LOCAL EXPLANATION METHODS FOR DEEP NEURAL NETWORKS LACK SENSITIVITY TO PARAMETER VALUES

Julius Adebayo, Justin Gilmer, Ian Goodfellow & Been Kim
Google Brain

ABSTRACT

Explaining the output of a complicated machine learning model like a deep neural network (DNN) is a central challenge in machine learning. Several proposed local explanation methods address this issue by identifying what dimensions of a single input are most responsible for a DNN's output. The goal of this work is to assess the sensitivity of local explanations to DNN parameter values. Somewhat surprisingly, we find that *DNNs with randomly-initialized weights produce explanations that are both visually and quantitatively similar to those produced by DNNs with learned weights.* Our conjecture is that this phenomenon occurs because these explanations are dominated by the lower level features of a DNN, and that a DNN's architecture provides a strong prior which significantly affects the representations learned at these lower layers.

1 INTRODUCTION

Understanding how a trained model derives its output, as well as the factors responsible, is a central challenge in machine learning (Vellido et al., 2012; Doshi-Velez et al., 2017). A local explanation identifies what dimensions of a *single input* was most responsible for a DNN's output. As DNNs get deployed in areas like medical diagnosis (Rajkomar et al., 2018) and imaging (Lakhani & Sundaram, 2017), reliance on explanations has grown. Given increasing reliance on local model explanations for decision making, it is important to assess explanation quality, and characterize their fidelity to the model being explained (Doshi-Velez & Kim, 2017; Weller, 2017). Towards this end, *we seek to assess the fidelity of local explanations to the parameter settings of a DNN model.* We use random initialization of the layers of a DNN to help assess parameter sensitivity.

Main Contributions

- We empirically assess local explanations for faithfulness by re-initializing the weights of the models in different ways. We then measure the similarity of local explanations for DNNs with random weights and those with learned weights, and find that these sets of explanations are both visually and quantitatively similar.
- With evidence from prior work (Ulyanov et al., 2017; Saxe et al., 2011), we posit that these local explanations are mostly invariant to random initializations because they capture low level features that are mostly dominated by the input.
- Specifically, we hypothesize that dependence of local explanations on lower level features is as follows: if we decompose the function learned by a DNN as $f(g(x;\gamma);\theta)$, where $g(x;\gamma)$ corresponds to the function learned by the lower layers, and $f(.;\theta)$ corresponds to the upper layers, then a local explanation, $E(x_t)$, for input x_t corresponds to $E(x_t) \propto h(g(x_t;\gamma))$, where h captures the intricacies of the local explanation methodology.

2 LOCAL EXPLANATION METHODS

In this section, we provide an overview of the explanation methods examined in this work. The local explanation methods examined in this work derive explanations as follows: given an input $x_t \in \mathbb{R}^d$,

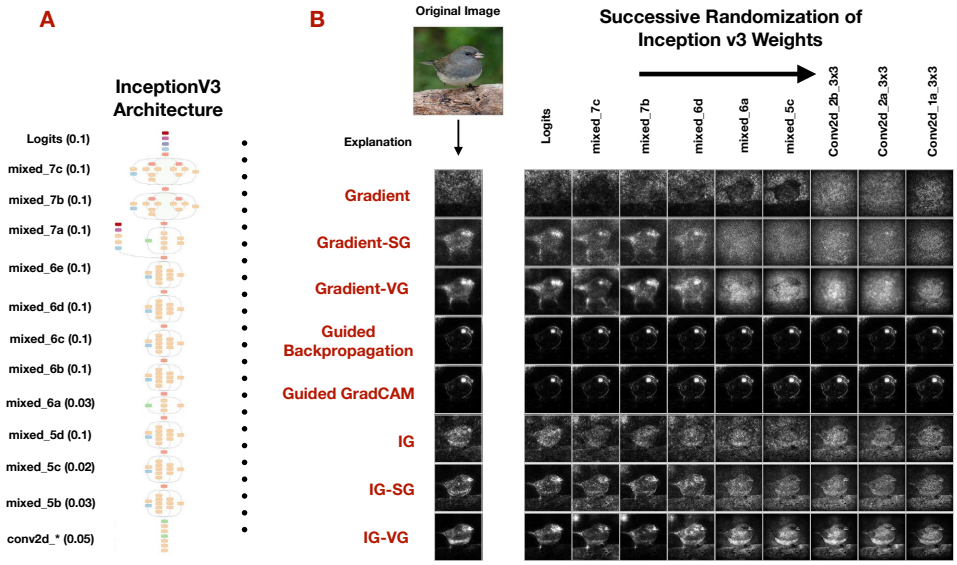

Figure 1: **Change in explanations for various methods as each successive inception block is randomized, starting from the logits layer. A**: Inception v3 architecture along with the names of the different blocks. The number in the parenthesis is the top-1 accuracy of the Inception model on a test set of 1000 images after randomization up that block. Initial top-1 accuracy for this class of images was 97 percent. Conv2d* refers collectively to the last 5 convolutional layers. **B-Left**: Shows the original explanations for the Junco bird in the first column as well as the label for each explanation type shown. **B-Right**: Shows successive explanations as each block is randomized. We show images for 9 blocks of randomization. Coordinate (Gradient, mixed7b) shows the gradient explanation for the network in which the top layers starting from Logits up to mixed7b have been reinitialized. The last column corresponds to a network where all the weights have been completely reinitialized. See Appendix for more examples.

and a DNN that computes a function $S(x_t) \in \mathbb{R}^C$, where C is the total number of output neurons, we seek an explanation $E_{ct} \in \mathbb{R}^d$, where e_{it}–the ith dimension of E_{ct}–quantifies the importance of each input dimension x_{it} to the class-specific output $S_c(x_t)$.

2.1 Overview of Methods

Gradient with respect to input (Input-Output Gradient). The input-output gradient explanation for an input, x_t, is defined as: $E_{grad}(x_t) = \frac{\partial S_c(x)}{\partial x_t}$, which is a local linear approximation of $S_c(x)$ around x_t (Baehrens et al., 2010; Simonyan et al., 2013).

Integrated Gradients (IG). Sundararajan et al. (2017) address the gradient saturation problem of the input-output gradient by summing over scaled versions of the input. The integrated gradient explanation for an input x_t is defined as: $E_{IG}(x_t) = (x_t - \bar{x}) * \int \frac{\partial S_c(\bar{x}+\alpha(x_t-\bar{x})}{\partial x_t} d\alpha$, where $\bar{x}$ is the baseline input that represents the absence of a feature in the original sample x_t.

Guided Backpropagation (GBP). GBP, Springenberg et al. (2014), builds on the 'Deconvnet' methodology (Zeiler & Fergus, 2014). GBP modifies the gradient computation process during the backward pass in a DNN; masking out negative contributions.

Guided Grad-CAM. Grad-CAM (Selvaraju et al., 2016) maps the predicted scores for an input to the last convolutional layer of a DNN. This involves taking the gradient of the class-specific target, $S_c(x)$, with respect to the feature map from the last convolutional layer as follows: $E_{gcam}(x_t) = ReLU(\sum_k \alpha_k^c A^k), \alpha_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial S_c(x)}{\partial A_{ij}^k}$, where A^k are the feature maps. For

pixel-level explanation granularity, the authors propose to combine Grad-CAM with GBP as follows: $E^*_{gcam-gbp}(x_t) = E_{gcam}(x_t) \odot E_{gbp}(x_t)$, where $\odot$ is an element-wise product.

SmoothGrad (SG). Smilkov et al. (2017) define SmoothGrad as $E_{sgt}(x_t) = \frac{1}{N}\sum_i^N E_{mt}(x_t + \mathcal{N}(0, \sigma^2))$, where E_t, is an explanation derived from a gradient-based explanation method. SmoothGrad augments IG, GBP, and the input-output gradient to reduce visual noise.

VarGrad (VG) is a variance analog to SmoothGrad. VarGrad is defined as: $E_{vgt}(x_t) = Var(E_{mt}(x_t + \mathcal{N}(0, \sigma^2))$.

Summary. The local explanation methods described so far are the ones considered in this work. Our selection criteria for the approaches included for analysis was based on ease of implementation, running time, and memory requirements. While there are several different approaches for obtaining local-explanations (Ribeiro et al., 2016; Fong & Vedaldi, 2017; Dabkowski & Gal, 2017; Zintgraf et al., 2017; Pieter-Jan Kindermans, 2018), recent work Marco Ancona (2018); Lundberg & Lee (2017) has shown there are equivalences among several of the previously proposed.

3 Local Explanation Specificity

In this section, we provide a discussion of the key results from out experiments. See attached Appendix for discussion of experimental details and additional figures demonstrating out results. First, we randomly reinitialize the weights of a DNN starting from the top layers going all the way to the first layer. Second, we independently reinitialize the weights of each layer. With these randomization schemes, we then visually, and quantitatively assess the change in local explanations for a model with learned weights and one with randomized weights. To quantitatively assess the similarity between two explanations, for a given sample, we use the Spearman's rank order correlation metric inspired by the work of Ghorbani et al. (2017).

Cascading Network Randomization - Inception v3 on ImageNet. As figure 3 indicates, guided back-propagation and guided grad-CAM show no change in the explanation produced regardless of the degradation to the network. We observe an initial decline in rank correlation for integrated gradients and gradients, however, we see a remarkable consistency as well through major degradation of the network, particularly the middle blocks. Surprisingly, the input-output gradient shows the most change of the methods tested as the re-initialization approaches the lower layers of the network. We observe similar results for a CNN and MLP on MNIST.

The architecture is a strong prior. Work in Saxe et al. (2011) show that features learned from CNNs with random weights perform surprisingly well in a downstream classification task when fed to a linear classifier. The authors showed that certain CNNs with random weights still maintain translation in-variance and are frequency selective. To further this point, Alain & Bengio (2016) find that the features extracted from a randomly-initialized 3-hidden layer CNN on MNIST (the same architecture that is used in this work) lead to a 2 percent test error accuracy. As part of their analysis, they find that the best performing randomly-initialized features are those derived right after the Relu activation of features derived from the first layer. Taken together, these findings highlight the following:

> *the architecture of DNN is a strong prior on the input, and with random initialization, is able to capture low-level input structure particularly for images.*

4 Conclusion

We empirically assess local explanations of a DNN derived from several different methods to find that DNNs with randomly-initialized weights produce explanations that are both visually and quantitatively similar to those produced by DNNs with learned weights. We posit that this phenomenon occurs because local explanations are dominated by lower level features, and that a DNN's architecture provides a strong prior which significantly affects the representations learned at these lower layers even for randomly-initialized weights.

ACKNOWLEDGMENTS

We thank Jaime Smith for providing the idea for VarGrad. We thank members of the Google PAIR team for open-source implementations of the local explanation methods used in this work and for providing useful feedback.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, pp. 6970–6979, 2017.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. 2017.

Finale Doshi-Velez, Mason Kortz, Ryan Budish, Chris Bavitz, Sam Gershman, David O'Brien, Stuart Schieber, James Waldo, David Weinberger, and Alexandra Wood. Accountability of ai under the law: The role of explanation. *arXiv preprint arXiv:1711.01134*, 2017.

Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *arXiv preprint arXiv:1704.03296*, 2017.

Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. *arXiv preprint arXiv:1710.10547*, 2017.

Paras Lakhani and Baskaran Sundaram. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology*, 284(2): 574–582, 2017.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pp. 4768–4777, 2017.

Cengiz ztireli Markus Gross Marco Ancona, Enea Ceolini. Towards better understanding of gradient-based attribution methods for deep neural networks. *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Sy21R9JAW`. accepted as poster.

Maximilian Alber Klaus-Robert Mller Dumitru Erhan Been Kim Sven Dhne Pieter-Jan Kindermans, Kristof T. Schtt. Learning how to explain neural networks: Patternnet and patternattribution. *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Hkn7CBaTW`.

Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Peter J Liu, Xiaobing Liu, Mimi Sun, Patrik Sundberg, Hector Yee, et al. Scalable and accurate deep learning for electronic health records. *arXiv preprint arXiv:1801.07860*, 2018.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM, 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Andrew M Saxe, Pang Wei Koh, Zhenghao Chen, Maneesh Bhand, Bipin Suresh, and Andrew Y Ng. On random weights and unsupervised feature learning. In *ICML*, pp. 1089–1096, 2011.

Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *arXiv preprint arXiv:1711.10925*, 2017.

Alfredo Vellido, José David Martín-Guerrero, and Paulo JG Lisboa. Making machine learning models interpretable. In *ESANN*, volume 12, pp. 163–172. Citeseer, 2012.

Adrian Weller. Challenges for transparency. *arXiv preprint arXiv:1708.01870*, 2017.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.

A SUCCESSIVE RANDOMIZATION VISUALIZATION

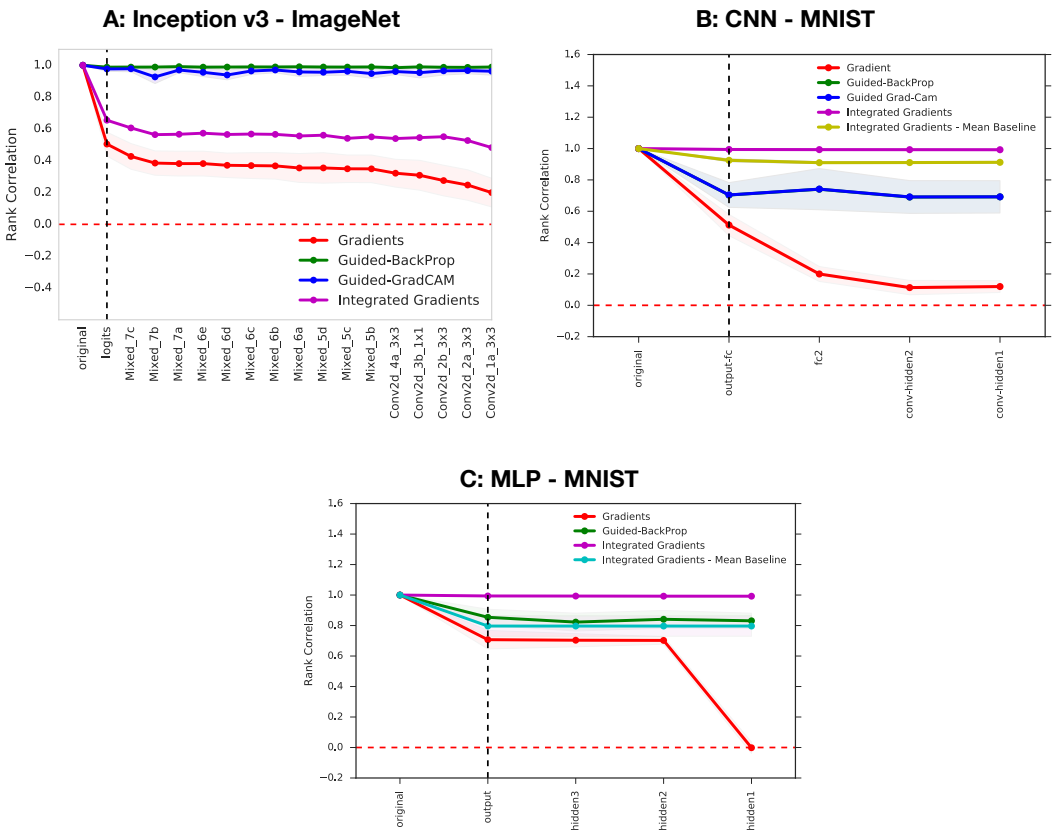

Figure 2: **Successive reinitialization starting from top layers for (A) Inception v3 on ImageNet, (B) CNN on MNIST, and (C) MLP on MNIST.** In all plots, y axis is the rank correlation between original explanation and the randomized explanation derived for randomization up to that layer or block (inception), while the x axis corresponds to the layers/blocks of the DNN starting from the output layer. The black dashed line indicates where successive randomization of the network begins, which is at the first layer. A: Rank correlation plot for Inception v3 trained on ImageNet. B: Rank correlation explanation similarity plot for a 3 hidden-layer CNN on MNIST. C: Rank correlation plot for a 3-hidden layer feed forward network on MNIST.

B BACKGROUND & EXPERIMENTAL DETAILS

We give an overview of the models and data sets used in our experiments.

B.1 EXPERIMENTAL DETAILS

Data sets. To perform our randomization tests, we used the following data sets: ILSVRC 2012 (ImageNet classification challenge data set) Russakovsky et al. (2015), and the MNIST data set LeCun (1998).

Models. we perform our randomization tests on a variety of models across the data sets previously mentioned as follows: a pre-trained Inception v3 model Szegedy et al. (2016) on ImageNet dataset, a multi-layer perceptron model (3 hidden layers) trained on the MNIST, a 3 hidden layer CNN also trained on the MNIST.

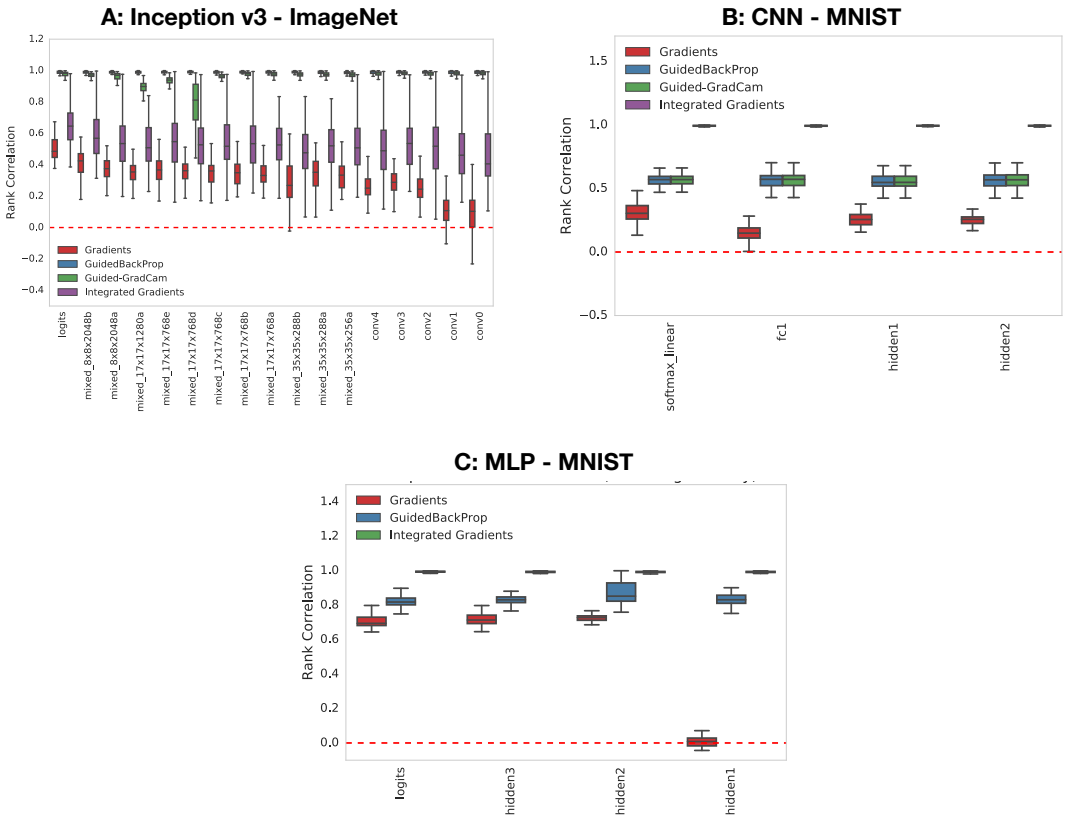

Figure 3: **Independent reinitialization of each layer (A) Inception v3 on ImageNet, (B) CNN on MNIST, and (C) MLP on MNIST.** In all plots, y axis is the rank correlation between original explanation and the randomized explanation derived for independent randomization of that layer or block (inception), while the x axis corresponds to the layers/blocks of the DNN starting from the output layer. The red dashed line indicates zero rank correlation. A: Rank correlation plot for InceptionV3 trained on ImageNet. B: Rank correlation explanation similarity plot for a 3 hidden-layer CNN on MNIST. C: Rank correlation plot for a 3-hidden layer feed forward network on MNIST.

Explanation Similarity. To quantitatively assess the similarity between two explanations, for a given sample, we use the Spearman's rank order correlation metric inspired by Ghorbani et al. (2017). The key utility of a local explanation lies in its ranking of the relevant of parts of the input; hence, a natural metric for comparing the similarity in conveyed information for two local explanations is the Spearman rank correlation coefficient between two different explanations.

For a quantitative comparison on each dataset, we use a test bed of 200 images. For example, for the ImageNet dataset, we selected 200 images from the validation set that span 10 highly varied classes ranging from Banjo to Scuba diving. For MNIST, we randomly select 200 images from the test data on which to compute explanations for each of the tested methods. In the figures shown indicating rank correlation between true and randomized models, each point is an average of the correlation coefficient over 200 images, and the 1-std band is shown around each curve to provide a sense for the variability of the rank correlation estimate.

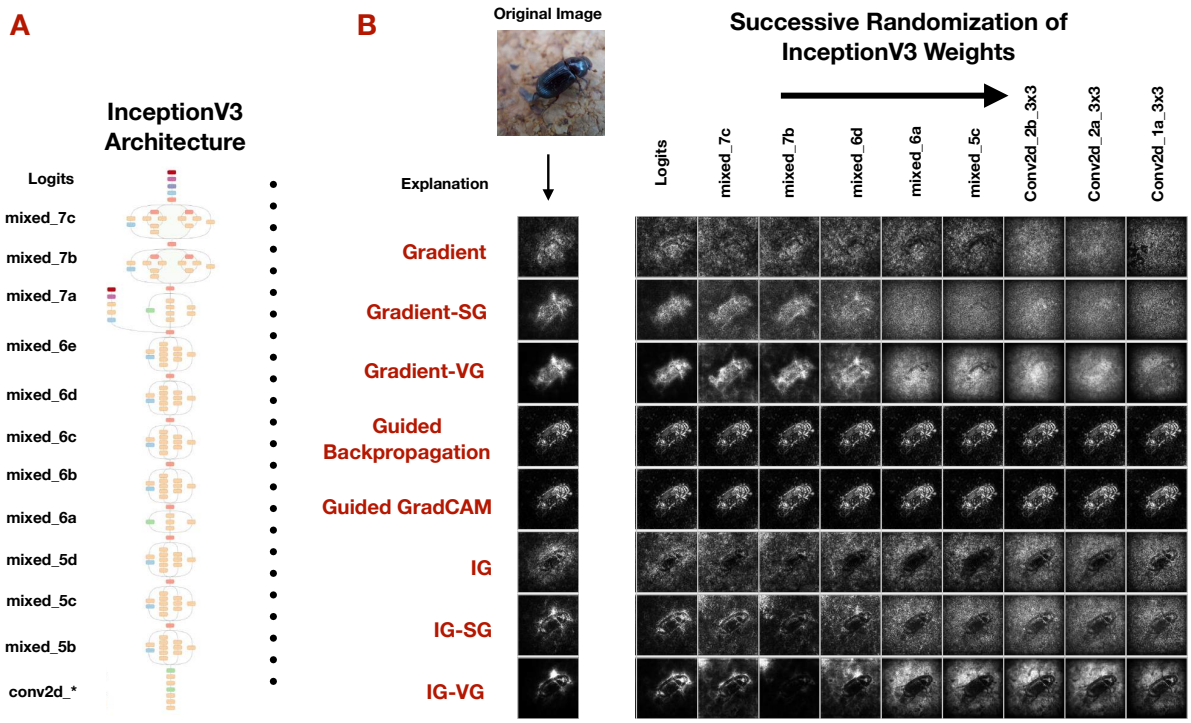

Figure 4: **Bug**

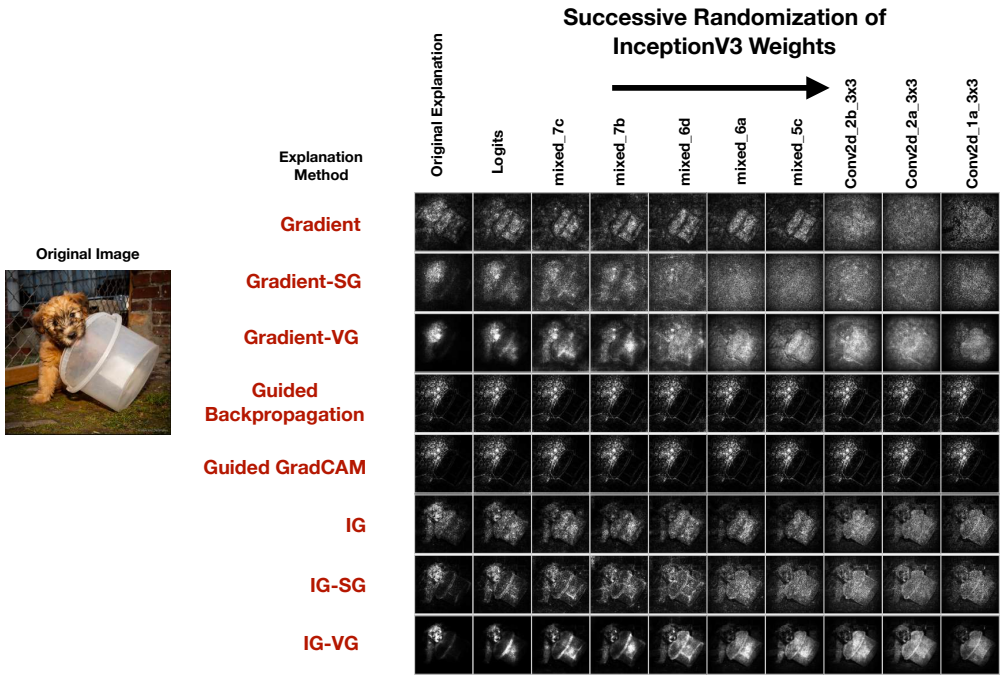

Figure 5: **Dog**

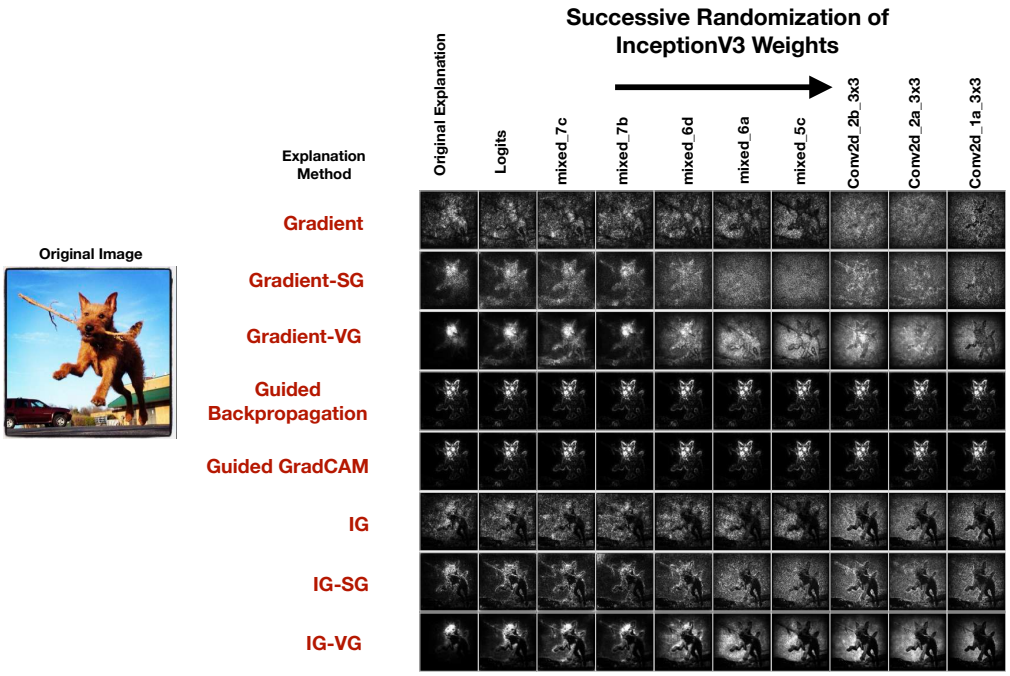

Figure 6: **Dog**

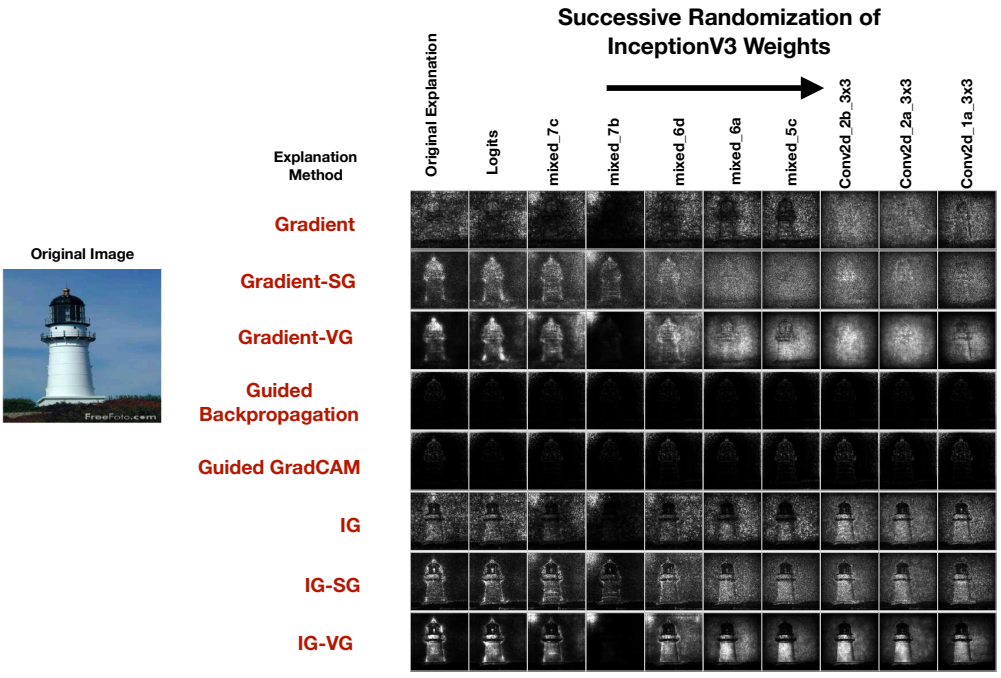

Figure 7: **Light House**

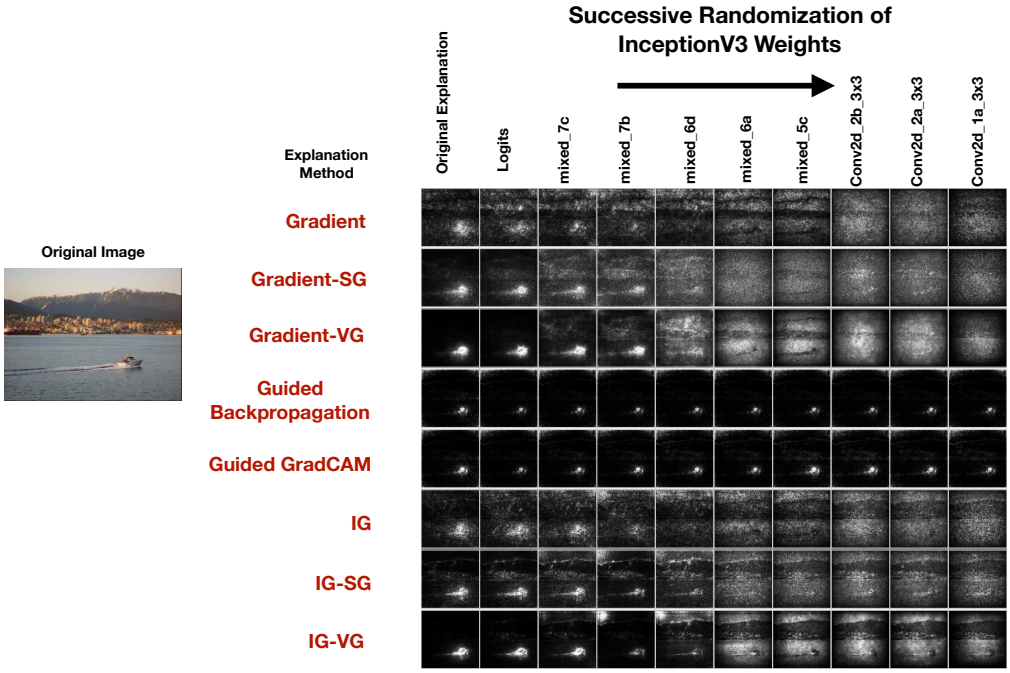

Figure 8: **Speed-boat**

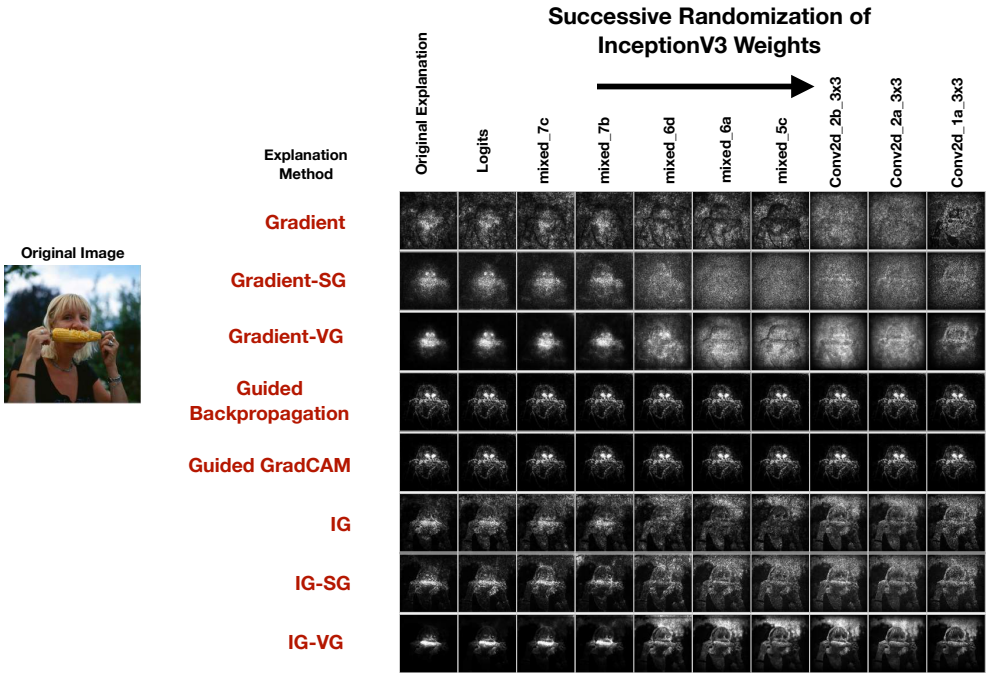

Figure 9: **Corn**