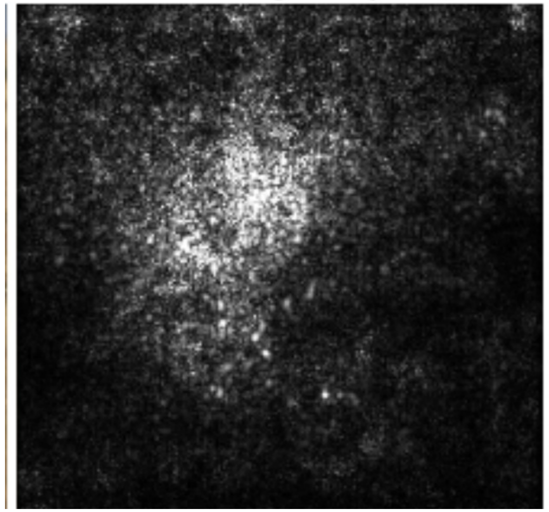
# SmoothGrad: removing noise by adding noise

**Authors:** Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, Martin Wattenberg

**Presented by:** Vignav Ramesh, Zelin (James) Li, Paul Liu

# Motivation

- We want **post-hoc explanations of image classifiers**
- *Solution*: **Sensitivity maps** (a.k.a. saliency maps, pixel attribution maps)
  - Visual interpretation of gradient of class activation function w.r.t input image
  - Structured as grayscale image w/ dimension same as input image
    - Brightness of pixel ∝ importance to classification decision

# Background: Gradients as Sensitivity Maps

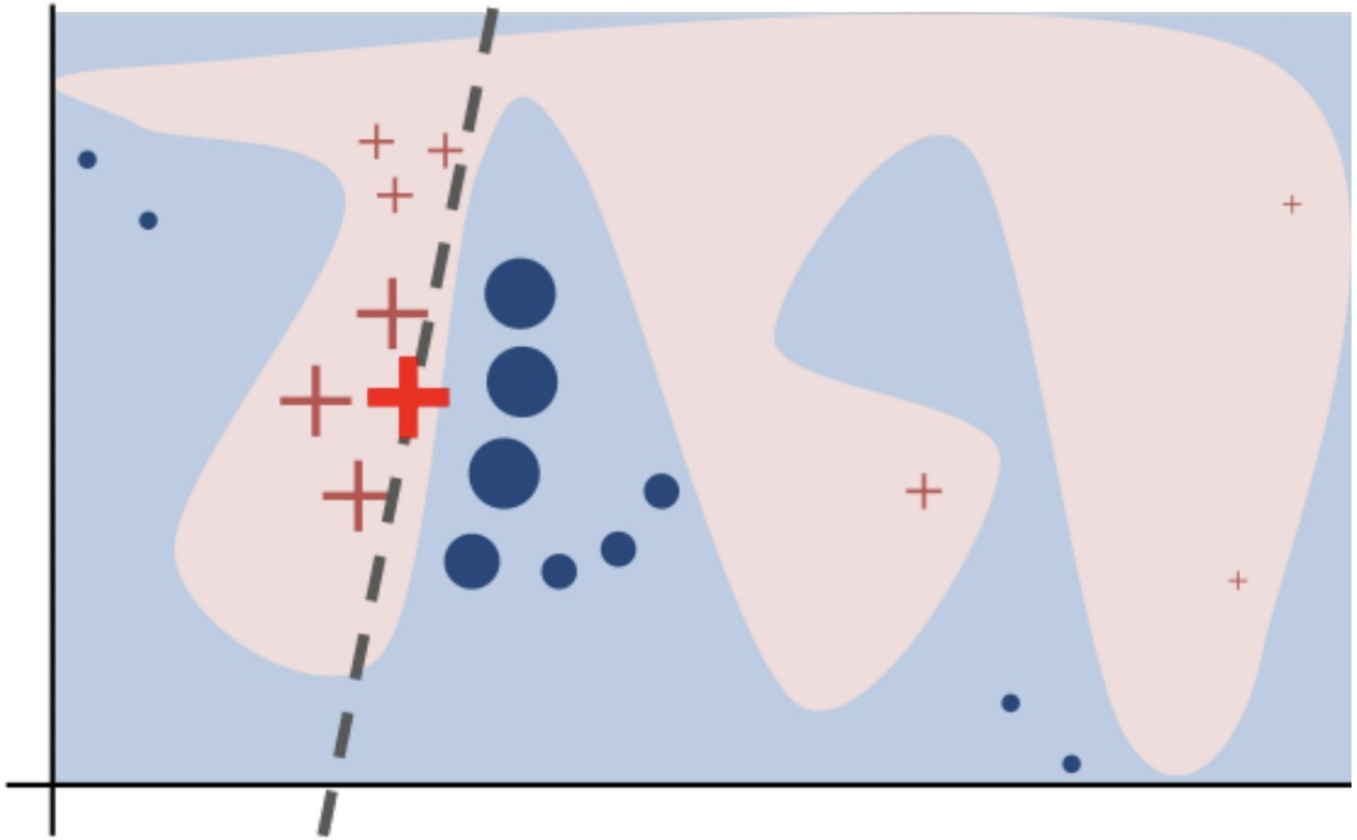
- $N$: network that classifies images into one class from set $C$
- Given input image $x$, $N$ typically computes class activation function $S_c$ for each $c \in C$
- Final classification $class(x) = \operatorname{argmax}_{c \in C} S_c(x)$
- **Sensitivity map given by**

$$M_c(x) = \partial S_c(x) / \partial x$$

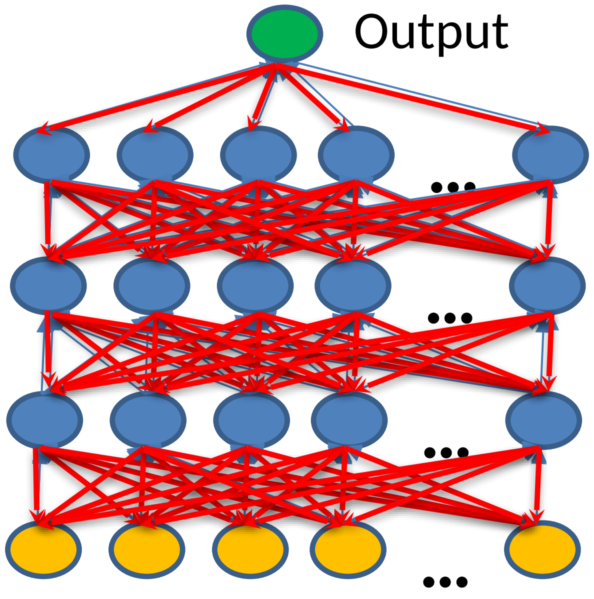
- ***Intuition***: $M_c$ represents <u>how much difference</u> a <u>tiny change</u> in each pixel of $x$ would make to the classification score for class $c$

# Related Work: Perturbation Methods

- **Key idea:** generate a perturbed dataset to fit an explainable model
  - LIME
  - KernelSHAP



# Related Work: Backpropagation

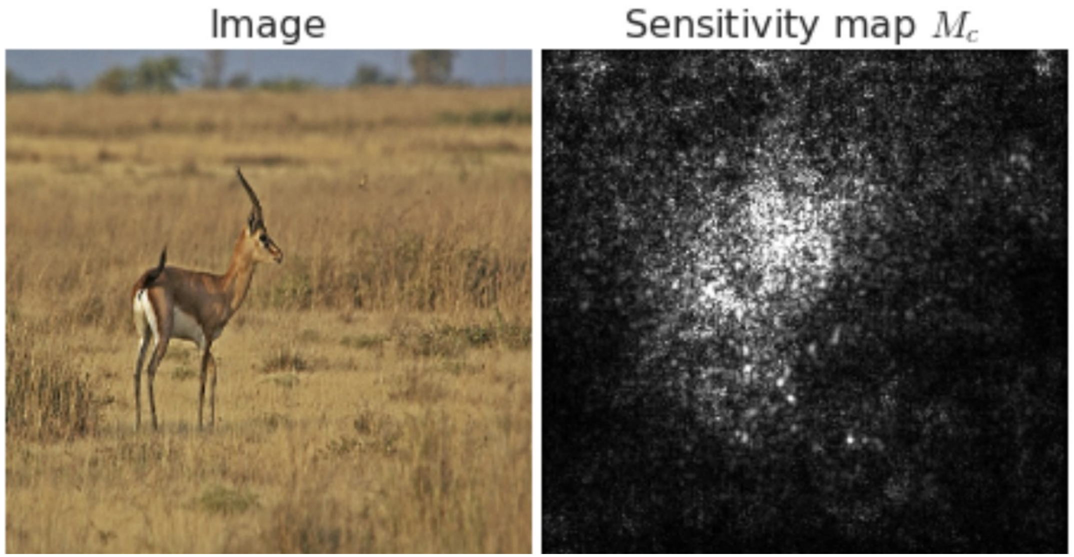



Yellow = inputs

- **Key idea:** backpropagate importance through the network
  - Vanilla gradients
  - Layerwise relevance propagation (Bach et al.)
  - Integrated gradients (Sundararajan et al.)
  - DeepLIFT (Shrikumar et al.)
  - Deconvolution (Zeiler & Fergus, 2014)
  - Guided Backpropagation (Springenberg et al, 2014)

# Limitations of Sensitivity Maps

- **Visually noisy**
  - Often highlight pixels that–to a human eye–seem randomly selected
  - *a priori*, we cannot know if this noise reflects an underlying truth about how networks perform classification, or is due to more superficial fa
    - The Smoo n - we'll get to this soon!

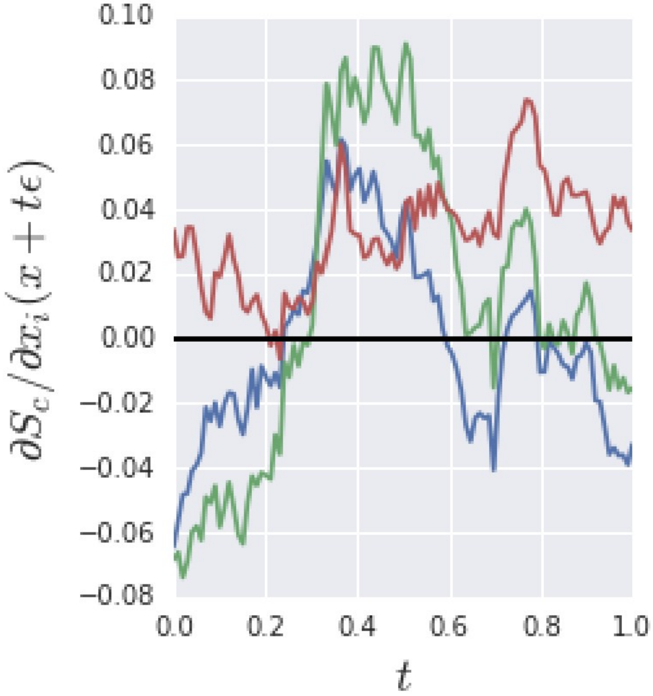


# Theory Behind SmoothGrad: Noisy Gradients

- **Key idea behind SmoothGrad:** noisy maps are due to noisy gradients

- Derivative of $S_c$ may fluctuate sharply at small scales
  - Apparent noise one sees in a sensitivity map may be due to essentially meaningless local variations in partial derivatives

# Noisy Gradients (cont'd)




- Given these rapid fluctuations, gradient of $S_c$ at any given point will be less meaningful than a local average of gradient values.

# SmoothGrad: Intuition

- Recall that **noisy maps are due to noisy gradients**

- **Simple solution**:
  - take an image of interest
  - sample similar images by adding Gaussian noise to the image
  - take the average of the resulting sensitivity maps for each sampled image
    - This <u>smoothes</u> the gradient

# SmoothGrad: Algorithm

1. Take random samples in a neighborhood of an input *x* with added noise
2. Average the resul

$$\hat{M}_c(x) = \frac{1}{n} \sum_{1}^{n} M_c(x + \mathcal{N}(0, \sigma^2))$$

$n$ is the number of samples, and $N(0, \sigma^2)$ represents Gaussian noise with standard deviation $\sigma$.

# Experimental Setup

- Performed SmoothGrad on visualizations of two neural networks:
  - Inception v3 model by Google that was trained on the ILSVRC-2013 dataset
  - Convolutional MNIST model based on the TensorFlow tutorial

# Choosing Hyperparameters ($\sigma$: std. dev.)

$$\hat{M}_c(x) = \frac{1}{n}\sum_{1}^{n} M_c(x + \mathcal{N}(0, \sigma^2))$$

$\sigma$ : the standard deviation of the Gaussian noise

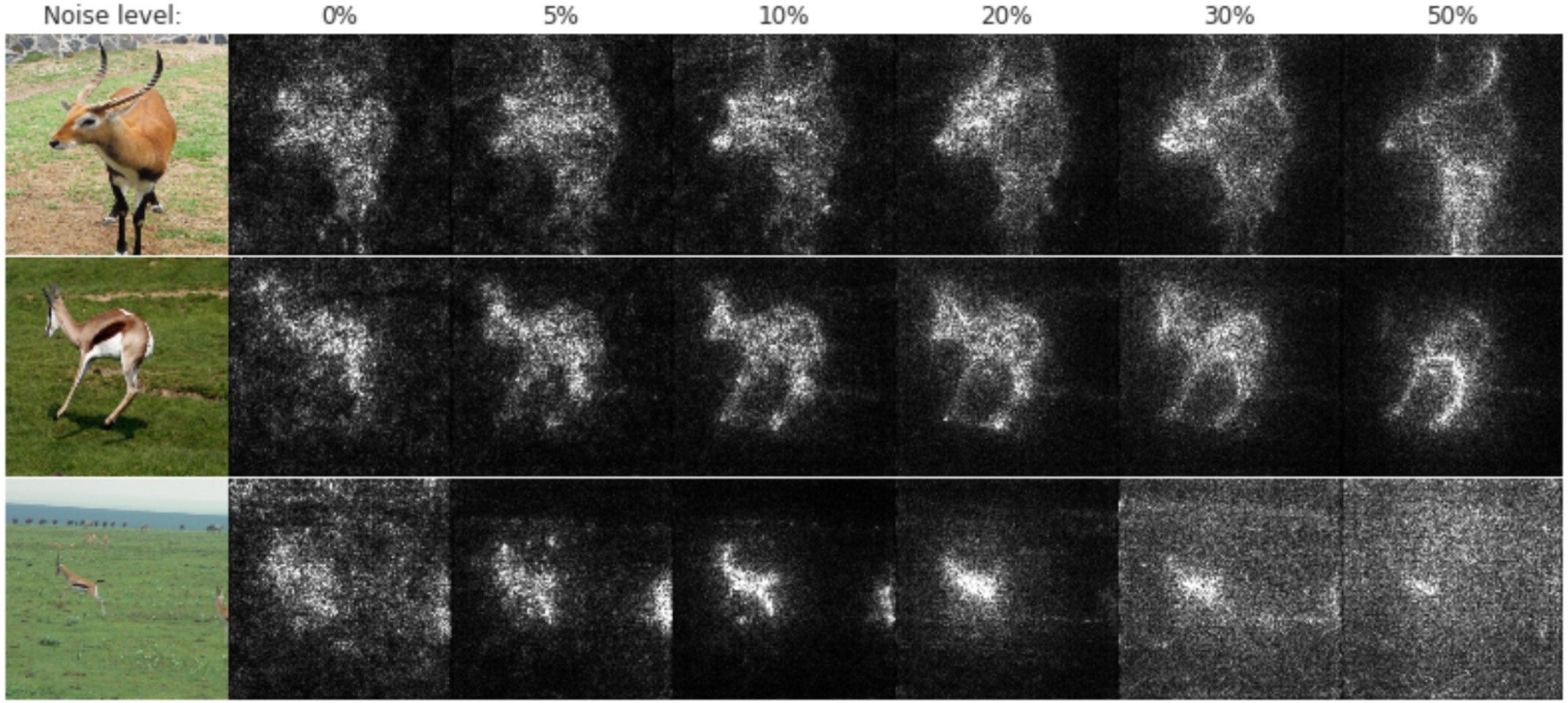



*Figure 3.* Effect of noise level (columns) on our method for 5 images of the gazelle class in ImageNet (rows). Each sensitivity map is obtained by applying Gaussian noise $\mathcal{N}(0, \sigma^2)$ to the input pixels for 50 samples, and averaging them. The noise level corresponds to $\sigma/(x_{max} - x_{min})$.

# Choosing Hyperparameters (*n*: sample size)

$$\hat{M}_c(x) = \frac{1}{n}\sum_{1}^{n} M_c(x + \mathcal{N}(0, \sigma^2))$$

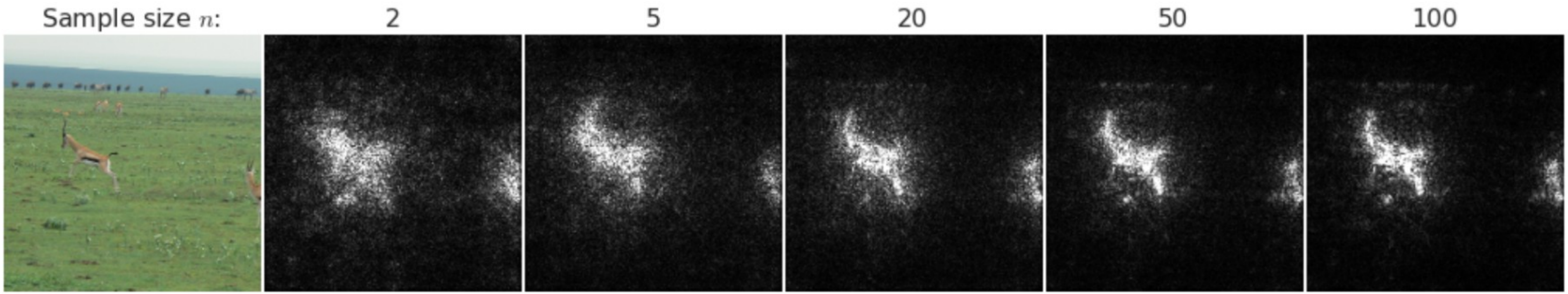
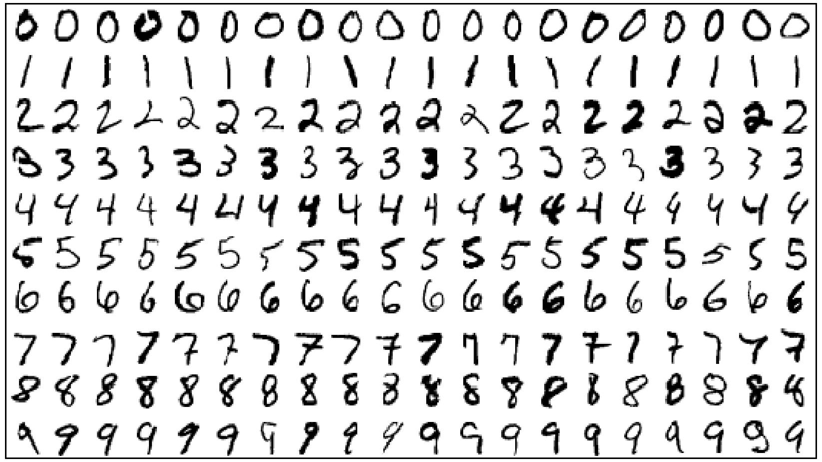
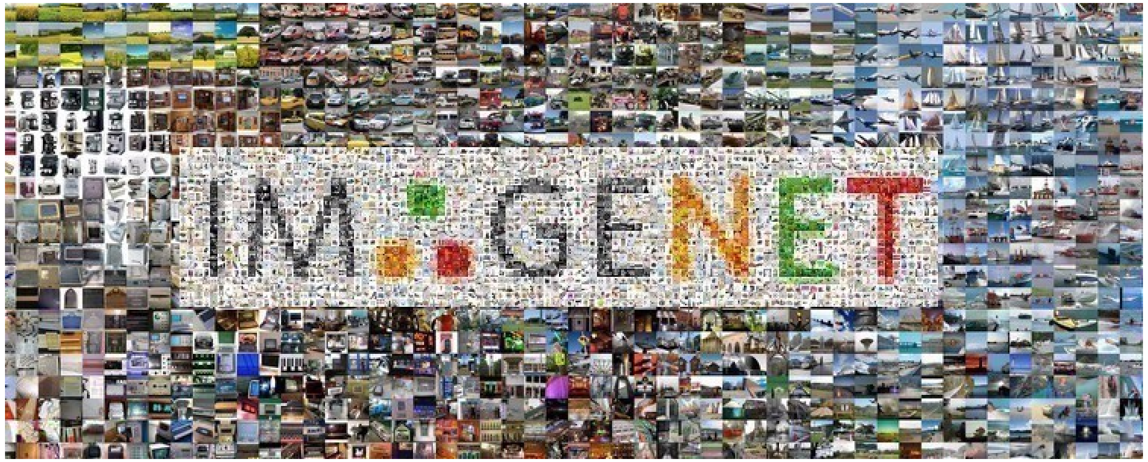



*Figure 4.* Effect of sample size on the estimated gradient for inception. 10% noise was applied to each image.

# Qualitative Results: Visualization Techniques

- **Absolute Value of Gradients**
  - depends on the characteristics of dataset





# Qualitative Results: Visualization Techniques

- **Absolute Value of Gradients**
  - depends on the characteristics of dataset
- **Capping outlying values**
  - presence of few pixels that have much higher gradients than the average
  - capping to 99 percentile
- **Multiplying maps with images**
  - produce simpler & sharper images (Shrikumar et al., 2017; Sundararajan et al., 2017)
  - Downside: Pixels with values of 0 will never show up on the sensitivity map.
  - Upside: when viewing the importance of the feature as contribution to the image

# Qualitative Results: Visual Coherence

***Definition*** (Visual Coherence): Highlights are only on the object of interest, not the background

Comparison with three gradient-based methods

- Vanilla gradient
- Integrated Gradients
- Guided BackProp

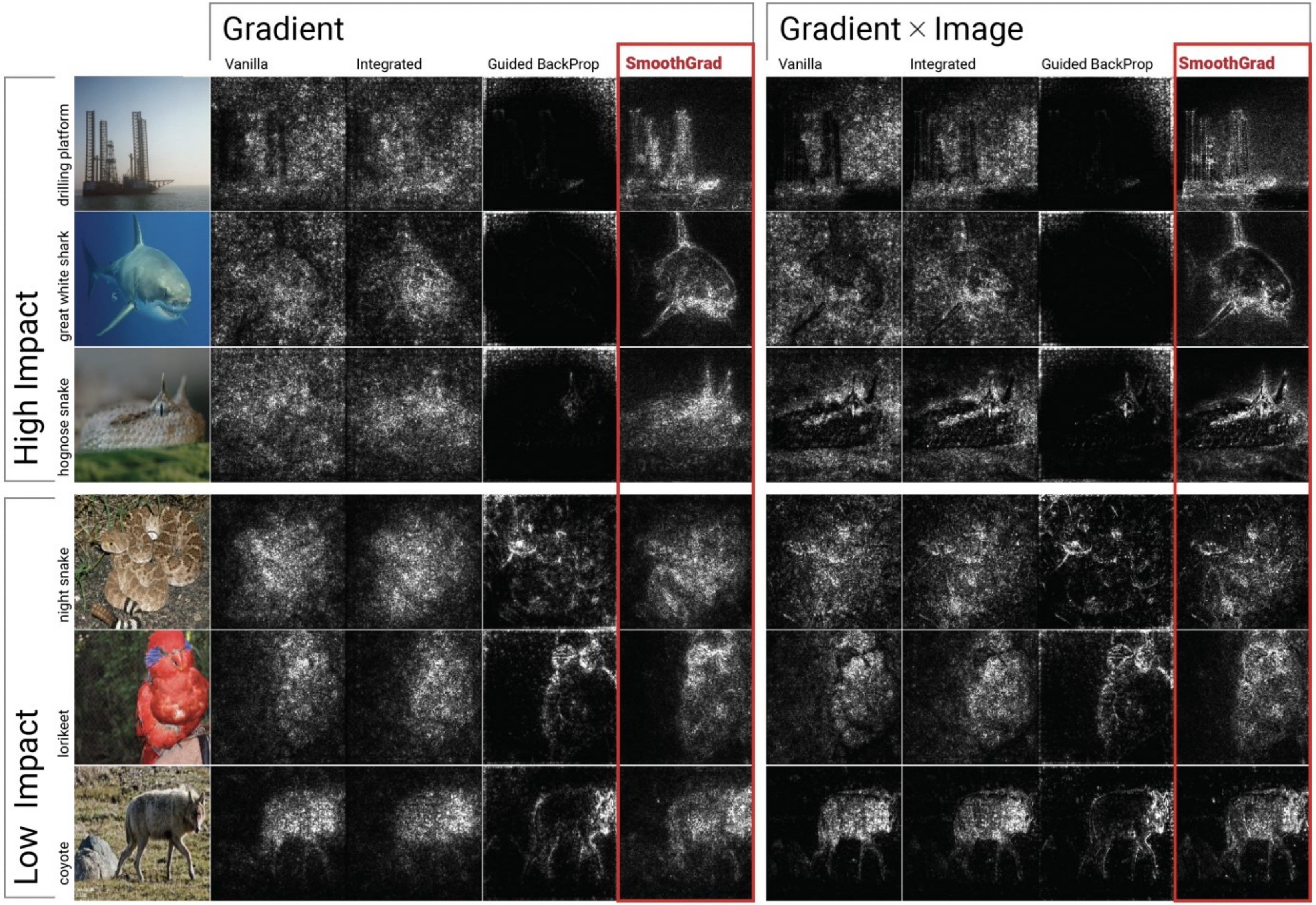



*Figure 5.* Qualitative evaluation of different methods. First three (last three) rows show examples where applying SMOOTHGRAD had high (low) impact on the quality of sensitivity map.

# Qualitative Results: Discriminativity

***Definition*** (Discriminativity): the ability to explain / distinguish separate objects without confusion

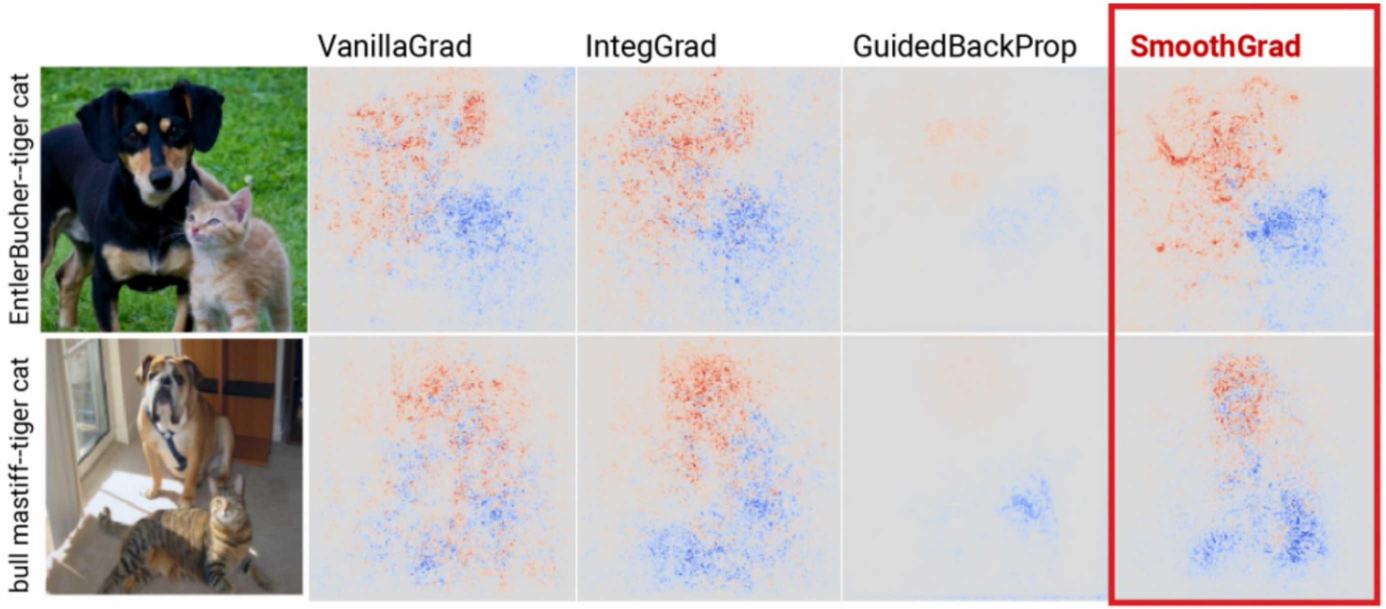



*Figure 6.* Discriminativity of different methods. For each image, we visualize the difference scale($\partial y_1/\partial x$) − scale($\partial y_2/\partial x$) where $y_1$ and $y_2$ are the logits for the first and the second class (i.e., cat or dog) and scale() normalizes the gradient values to be between $[0, 1]$. The values are plotted using a diverging color map $[-1, 0, 1] \mapsto$ [blue, gray, red]. Each method is represented in columns.

# Qualitative Results: Discriminativity

***Open Problem***

Which properties affect the discriminativity of a given methods?

- Why did GBP show the worst performance?

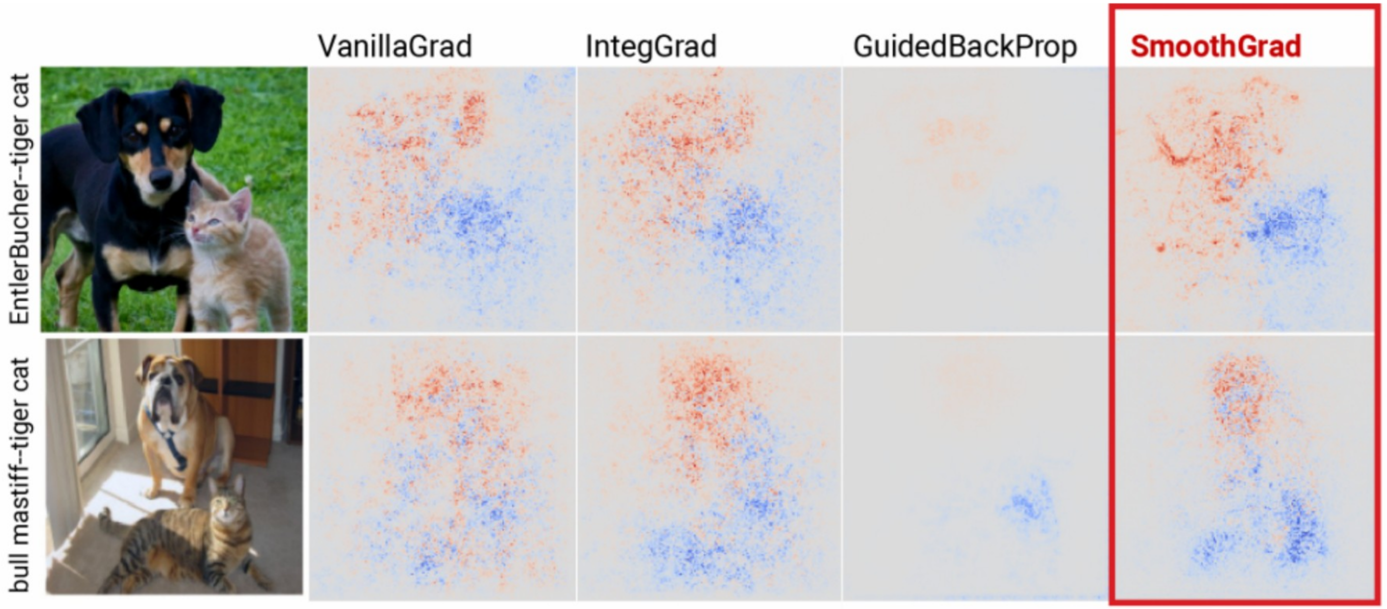
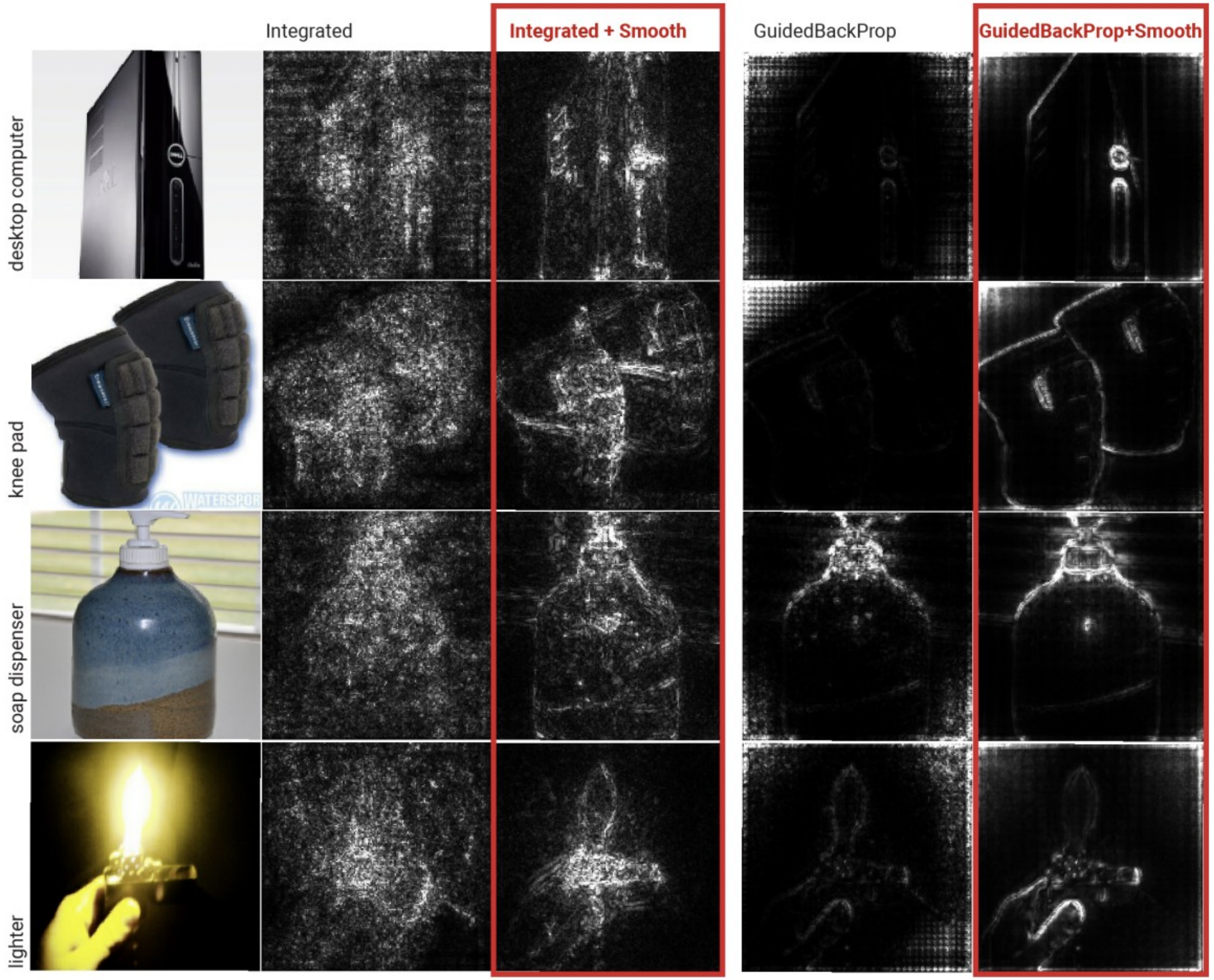



*Figure 6.* Discriminativity of different methods. For each image, we visualize the difference scale$(\partial y_1/\partial x)$ − scale$(\partial y_2/\partial x)$ where $y_1$ and $y_2$ are the logits for the first and the second class (i.e., cat or dog) and scale() normalizes the gradient values to be between $[0, 1]$. The values are plotted using a diverging color map $[-1, 0, 1] \mapsto$ [blue, gray, red]. Each method is represented in columns.

# Combining with Other Methods

The same smoothing procedure can be used to augment any gradient-based method.



# Limitations / Discussion

- Completely qualitative results, can we get **quantitative metrics**?
- Noisy sensitivity maps are **due to noisy gradients**
  - Is this true?
  - Future work: look for further evidence and theoretical arguments
- Does SmoothGrad **generalize** to other networks & tasks?
- How do we tradeoff between making picture pretty and being faithful to the model? Do you think SmoothGrad handled this tradeoff well?

# Thank you!

## Questions?