraction. Pure mathematical. Thanks

+1 +1 · Reply



**Kyle Atkins** shared this via Google+ 3 months ago - Shared publicly

+1 · Reply



**Loreto Parisi** 3 months ago - Shared publicly

terrific!

+1 · Reply



**Reinhard Lang** 4 months ago - Shared publicly

Hi! Awesome work! Are the hi-res versions in the Inceptionism gallery free to use under the licence CC4.0?

+1 +1 · Reply



**Abhijeet Bhorkar** shared this via Google+ 4 months ago - Shared publicly

+1 · Reply



**Jim Scortch** 7 months ago - Shared publicly

Seeking to apply this to voice controls and motion capture.

+1 +1



**Bo Zhao** 6 months ago - Shared publicly

Interesting

+1 · Reply





**Thomas Lefort** shared this via Google+ · 4 months ago · Shared publicly



· Reply

Show more

Labels: [Computer Vision](#) , [Google Brain](#) , [Image Classification](#) , [Neural Networks](#)



Google

Google · Privacy · Terms