Deep Dream Image Generation

Liu Yiming

University of California, Irvine

June 2017

Abstract

Deep Dream is an interesting concept which originated from the active research into the architecture and functionality of convolutional neural networks. In this paper, we will discuss the methods used in generating deep dream images, and explore the approaches, which can be employed to control and regularize the dreaming process, and finally we will apply this technique to generate some stylish images and discuss its potential future development.

1 Introduction

Deep dream is a process first discussed in a Google research blog post[1]. In this paper, we will discuss it in the context of a specific convolutional neural network architecture called GoogLeNet, which is also the network discussed in the original blog post.

The critical building block, which characterizes GoogLeNet is a particularly designed convolutional layer called 'Inception Module', and it is constructed as shown in Figure 1. There are total of 9 inception modules within the GoogLeNet. Moreover, each convolution layer within the inception module will output a bunch of feature maps, and the final output of the inception module will be an concatenation of all those feature maps along the depth direction, and RELU activation function is used in all convolutional layers. As far as this paper is concerned, we will name these inception modules as inception-3a, inception-3b, inception-4a, inception-4b, inception-4c, inception-4d, inception-4e, inception-5a and inception-5b, in the order of network input to output. A complete and thorough descriptions of the GoogLeNet architecture can be found in the paper referenced[2].

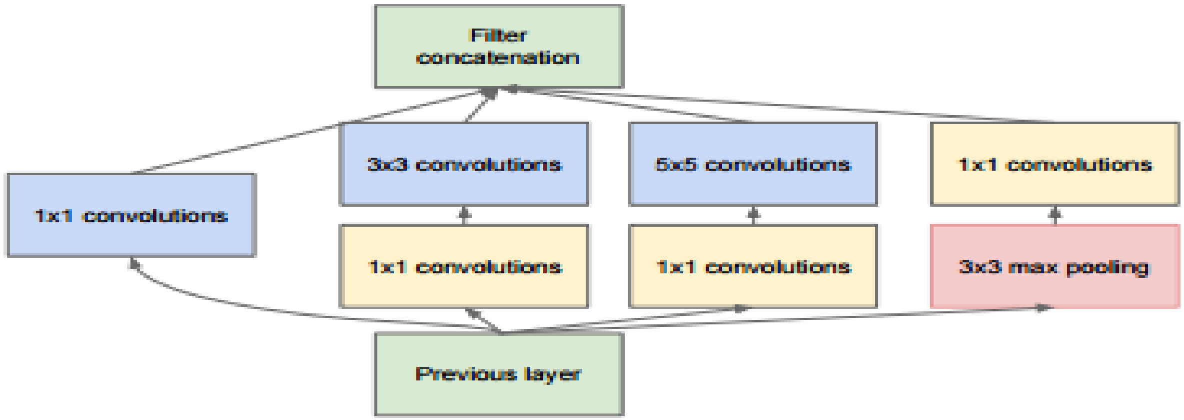

Figure 1: [2] Basic structure of inception module

2 Methods

2.1 Vanilla Deep Dream

To Perform the base version of deep dream on a selected image, we will normally proceed as follows.

1. We select a specific layer, normally in our case a inception layer, at which we will dream.
2. We feed the selected image forward until the layer at which, we would like to dream.
3. Back-propagate the gradients of a specified score function from the selected layer to the image as in the training stage of network.
4. Update the image with a gradient ascent.

More specifically, assuming , we have a colored image, which can be represented as a 3-dimensional array $\mathbf{I}_{i*j*3}$, i and j representing the height and width of the image respectively, and we choose to dream at inception-4c, for instance. Thus when we feed the image $\mathbf{I}$ through the network til the inception-4c layer, we will obtain a output from inception-4c layer, which will be another 3-dimensional array $\mathbf{F}_{h*w*n}$, h,w, and n representing the height of each feature map, width of each feature map, and total number of feature maps respectively.

Then we will define our score function as:

$$\mathbf{L} = \sqrt{\sum_{h,w,n} \mathbf{F}^2_{h,w,n}}$$

and Gradient with respect to $\mathbf{I}$ will be:

$$\mathbf{G} = \frac{d\mathbf{L}}{d\mathbf{I}}$$

which is a 3-dimensional array with the same shape as image $\mathbf{I}$, and finally we will update the original image $\mathbf{I}$ by:

$$\mathbf{L}^{new} = \mathbf{L}^{old} + \lambda \mathbf{G}_{normalized}$$

Where, λ is the step-size and $\mathbf{G}_{normalized} = \mathbf{G}/(\sum |\mathbf{G}|/(i*j*3))$ Above update process will be performed iteratively. In essence, what we want to achieve is to modify the images in order to boost the neuron activations. Thus, our original image will become closer and closer to what the chosen layer of the network wants to see and detect.

2.2 Guided Deep Dream

One of the major issue of above process, which will be illustrated with examples later, is that we lack direct control over the dreaming outcome. One way to improve upon this aspect is to use a second image to guide the dreaming process. In terms of math, we can assume we have a second image $\mathbf{M}$, the spatial size of the image can theoretically be arbitrary. Then, we feed the second image forward into the network until the selected layer, and we will obtain a second set of feature maps, denoted as $\mathbf{Y}$. Afterwards, we shall flatten both $\mathbf{F}$ and $\mathbf{Y}$ along depth direction, effectively turning both of them into matrices with n rows, and each row now contains the neuron activations of one feature map. We will denote the flattened $\mathbf{F}$ and $\mathbf{Y}$ as $\mathbf{F}_{flat}$ and $\mathbf{Y}_{flat}$ respectively.

Subsequently, we will define product between the transpose of $\mathbf{F}_{flat}$ and $\mathbf{Y}_{flat}$ as:

$$\mathbf{A} = \mathbf{F}_{flat}^{T} \mathbf{Y}_{flat}$$

By far, we can note each element of $\mathbf{A}$ is in fact a evaluation of the correlation between the neuron activations of a specific receptive field when image $\mathbf{I}$ is fed into the network and the neuron activations of a specific receptive field when image $\mathbf{M}$ is fed into the network. Thus, we can find:

$$\mathbf{I} = \text{argmax}(\mathbf{A}), \text{ along the column direction}$$

Then, we can use $\mathbf{I}$ to select corresponding columns of $\mathbf{Y}_{flat}$, and form a new matrix $\mathbf{T}_{m*n}$ using selected columns.

Finally, we define a new score function:

$$\mathbf{L} = \sqrt{\sum_{m,n} \mathbf{T}_{m,n}^2}$$

, and we will modify the image iteratively to maximize the above score function as we did before. In general, we are essentially trying to boost the neuron activations according to the activations of a second image, hence guiding the dreaming process.

3 Discussion

3.1 Initial Motivation

The original motivation behind deep dream is to help us better understand how a CNN sees the world, since the whole process of deep dream is about fleshing out what the network, or more specifically one or a set of feature maps, want to detect. Figure 3 shows the deep dreamed Figure 2 on different inception layer. Generally, we can note from the fleshed out patterns, lower inception layers are more oriented to detect low level features such as edges and texture, while higher inception layers tend to like more explicit shapes. Furthermore, this idea can be effectively translated to visualize any feature map within the network, in which we are interested.

Figure 2: Base image of Figure 3.

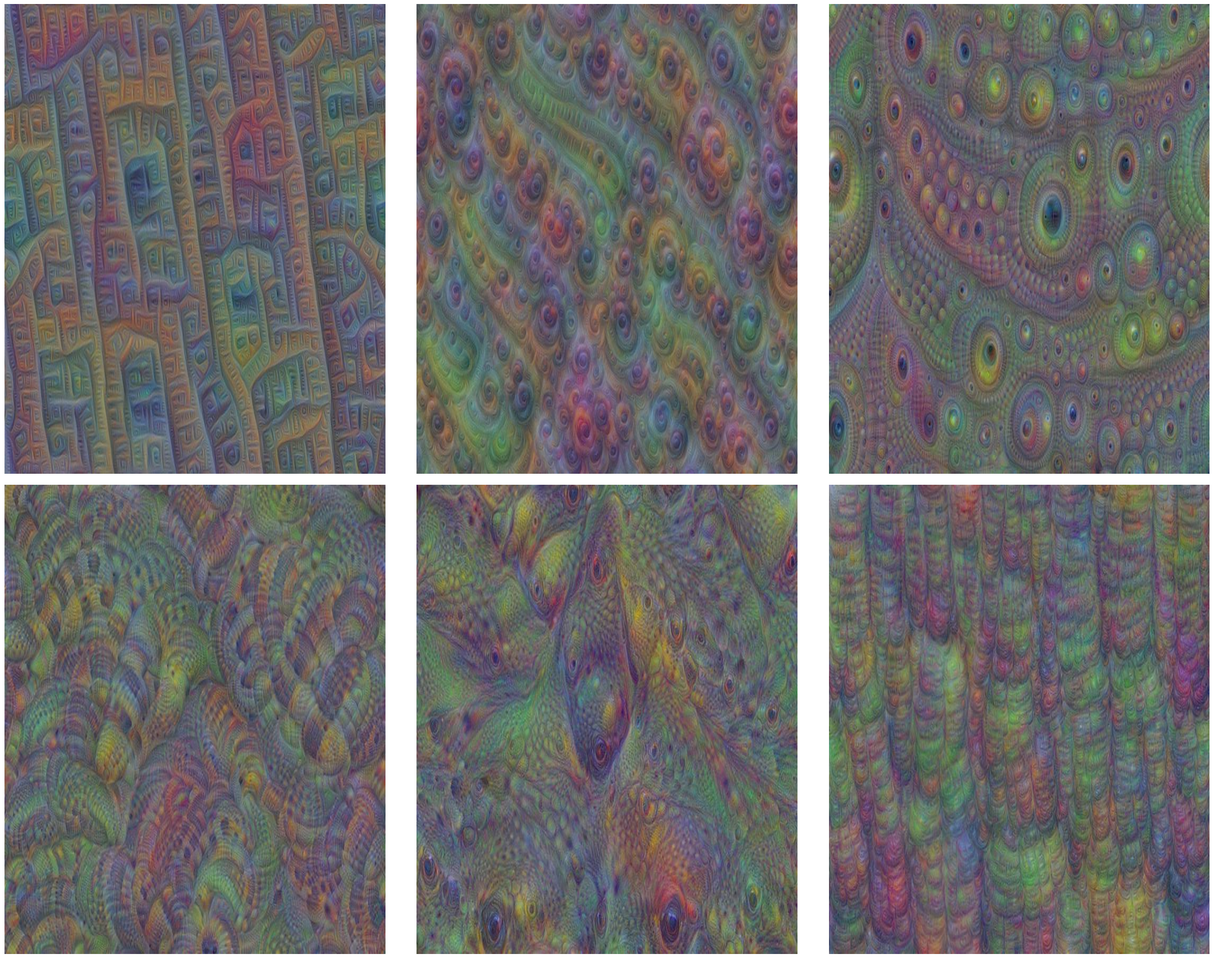

Figure 3: Vanilla Deep Dream on white noise. Top row from left to right: inception-3a, inception-3b and inception-4b. Bottom row from left to right: inception-4c, inception-4d and inception-4e .

3.2 Subset Dreaming

One of the major drawback of vanilla deep dream is the lack of variety, and sometimes it would produce overly creepy imagery. For instance, Figure 4 shows a dreamed image at inception-4d, and we can see, the dreamed version although has its own style, is overly-occupied by all kinds of amalgamations of various animals. This behavior is pretty common when we use deep dream to generate new images, as normally the trained model would be trained on ImageNet data set, Which consists of very large number of animal photos. Thus our network would inherently be sensitive towards features related to animals. However, sometimes this effect might not be as desirable as it would be in some other cases. One effective way to alleviate this issue without retraining network on brand new data set is to subset the feature maps we dream at. The intuitions are although our network in general likes to detect animals, not every feature map in the selected layer is as excited as some other feature maps in terms of detecting animals. Thus if we happen to have selected those maps, which are not focused in detecting animals, we should be able to obtain more organized dreamed images, and since we are only selecting a subset of feature maps, it's also effective to reduce the cluttering of different shape and styles. By comparing the dreamed images shown in Figure 5 to the dreamed image in Figure 4, we can clearly verify, by selecting different feature maps, at which we dream, we are able to create a great variety of new images with different art styles, even without actually changing the inception layer. Moreover, considering the sheer number of feature maps we have in each inception layer of the GoogLeNet, this trick also opens up a huge design space to explore, as not only we can dream upon individual feature map, but we can also dream upon a combination of several feature maps. A dreamed version of the base image shown in figure 4, on a combination of 55th feature map and 77th feature map can be found in the appendix section. However, one glaring issue with this approach is we have to manually experiment with different feature maps to have an idea of what kind of effects can be achieved on dreamed image, which might become cumbersome in some cases.

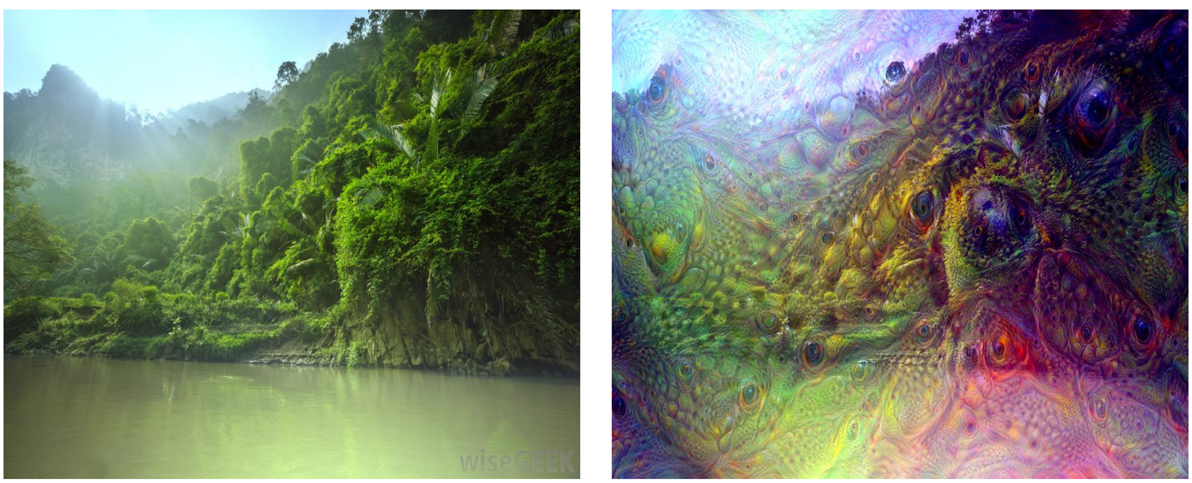

Figure 4: From left to right: A base image, and dreamed version of base image at inception-4d .

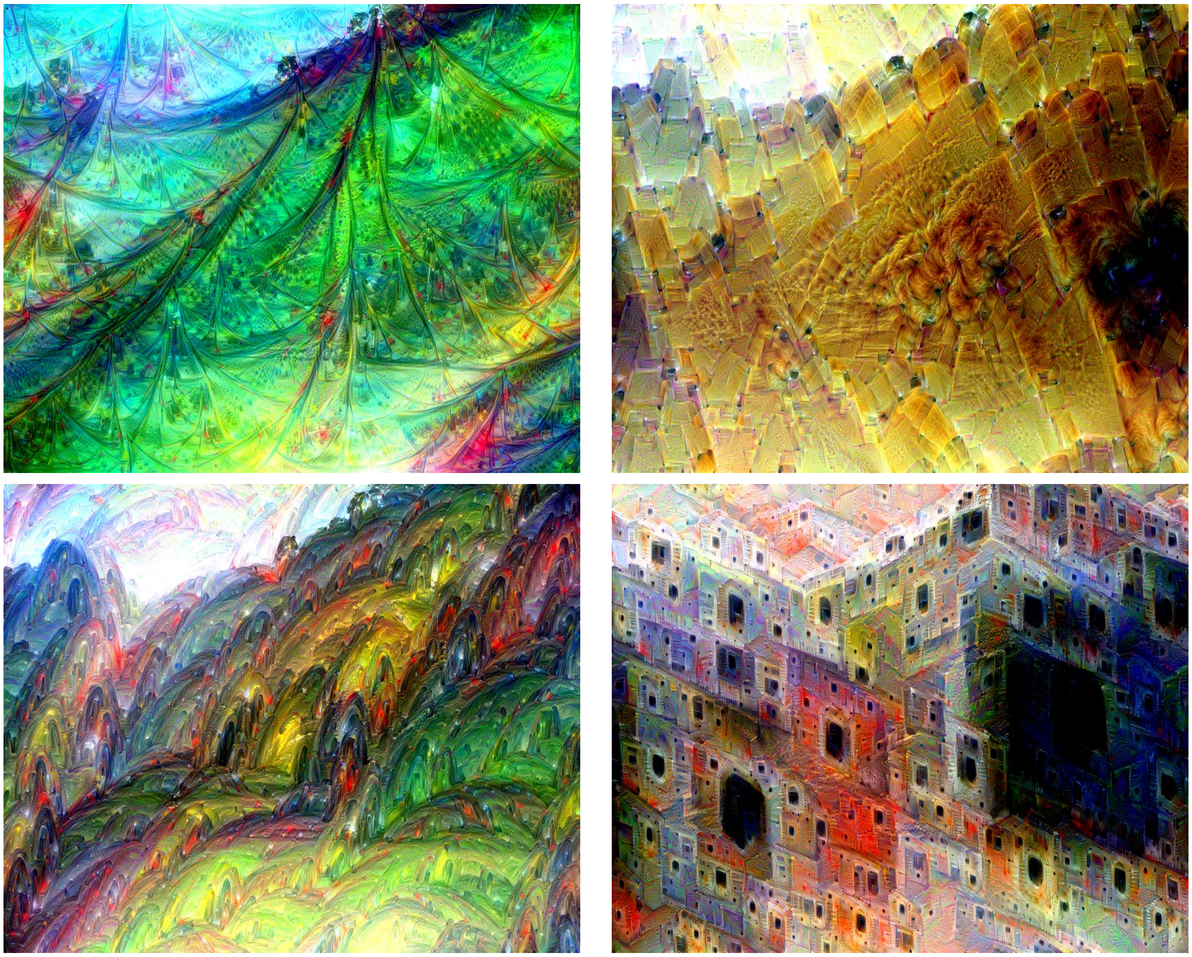

Figure 5: Deep dream with the same base image and inception layer as in Figure 4, but on different feature maps.Top row from left to right: 10th feature map and 22ed feature map. Bottom row from left to right: 55th feature map and 77th feature map

3.3 Guided Deep Dream In Action

Another effective approach to exact more control over the resulting dreamed image is using the guided dream process described in previous section. Figure 6 shows the dreamed version of a base image at inception-3b with vanilla deep dream. We can see obviously the resulting image is cluttered with animalistic features. Then, we apply a guide image, which is devoid of features related to dogs and animal in general, and still dream at the same layer. The resulting image, shown in Figure 7, is much more pleasant, as cluttering of animal features is significantly reduced compared to previous one.

Another aspect of guided deep dream is that it seems to be able to effectively combine the style of guide image, target image and inception layer, especially when we dream at a lower layer of the network. This effect can be observed in the resulting

Figure 6: From left to right: A base image, and dreamed version of base image at inception-3b

Figure 7: From left to right: The guide image, and dreamed version of base image at inception-3b

image in Figure 7. More examples of guided deep dream can be seen in Figure 8 and Figure 9

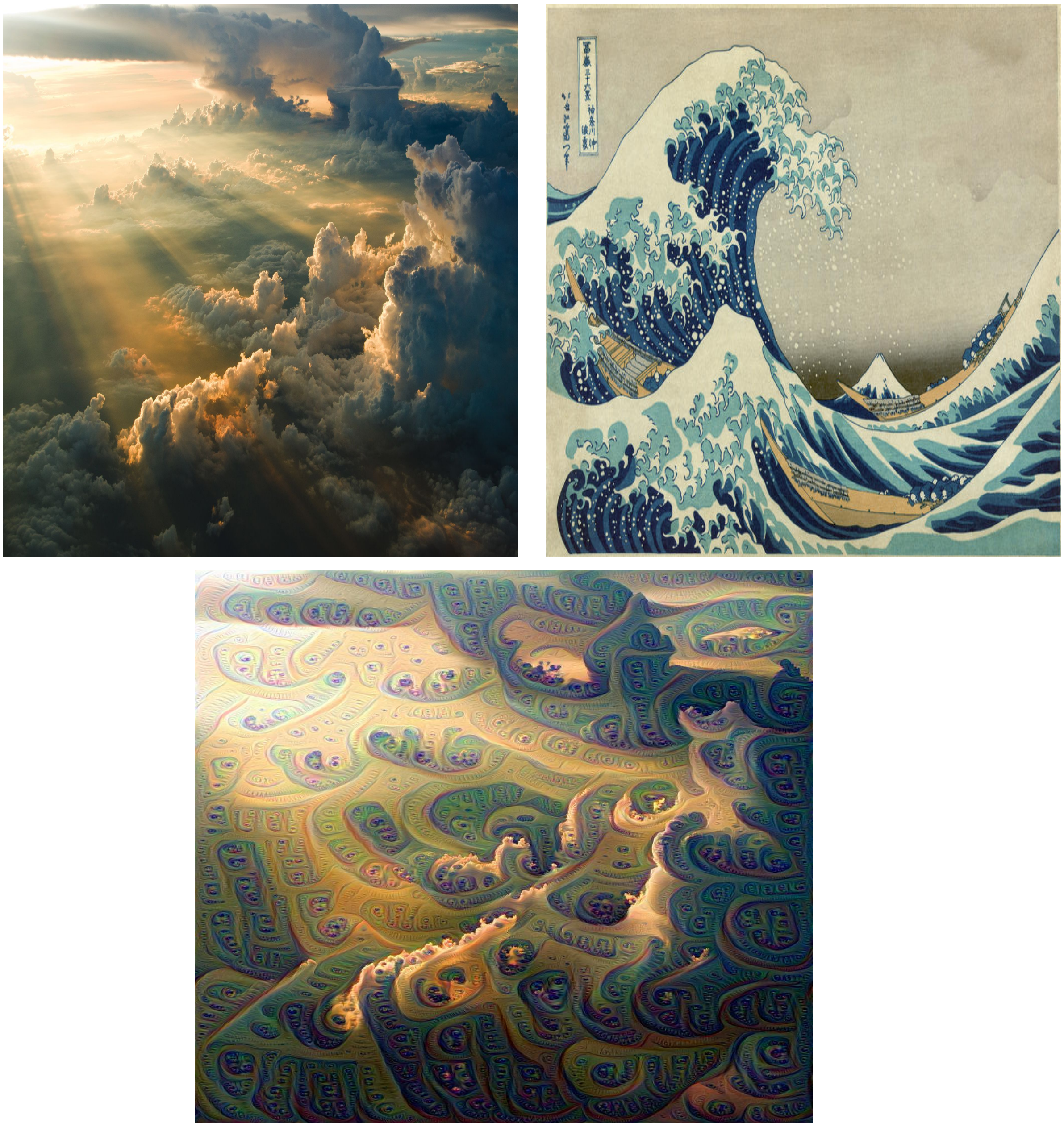

Figure 8: Top row from left to right: base image and guide image. Bottom row: dreamed version of base image at inception-3a

Figure 9: Top row from left to right: base image and guide image. Bottom row: dreamed version of base image at inception-4a

3.4 Summary and Future

Although the idea of deep dream is simple, it is also the fundamental idea employed in many other image generation techniques using convolutional neural networks, i.e. optimizing a score or loss function with respect to the actual image. Thus it can be combined with other techniques such as neural art style transfer, and it's also in itself a great technique to combine the style of different images and create new art style altogether as shown in above discussion. Moreover, even for people who have no desire to generate intriguing images, deep dream can also be effectively applied as a visualization tool to provide valuable insights into the function of a trained convolutional neural network. One of the aspects of deep dream , which can be further discussed in the future is what I call " reversed deep dream ", in which we try to suppress the neuron activations instead of boosting them. An example of reversed deep dream can be seen in the appendix section.

References

[1] Inceptionism: Going deeper into neural networks. `https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html`. Accessed: 2017-06-14.

[2] Christian Szegedy et al. Going deeper with convolutions. *arXiv:1409.4842*, 2014.

Appendices

Figure 10: Dreamed version of the base image used in Figure 4 at inception-4d 55th feature map + inception-4d 77th feature map

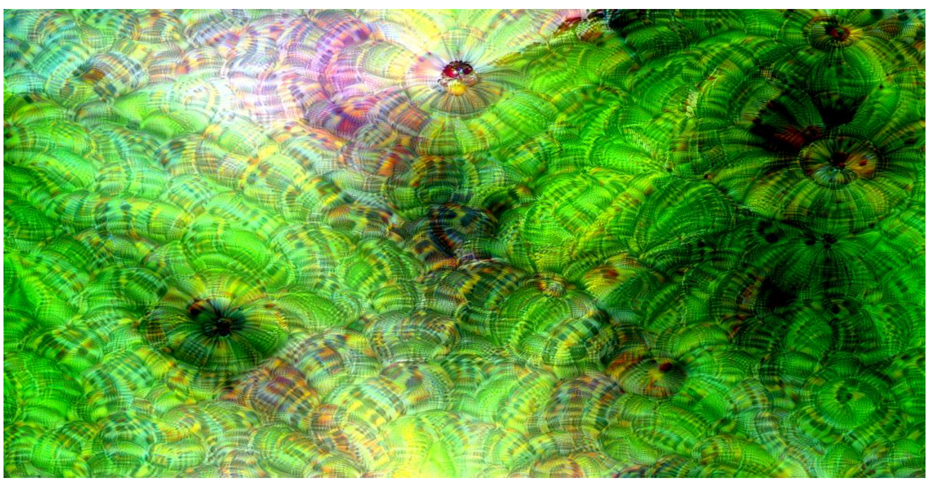

*Figure 11: Reversed-dream version of the base image used in Figure 4 at the top level 3*3 convolutional pre-relu layer within inception-4d. Note: we need to choose a pre-relu layer in order to have negative activations.*