DeepDream: What I Dreamt While At College

Minh B. T. Nguyen
Department of Electrical and Computer Engineering
Virginia Tech
mnguyen0226@vt.edu

Abstract

"Dreams are messages from the deep" wear human's desires or regrets appears in human's conscious when they sleep. However, dreams tend to be disappeared if a person are unintentionally not reminding himself/herself about the dream. Dreams are occasionally shared with love-ones verbally since the images and actions in dreams are unpredictable with unexplainable behaviours. DeepDream algorithm allow a person to transform the image of the normal location to a dreamy-element-filled location. By applying gradient ascent, image pyramid, and gaussian smoothing, we can get vivid results [1].

1 Introduction

The main problem I am solving is to creating eye-pleasing art with deep learning. In additional, the DeepDream algorithm is a great tool to explain the behaviour of a deep neural networks on an image. DeepDream algorithm is an upgraded version of Inceptionism created by Google researcher, Alexander Mordvintsev, with a fascinating background story (1). In short, Alexander had a nightmare and woke up around 2AM, he then coded up the algorithm in Caffe and got interesting results.

2 Background

Interpretability in deep learning computer vision deployment project is occasionally ignore due to the black-box implementation, the focus on accuracy-related metrics, and the lack of powerful visualization tool (2). How does one checks whether the model has correctly learned the correct features? There are several attempts to mathematically understand deep learning model's prediction such as LIME (3), Attribution (4), and Grad-CAM (5). In addition, in recent years, the usage of deep learning to create artistic imagery demanded methods such as Neural Style Transfer (6). There is a lack of tool that can both doing model interpretability while generating artistic imagery.

I am interested in Explainable AI techniques since they help me debug during deep learning architecture designing and training process. Besides, as a student, I faced several stressful situations as well as solving joyful challenges which caused me to dreamed several nights. I would like to use deep learning to visualize and share my dream.

There are two main challenges: Building, training, testing ResNet model from scratch in large dataset while aiming for high accuracy; and creating eye-pleasing visualization.

There are two main contributions to the literature in this work: (1) to train VGG11, VGG16, VGG19, ResNet50, ResNet101, ResNet152 on MIT Indoor dataset, and (2) to apply DeepDream algorithm to the six trained models.

[1] https://github.com/mnguyen0226/deepdream_college/tree/main/results/gallery/class

3 Methodology

In this section, I will discuss and explain the dataset, the six models, and the DeepDream algorithm.

3.1 Dataset

The dataset I will use is MIT Indoor Scene (7) (8). According to "MIT Indoor Scenes" dataset published on kaggle.com, the dataset contain 67 "Indoor" categories with the total of 15620 images (8). It is noted that the number of images varies across categories but there are at least 100 images per category (8). The images are RGB and in .jpg format. There are a total of (67x80) 5360 images for training and (67x20) 1340 images for testing. Since the images are in different size, I will need to do some image reshaping. I chose this dataset because it is light while complex, it is used in several research papers (9), (10), (11), and its categories match with my dream theme.

Figure 1: Samples from MIT Indoor Scenes Dataset: Bar, Auditorium, Bowling Alley(8)

3.2 Models

I will implement the six classical deep learning architecture: VGG11, VGG16, VGG19, ResNet50, ResNet101, ResNet152. The choice of the six architectures is based on their high-performance on ImageNet dataset, their easy-to-understand and easy-to-implement in Pytorch. In his research, Alexander applied the pretrained VGG16 and ResNet50 (trained on ImageNet and Place365) (1). However, in this work, I will train all six architecture from scratch on the MIT Indoor dataset. The models will learn the distinct features of the dataset such as specific building-structure shape, which is ideal in the case I want to visualize what I dream at Virginia Tech. Since both VGG and ResNet has been explained in class, I will not explain the architecture in here. More detail of the architecture can be found at `https://paperswithcode.com/method/resnet` and `https://paperswithcode.com/method/vgg`.

3.3 DeepDream Algorithm

I will use the "DeepDream Algorithm" which is "an artistic algorithm where a pretrained CNN is fed an image and optimized to amplify the features it 'sees' in the image" (1). The main ideas is to apply "gradient ascent" at several activation function instead of gradient descent. In addition, the image pyramid and gaussian smoothing methods recommended by Aleksa Gordić which provide a more vivid "dreaming" will be applied (12).

In terms of gradient ascent, depending on the layer that we do gradient ascent, we will get different results. Specifically, if we do gradient ascent on the lower layers (first couple of network layers), we will get interesting patterns of edges and corners. On the other hands, if we do gradient ascent on deeper layer (layers near to the output layer), then we will get more complex features of the training set such as tables or buildings.

In terms of image pyramid, this technique will add additional details when we do gradient ascent. This is a digital image processing by subsampling and image with different sizes, called levels (13). The figure 11 demonstrate the image pyramid procedure. Due to the receptive field of neural network, when we feed the network with different size of the image, we will see different features in all of them, and by adding all those features, we will get a very detailed DeepDream results. After getting different levels of the image, I will stack them together in the center to get a detail-filled image.

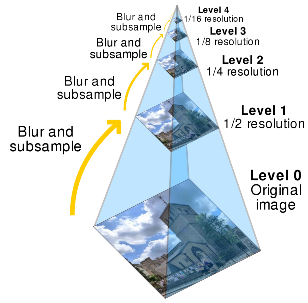

Figure 2: Image Pyramid

In terms of gaussian smoothing and normalization, this technique will make sure that our gradient will not explode. Specifically, if we do not apply gaussian filter to the gradient after doing gradient ascent at each layer, the gradient will quickly explode out of the image's boundary. After smoothing the image, we will do normalization of the gradient again.

3.4 Pipeline

Attached below is the end-to-end DeepDream pipeline. First the models are trained from scratch with MIT Indoor Dataset. The selected Virginia Tech image will then be fetched into image pyramid to get more details of the original image, and the processed images will be put into the trained models. Next, gradient ascent will be applied to output "dreamy features" of the image. Later, the gaussian filter will be apply to avoid gradient explosion outside of the image's boundaries.

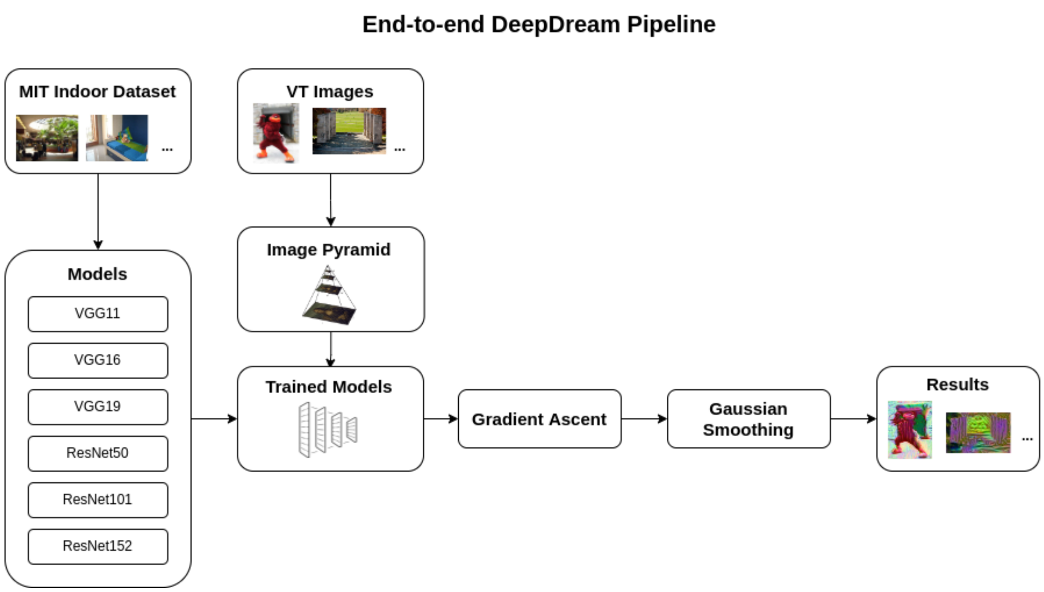

Figure 3: DeepDream Pipeline

For all VGG11, VGG16, VGG19, I will apply gradient ascent after each ReLU activation function. For all ResNet50, ResNet101, ResNet152, I will apply gradient ascent after ReLU activation function and bottleneck layers. The results will be shown in the next section. The code and results for this project is open-sourced [2].

[2] https://github.com/mnguyen0226/deepdream_college

4 Results

In terms of training models, for all six models, I trained with GPU for 100 epochs with train-test split of 80%-20%. Here is the training and testing results of each model on MIT Indoor dataset:

- VGG11 has training accuracy of 99 and testing accuracy of 55 with SGD optimizer.
- VGG16 has training accuracy of 99 and testing accuracy of 53 with SGD optimizer.
- VGG19 has training accuracy of 99 and testing accuracy of 49 with SGD optimizer.
- ResNet50 has training accuracy of 99 and testing accuracy of 45 with Adam optimizer.
- ResNet101 has training accuracy of 99 and testing accuracy of 43 with Adam optimizer.
- ResNet152 has training accuracy of 99 and testing accuracy of 44 with Adam optimizer.

In terms of DeepDream results, I will use Virginia Tech's Torgersen Bridge image as input image to all six deep learning architectures. I tried different combined configuration of DeepDream to get different results. The detailed configuration of each network can be found at [3]. All the DeepDream results will be added in the Appendix section. Besides Virginia Tech Torgersen Bridge image, I also experiment with different images which can be accessed at this gallery [4].

Figure 4: Torgersen Bridge Original Image

5 Discussion

In terms of the training and testing result, we can see that the model is overfit, partially because the models are not deep enough. This is understandable since the benchmark model of the MIT Indoor dataset is FOSNet with optimized features and layers for this dataset. There are rooms for testing accuracy improvements; however, with nearly 100% training accuracy on all dataset, this is very good for DeepDream algorithm.

There are no benchmark metric for DeepDream algorithm. However, from the experimental process, I found that certain settings work very well for DeepDream shown in figure 5:

- ResNet50 with gradient ascent after bottleneck layer 1.
- ResNet152 with gradient ascent after bottleneck layer 1.
- ResNet152 with gradient ascent after the first ReLU activation function.
- VGG11 with gradient ascent after the second ReLU activation function.
- VGG19 with gradient ascent after the second ReLU activation function.

[3] `https://github.com/mnguyen0226/deepdream_college/tree/main/src/notebooks/deepdream`

[4] `https://github.com/mnguyen0226/deepdream_college/tree/main/results/gallery`

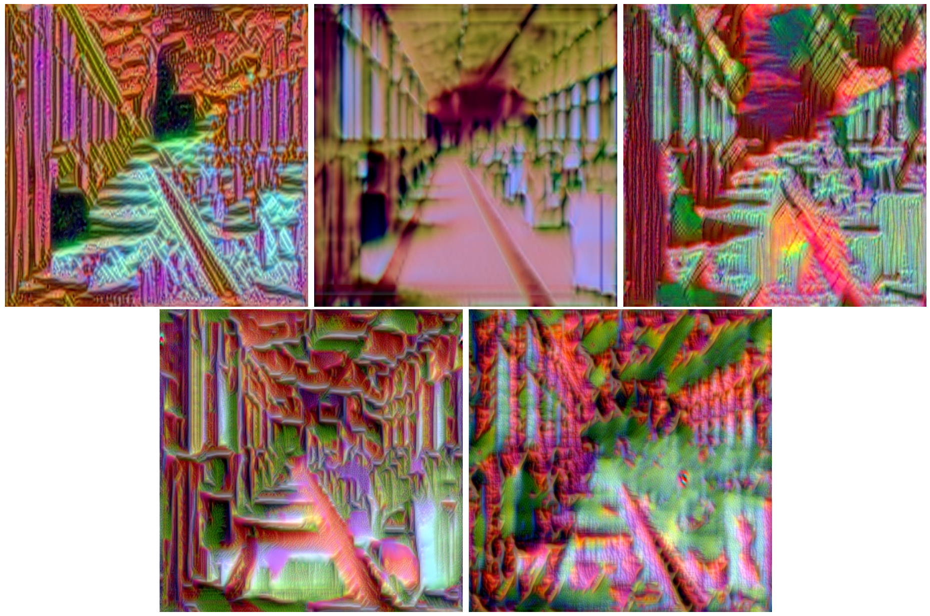

Figure 5: DeepDream Results From the Best Configurations

6 Conclusion

In this project, I have trained VGG11, VGG16, VGG19, ResNet50, ResNet101, ResNet152 with MIT Indoor dataset from scratch. DeepDream algorithm with several configurations have been applied and experience to provide mesmerized results. I hope that I can share with you my dreams while I am studying at Virginia Tech.

7 Acknowledgement

I would like to thanks Alexander Mordvintsev and Google Research team for publishing an inspiration blog post (1). I would like to thanks Aleka Gordic for his awesome published PyTorch implementation of DeepDream algorithm (12). I would like to say thanks to Dr. Xing from Statistic Department at Virginia Tech for his knowledge, unparalleled teaching energy, and awesome Deep Learning class.

References

[1] A. Mordvintsev, C. Olah, and M. Tyka, "Inceptionism: Going deeper into neural networks," 2015.

[2] X. Li, H. Xiong, X. Li, X. Wu, X. Zhang, J. Liu, J. Bian, and D. Dou, "Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond," *arXiv preprint arXiv:2103.10689*, 2021.

[3] S. Mishra, B. L. Sturm, and S. Dixon, "Local interpretable model-agnostic explanations for music content analysis.," in *ISMIR*, vol. 53, pp. 537–543, 2017.

[4] S. Jha, S. Raj, S. Fernandes, S. K. Jha, S. Jha, B. Jalaian, G. Verma, and A. Swami, "Attribution-based confidence metric for deep neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[5] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

[6] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 11, pp. 3365–3385, 2019.

[7] A. Quattoni and A. Torralba, “Recognizing indoor scenes,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 413–420, IEEE, 2009.

[8] M. Ahmad, “Indoor scene recognition cvpr,” *https://www.kaggle.com/itsahmad/indoor-scenes-cvpr-2019*, 2019.

[9] A. Yang, P. M. Esperança, and F. M. Carlucci, “Nas evaluation is frustratingly hard,” *arXiv preprint arXiv:1912.12522*, 2019.

[10] A. López-Cifuentes, M. Escudero-Viñolo, J. Bescós, and Á. García-Martín, “Semantic-aware scene recognition,” *Pattern Recognition*, vol. 102, p. 107256, 2020.

[11] M. Rahimzadeh, S. Parvin, E. Safi, and M. R. Mohammadi, “Wise-srnet: A novel architecture for enhancing image classification by learning spatial resolution of feature maps,” *arXiv preprint arXiv:2104.12294*, 2021.

[12] A. Gordić, “Pytorch implementation of deepdream algorithm,” *https://github.com/gordicaleksa/pytorch-deepdream*, 2019.

[13] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, “Pyramid methods in image processing,” *RCA engineer*, vol. 29, no. 6, pp. 33–41, 1984.

Appendices

Figure 6: VGG11 DeepDream Results

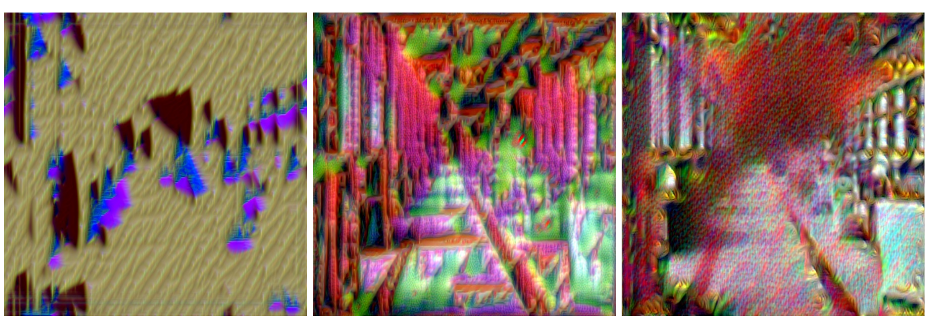

Figure 7: VGG16 DeepDream Results

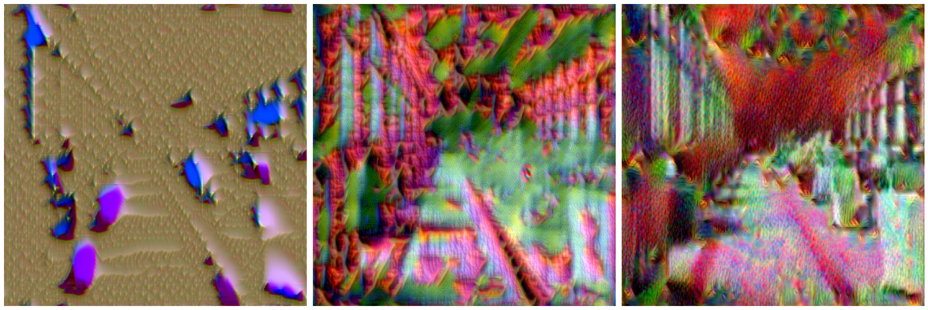

Figure 8: VGG19 DeepDream Results

Figure 9: ResNet50 DeepDream Results

Figure 10: ResNet101 DeepDream Results

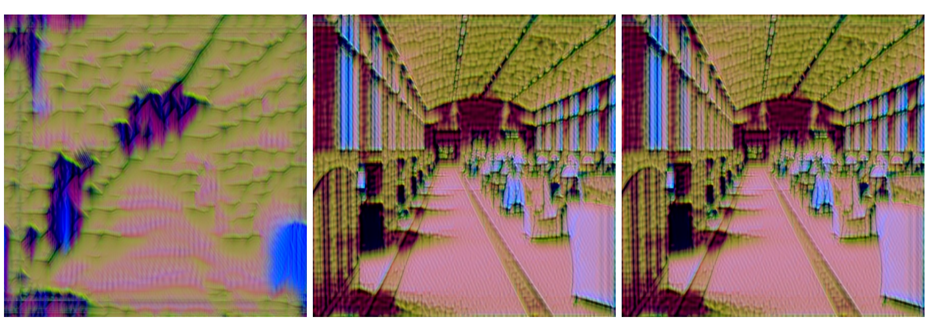

Figure 11: ResNet152 DeepDream Results