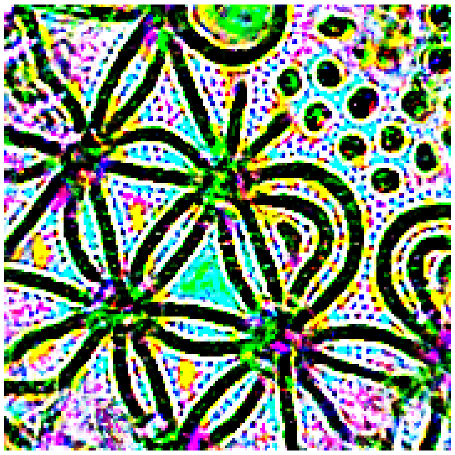
DeepMarianaTrench: a DeepDream implementation

Brian Pernut
CISE Department
University of Florida
Gainesville,Florida,United States
pernutbrian@gmail.com



ABSTRACT

In this work I aim to create an implementation of Google's DeepDream project that both improves on the quality of the "psychedelic" effect for the output pictures while being less computationally expensive and having a smaller dataset than the original version.

Specifically, we implement various convolutional neural networks with architectures derived from the VGG16/VGG19 architecture, where we introduce new layers such as rotations, dropout, and batch normalization, remove blocks of convolutional layers, change existing parameters, and more.

INTRODUCTION

Introduced in Google's blog post "Inceptionism: Going Deeper into Neural Networks ", DeepDream is a program that takes in an image and then returns a "psychedelic" version of that picture

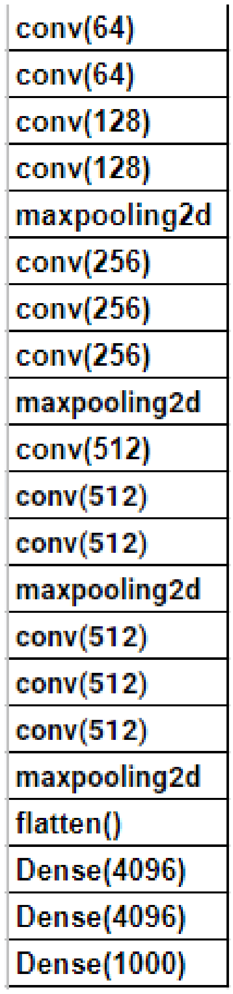
This occurs because each layer (or layer(s) selected) of the model amplifies the feature it is looking for. If any input resembles a feature, for example if a patch of grass is identified to be similar to fur of an animal, it will make it look more like the fur of an animal, which then leads the algorithm to detect it as fur of an animal, and the process repeats itself. This same is true for other features such as lines, curves, eyes, etc…

DeepDream is implemented by using a convolutional neural network as your "base" model, normally these are pre-trained CNN's (ex. InceptionV3 and VGG16) which will search for features, then a secondary model performs the "dreaming", which then performs gradient ASCENT on the images

METHOD

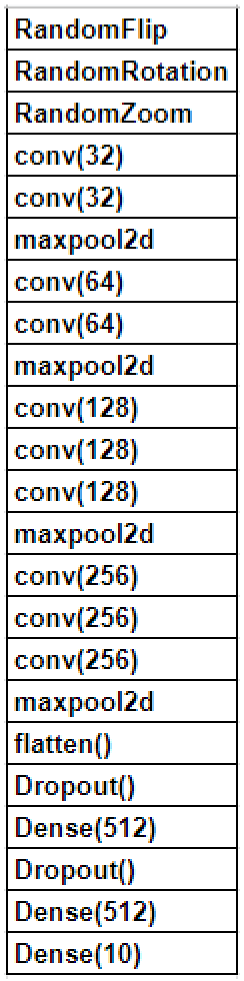
My implementation, DeepMarianaTrench, takes the same approach of the original DeepDream. It uses two models, a CNN with an architecture similar to VGG16/VGG19, and a model that performs the gradient ascent.

VGG16



conv(64)
conv(64)
conv(128)
conv(128)
maxpooling2d
conv(256)
conv(256)
conv(256)
maxpooling2d
conv(512)
conv(512)
conv(512)
maxpooling2d
conv(512)
conv(512)
conv(512)
maxpooling2d
flatten()
Dense(4096)
Dense(4096)
Dense(1000)

DeepMarianaTrench



RandomFlip
RandomRotation
RandomZoom
conv(32)
conv(32)
maxpool2d
conv(64)
conv(64)
maxpool2d
conv(128)
conv(128)
conv(128)
maxpool2d
conv(256)
conv(256)
conv(256)
maxpool2d
flatten()
Dropout()
Dense(512)
Dropout()
Dense(512)
Dense(10)

*note that layers are not equal in amount. pictures were distorted to preserve uniformity on the paper.

*all layers except the final dense,maxpools, and dropouts, have 'relu' as their activation function

The architecture for the models are indeed similar but have differences to allow us to train the classification model faster and with our own dataset, specifically the animals-10 dataset, which has images of butterflies, cats, chickens, cows, dogs, elephants,horses, sheep, spider, and squirrels.

The first change was that of the output layer from 1000 neurons to 10. This was done because VGG16 was originally trained on the imagenet dataset, which holds 1000 classes, while mine holds 10.

The second change was the removal of blocks of conv2d layers. The final 2 blocks of conv2d(512) (6 total convolutional layers) were removed, and one was eventually replaced by a 3 layer conv2d(32) block, placed before the conv2d(64) block.

The changes after these were implemented for the sake of improving the model by preventing overfitting (will elaborate later in the paper). Improving the classification ability of our model is crucial to recreating realistic psychedelic effects in our input pictures

These changes to prevent overfitting include the data augmentation layers which introduce new data constantly to the model, introducing an L2 regularizer as a kernel regularizar for the convolutional layers, and introducing dropout to the final block of dense layers.

Changes were also made during creation/testing that were later removed because they seemed to either have no effect on accuracy (training or validation), or were found to increase training time without any improvement. One of these changes were batch normalization layers. The introduction and removal of these layers had little impact compared to the changes stated above that were to deal with overfitting.

PREVIOUS WORK

Closely related works include the original DeepDream blog post by Alexander Mordvintsev. While other works on the topic of deep dreaming have been pursued, the topic, at least for static visual images has more or less been perfected. At this point, improvements to be made come from more accurate CNNs or less intensive CNNs that allow for users to get even more intense images without requiring more resources..

While work has been stagnant for static images, new versions of DeepDream that are applied to videos and audio have been created. Video,((Moniz, Kang, Póczos, 2013)). Audio, (Roberts, Resnick, Ardila, Ech, 2013).

RESULTS

Architecture/Model Version	Diverging Epoch	Val_Acc @ divergence
(32,64,128)-B&W -Dense(100)	~70	45%
(32,64,128,256)-B&W -Dense(128)	~100	52%
(32,64,128,256)-COLOR-Dense(256)	~85	53%
(32,64,128,256)-COLOR-Dense(512)	~105	53%
(32,64,128,256)-COLOR-Dense(512) w/ L2 and SGD	~75	57%

Above we have a few of the models that have been trained, along with the epoch that the training accuracy and validation accuracy began to diverge, alongside the val_Acc at divergence. For this paper we count a model as starting to diverge when the training accuracy is 3-4% better than the validation accuracy.

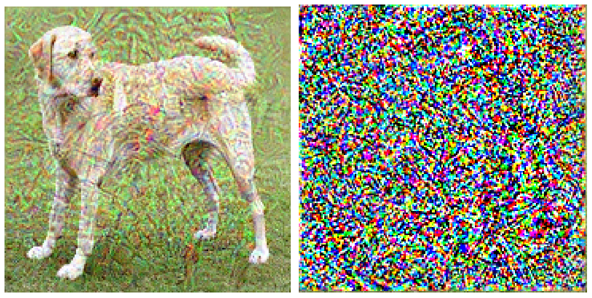
The reason this criteria was arbitrary and chosen to save time training. When training the model, the training and validation accuracy are very closely intertwined, around give or take 1%. It was a pattern in previous models that whenever the model would begin to diverge slightly (3-4%) it would continue to diverge and validation accuracy would stagnate.

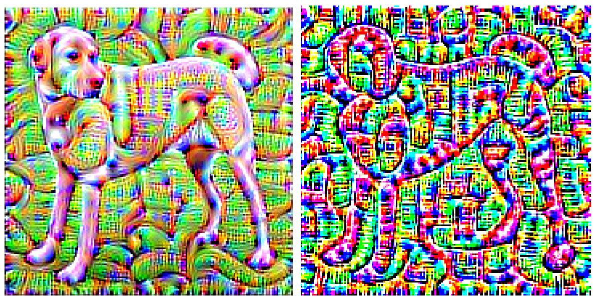
The first observation that is apparent from this chart is the fact that the number of convolutional blocks seems to greatly impact the accuracy of our model (Makes sense as to why VGG16 uses 5 and why VGG19 makes the blocks bigger). Our model responded mostly the same in terms of when it began to overfit regardless if images were black and white, or the number of neurons in the dense layer were increased.

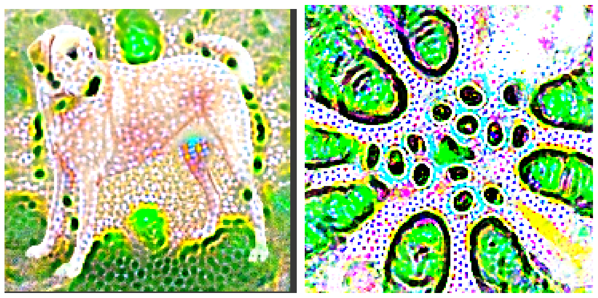
We also notice that we were able to increase the accuracy of the model in LESS epochs by introducing L2 regularization and introducing stochastic gradient descent for the optimizer (we were previously using 'adam'). For reference our SGD had a learning rate of 0.05 and a momentum value of 0.1.

The following images are taken from DeepDream applied with InceptionV3, VGG16, and DeepMarianaTrench respectively.

*base image


*images taken from InceptionV3


*images taken from VGG16

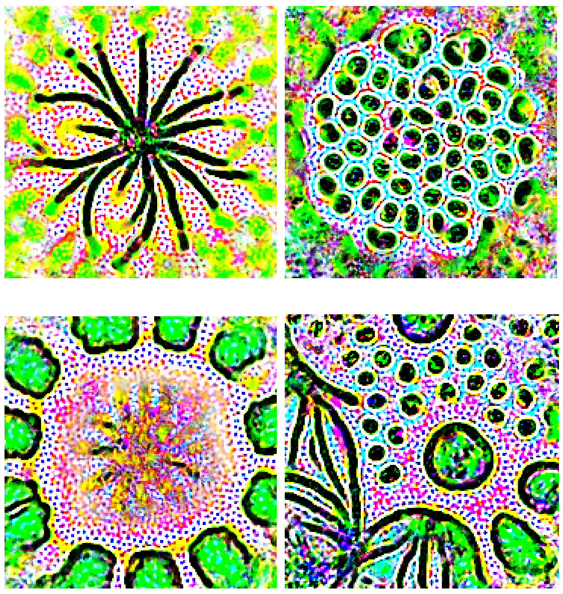

*images taken from DeepMarianaTrench

These images were taken from their respective CNN models. Remember that InceptionV3 and VGG16 were trained on the imagenet dataset while DeepMarianaTrench was trained on the animals-10 dataset, which holds fewer pictures total and classes, 1000 classes vs. 10 classes.

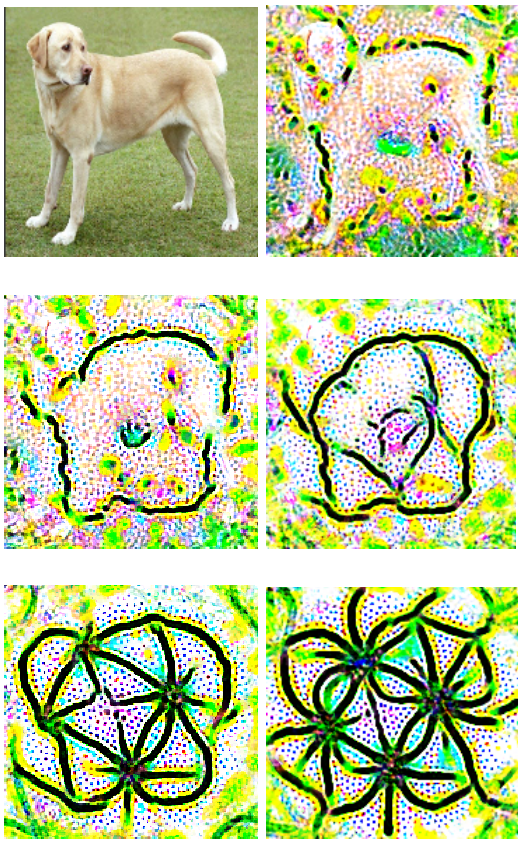
For these images ALL of the layers were excited.The images were run through 100 steps of deep dream, with the images on the left having a step size of 0.01 and the images on the right having a step size of 0.5. Step size affects the gradient and how much is added to the image in each step.

ANALYSIS

A visual analysis of the output of my model highlights that my model, if left to train, will create new abstract images that take on their own structures and designs, often symmetrical. This can be done by either increasing the number of steps of dreaming or increasing the step size. I prefer to increase the step size because it is a faster output but I also analyzed that different step sizes tend to create different image outputs for DeepMarianaTrench while for VGG16 and InceptionV3, the models retain more of the picture until they are entirely noise. It seems DeepMarianaTrench converges onto a structure/design instead of devolving into noise. The following images are output images from DeepMarianaTrench that I found visually appealing and interesting.



These images were created by a random selection of layers to excite, number of steps,step size, and pre-trained DeepMarianaTrench models. Some DeepMarianaTrench models were overfitted, some were stopped right as they began to diverge.



*analytical diagram

Here we have the progression of our base image over several hundred steps to analyze how the outputs join together and begin to create the “features” the model is looking for in the new output pictures, while to us they are new abstract images.

CONCLUSION

In conclusion, I was unable to create a program capable of creating “psychedelic” images. Instead I accidentally created a program that turned images into abstract art in its own unique and specific style.

I credit this to the fact that my CNN, DeepMarianaTrench, was unable to achieve high accuracy like VGG16 and InceptionV3 and also training on a much smaller dataset with classes that were more closely related than imagenet. My dataset is solely animals (mostly four legged mammals), while imagenet has a wider range of variety within its classes (animals,vehicles, other inanimate objects,etc…). My methods of preventing overtraining, while effective, only delayed the inevitable, although maybe this is what could have led to the current images that DeepMarianaTrench is outputting.

If I was to attempt this project again of creating psychedelic images, I would explore using a GAN or an autoencoder. A GAN with a dataset of images that have been psychedelicized, or an autoencoder that instead of denoising images, adds noise to them to the point they look like psychedelic images.

I would also have a more strict approach to data collection and model analyzing. For this project I just kept changing what I could and then testing to see if there were improvements and if there were I would keep going. It is difficult for me to pinpoint what made DeepMarianTrench overtime and thus hard for me to figure out what to change to improve the accuracy to the point it will allow me to create images like those from VGG16 and InceptionV3.

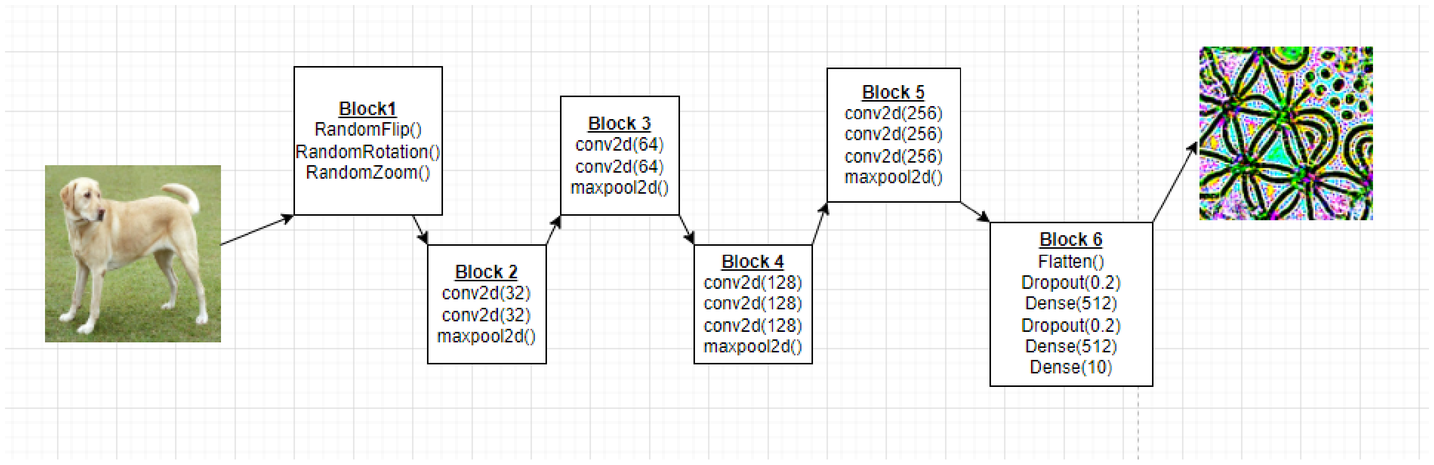
REFERENCES

[1] Moniz, J., Kang, E. and Póczos, B., 2016. *LucidDream: Controlled Temporally-Consistent DeepDream on Videos*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1911.11960>

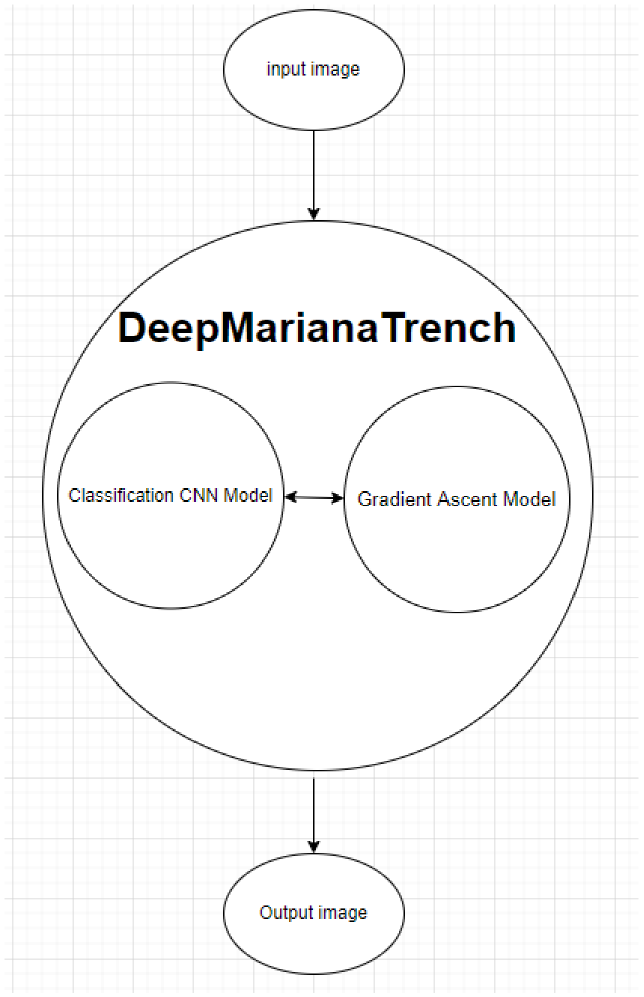
[2]Roberts, A., Resnick, C., Ardila, D. and Eck, D., 2016. *Audio Deepdream: Optimizing raw audio with convolutional networks*. [online] Google Research. Available at: <https://research.google/pubs/pub45859/>

[3] Mordvintsev, A., Olah, C. and Tyka, M., 2015. *Inceptionism: Going Deeper into Neural Networks*. [online] Google AI Blog. Available at: <https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html>

[4] Nguyen, A., Yosinski, J. and Clune, J., 2015. *Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1412.1897>



*illustrative diagram showing the at the layers an image goes through as it is converted into a DeepMarianaTrench image



*didactic diagram displaying the how DeepMarianaTrench interacts with itself and the enviroment

*analytical diagram is included in paper (pictures showing the progression of the models output