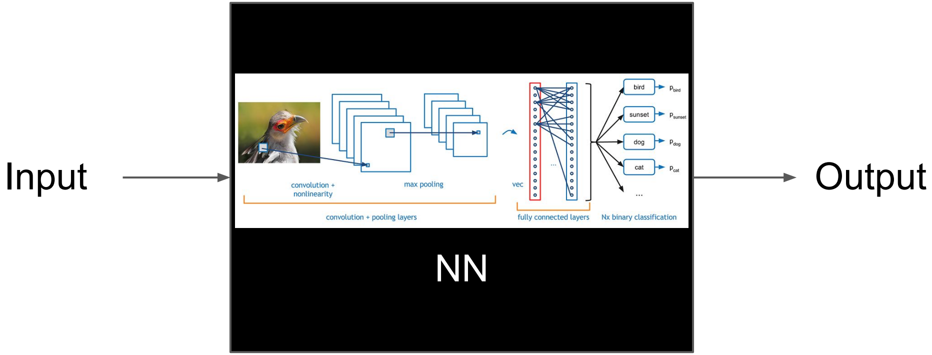
# Interpretability of Deep Neural Networks

Neo Christopher Chung

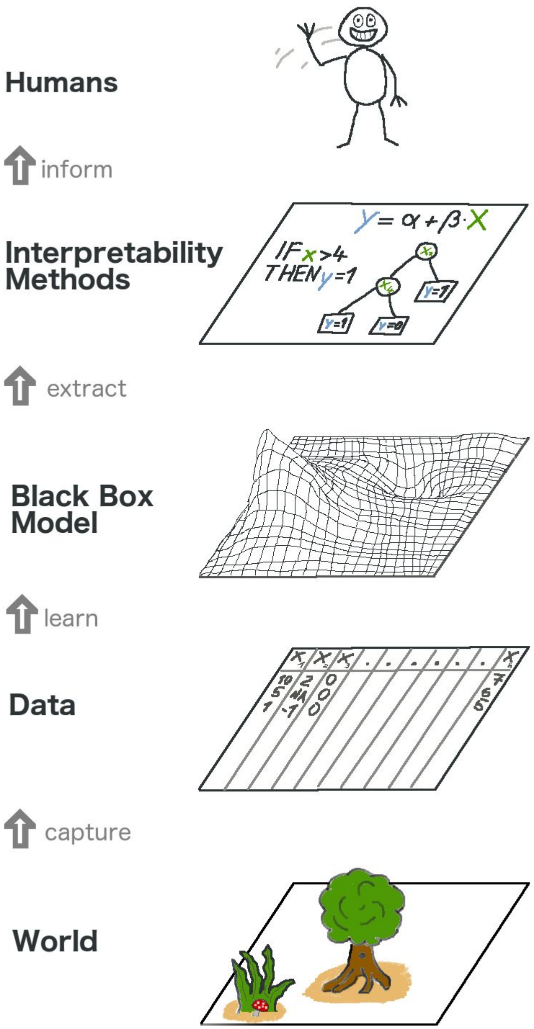
Lecture 11, 1000-719bMSB

# Interpretability of DNN



- Neural Networks are essentially a *black box.*
- Too many parameters, too much non-linearity, etc,
- It is not clear *why* they make a certain classification.

- Interpretability and explainability are the biggest challenge in adopting DNNs in life sciences
- In biology, we often want to know mechanistic properties (not simply prediction)
- In medicine, doctors want to know why a certain recommendation is made by an algorithm



Humans

inform

Interpretability Methods

extract

Black Box Model

learn

Data

capture

World

Interpretable Machine Learning (Molnar)

# Two general approaches to interpretability

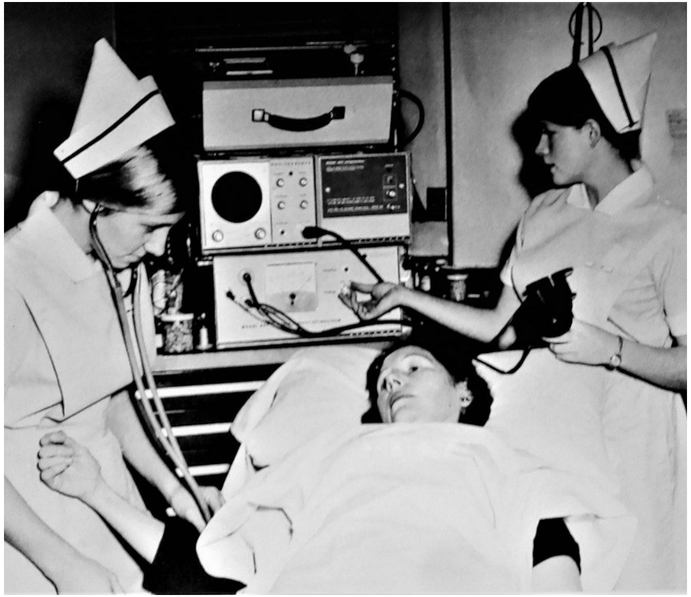
Interpretability of ML/DL can feel enigmatic. Where do we start?

**Model-centric:** explain how the model works in a “simplified” manner while being faithful to the model

**Human-centric:** explain the model works in a “understandable” manner to humans

→ In the best case scenario, both would ideally converge to the same solution

# Cardiovascular diseases, back in the days



# Prognosis of a heart attack

Cardiovascular diseases the leading cause of death

Fatality rates from heart attacks **were** extremely high.

In 1980s, when a heart attack patient is admitted (University of California, San Diego Medical Center), they would measure **19 variables** within 24 hours:

Blood pressure, age, and 17 clinical variables known to be **highly informative** of the patient's condition.

Additionally, temperature, humidity, upper atmospheric conditions, levels of airborne pollutants, and other meteorological variables.

But they were not being used in clinical practices. How to improve the prognosis?

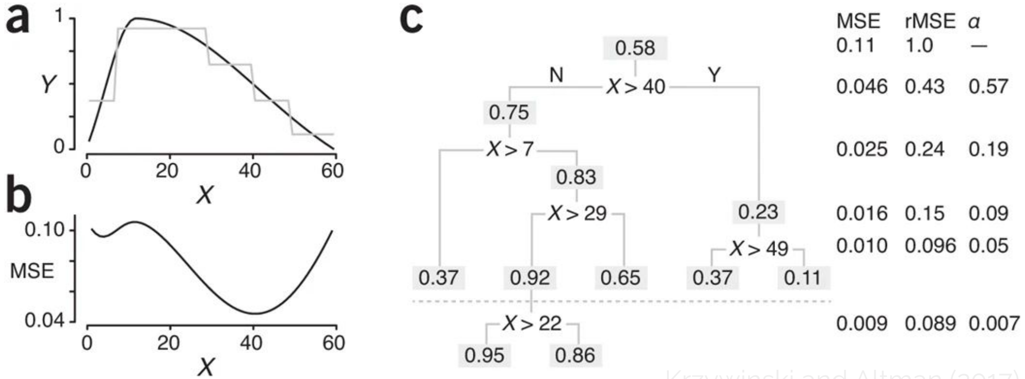
# CART™

Breiman, Friedman, Olshen, Stone developed Classification and Regression Trees (CART).

Select a clinical variable, and split (e.g., binary).
No stopping rule, repeat until no more split is possible.
Minimizing a cost function, a greedy algorithm.

Decision trees allowed clinicians to trust the model and apply even without a computer.

Nowadays, more than 90% survival.



Krzywinski and Altman (2017)

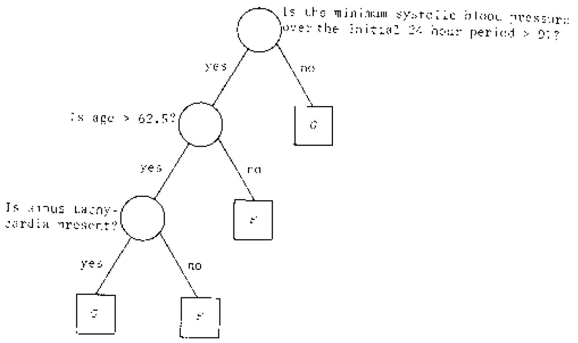

Is the minimum systolic blood pressure over the initial 24 hour period > 91?
yes
no
G
Is age > 62.5?
yes
no
F
Is sinus tachycardia present?
yes
no
G
F

FIGURE 1.1

Breiman et al. (1984)

# What is interpretability?

“We define interpretable machine learning as the extraction of relevant **knowledge** from a machine-learning model concerning relationships either contained in data or learned by the model.” – Murdoch et al. (2019)

“Interpretability is the degree to which a human can **understand the cause of a decision**” – Miller (2017)

“The higher the interpretability of a machine learning model, the easier it is for someone to comprehend **why certain decisions or predictions have been made**.” – Molnar (2022)

# Why it’s so difficult to define interpretability

What is an explanation?

Or a sufficient explanation?

What is understandable to humans?

What if an explanation is understandable to a doctor but not a patient?

How simple should an interpretable model be?

Is a simple model always more interpretable?

How do we compare explanations?

→ **No definition and no quantification**

# Interpretation in a larger context

**Model-based (inherent) interpretability**

Requires modification of existing models

Potentially lower performances

Direct understanding

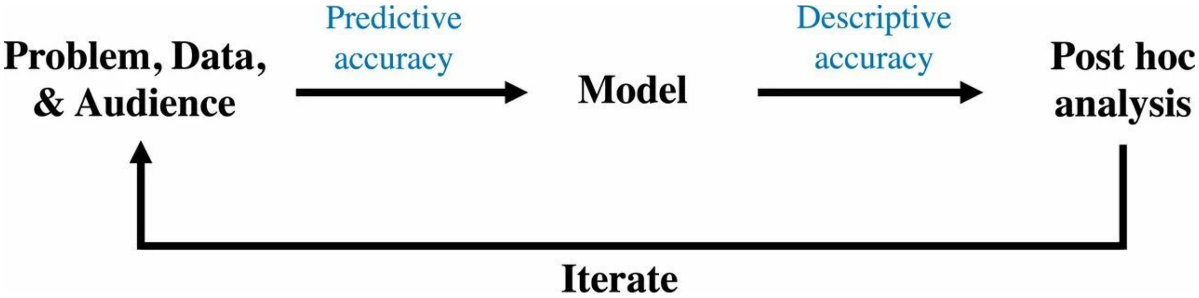
Simpler models/systems

**Post-hoc interpretability**

No modification of a model

No change in performance

Potentially ambiguous interpretation

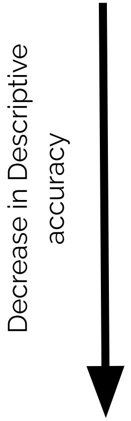


Murdoch et al. (2019)

# Trade-off

Predictive accuracy: the performance of the trained ML model

Descriptive accuracy: the accuracy of the post-hoc interpretability


Decrease in Descriptive accuracy

The full model (e.g., coefficients and weights)
Approximation
The top predictors
Examples/prototypes
Linear local approximation
Model compression

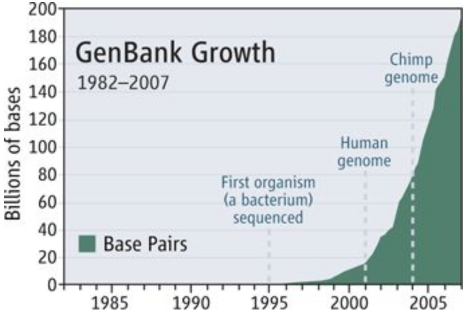
# Why do we make a trade-off?

Increasing data!

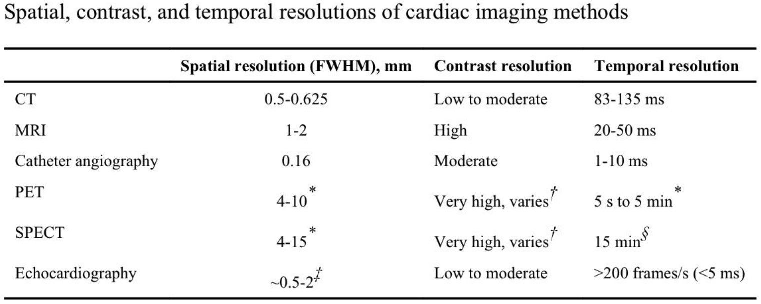
Increasing parameter space
e.g., more genes and pixels

→

Fuller understanding &
Less interpretable



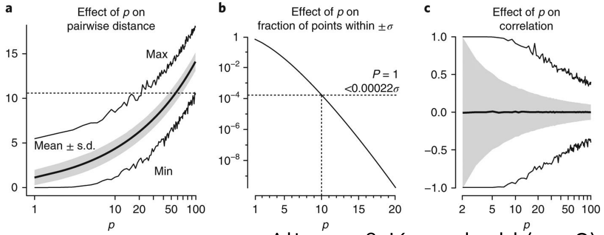
Pennisi 2008



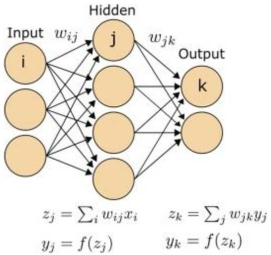
Spatial, contrast, and temporal resolutions of cardiac imaging methods

| | Spatial resolution (FWHM), mm | Contrast resolution | Temporal resolution |
|---|---|---|---|
| CT | 0.5-0.625 | Low to moderate | 83-135 ms |
| MRI | 1-2 | High | 20-50 ms |
| Catheter angiography | 0.16 | Moderate | 1-10 ms |
| PET | 4-10* | Very high, varies† | 5 s to 5 min* |
| SPECT | 4-15* | Very high, varies† | 15 min§ |
| Echocardiography | ~0.5-2‡ | Low to moderate | >200 frames/s (<5 ms) |

Lin and Alessio 2009



Altman & Krzywinski (2018)



More complex models, non-linearity
e.g., GAM, DNN

→

Better performance &
Less interpretable

# Machine Learning and Interpretability

Simpler models/algorithms
Easier to interpret

→

More complex models/algorithms
Harder to interpret

Linear Model

Kernel Methods

Generalized Linear Models

Generalized Additive Models

Decision trees

Rules-based

Bagging, Boosting, Ensemble Models

Perceptron

Sparse DL

Convolutional NN

# Large data means noisy data

$$\mathbf{Y} = \mathbf{B}\mathbf{X} + \mathbf{E}$$

Based on observed **X** and **Y**, **B** must be estimated.

In many systems, many variables are expected **not to contribute** to the outcome.

e.g., background pixels in CT/PET images unimportant for tumor/survival/prognosis

e.g., most of genes not related to clinical phenotypes in a genome-wide association study

When data are collected from real world, all of estimated coefficients will be **likely non-zero.**

# Regularization and shrinkage

Lasso adds a $L_1$ penalty to the least squares:

$$\min_{\beta_0, \beta} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - x_i^T \beta)^2 \right\} \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq t.$$

B = {$B_1$, …, $B_p$}; t controls the regularization

Ridge ($L_2$ penalty)

Elastic Net (combining $L_1$ & $L_2$ penalties)

# Lasso example: prostate cancer (Stamey et al. 1989)

Men who were about to receive radical prostatectomy

Levels of **prostate-specific antigen** and clinical variables:
age, cancer vol, , prostate weight, benign prostatic hyperplasia amount, etc

Fit a linear model,
with a lasso penalty

Shrinkage effect (t) was
selected by cross validation

Three clinical variables are
selected

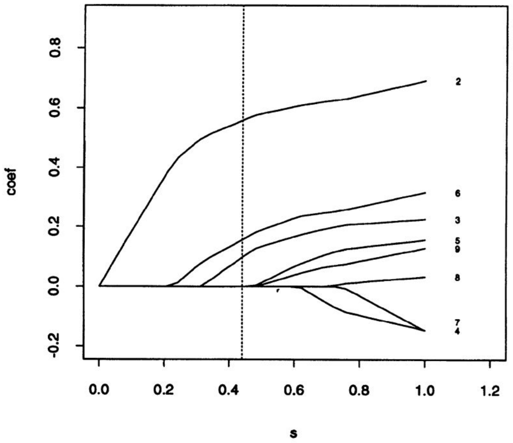


Fig. 5. Lasso shrinkage of coefficients in the prostate cancer example: each curve represents a coefficient (labelled on the right) as a function of the (scaled) lasso parameter $s = t/\Sigma|\beta_j^o|$ (the intercept is not plotted); the broken line represents the model for $\hat{s} = 0.44$, selected by generalized cross-validation

Tibshirani (1996)

# Shrinkage & variable selection

A large number of predictors → variable/feature selection.

Nowadays, a typical clinical studies would collect > hundreds of variables.

How to intelligently regularize is one of the central goals.

In the feature space of DNN:

Srivastava et al. 2014 Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Lemhadri et al 2021 A neural network with feature sparsity

Interpretability methods for neural networks also **implicitly or explicitly**:

Ross et al. 2017 The Neural LASSO: Local Linear Sparsity for Interpretable Explanations

# How to interpret a black box model (e.g., DNNs)

Shrinkage, variable selection, CART, and other aforementioned methods are well applicable for ML/DL

Attempting to make senses out of diverse interpretability methods: **global vs. local explanations**

Interpretability is an active area of research and there are **no consensus on what is the best** or the most accurate approach

Explanations in practice require deep knowledge of application domains and **how `interpretability’ would be used** subsequently

**Post-hoc local explanations**: importance estimators or saliency maps

# Global Explanations

The overall behavior of a model with respect to certain features

Most often, we look at a change in a prediction (probability)

For non-linear models, global interpretability may not be accurate at all points

Depending on the methods, it may hide very important behaviors!

Some approximation is necessary to reasonably reduce the large surface area

# Surrogate Model

Train an (inherently) interpretable model to emulate a blackbox model

Approximate the predictions of the underlying model

Must be as interpretable as possible

Efficient, fast, and affordable computation

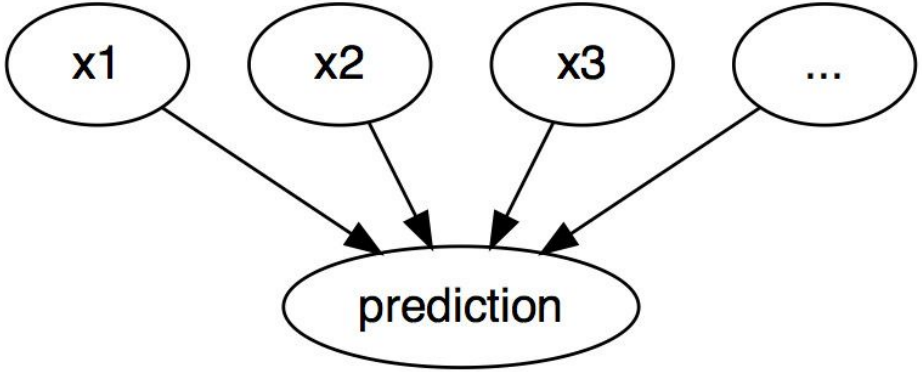
Train the interpretable model using the predictions of a target model

# Local Explanations

How do individual predictions (or probabilities) change with respect to the change in features

Counterfactual logics is fundamental in casualty (or causal inference):

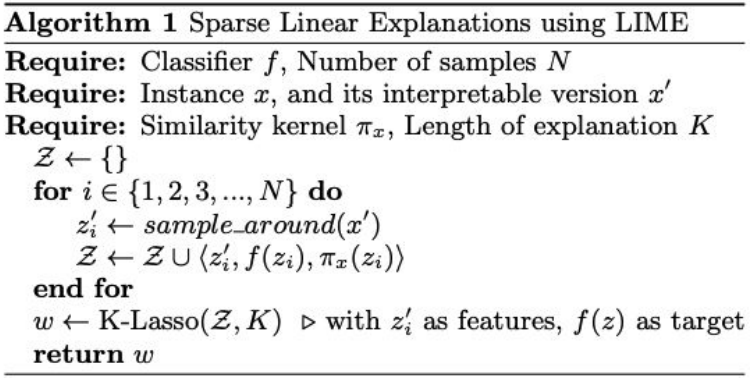
“what would happen to the prediction, if x changes”



# Local interpretable model-agnostic explanations

(LIME) Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin

1. Select a data sample (i.e., observation)
2. Perturb them (multiple times) and obtain the predictions
3. Weight the new samples according to their distances to the target
4. Train a weighted interpretable model (typically, LM with a Lasso) only on the new perturbed samples
5. Explain the prediction by interpreting the "local" model



**Algorithm 1** Sparse Linear Explanations using LIME

```
Require: Classifier $f$, Number of samples $N$
Require: Instance $x$, and its interpretable version $x'$
Require: Similarity kernel $\pi_x$, Length of explanation $K$
  $\mathcal{Z} \leftarrow \{\}$
  for $i \in \{1, 2, 3, ..., N\}$ do
      $z'_i \leftarrow sample\_around(x')$
      $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
  end for
  $w \leftarrow$ K-Lasso($\mathcal{Z}, K$)  ▷ with $z'_i$ as features, $f(z)$ as target
  return $w$
```

# Shapley values (Shapley, 1951) Nobel prize 2012

To each cooperative game it assigns a unique distribution (among the players) of a total surplus generated by the coalition of all players. (Wikipedia)

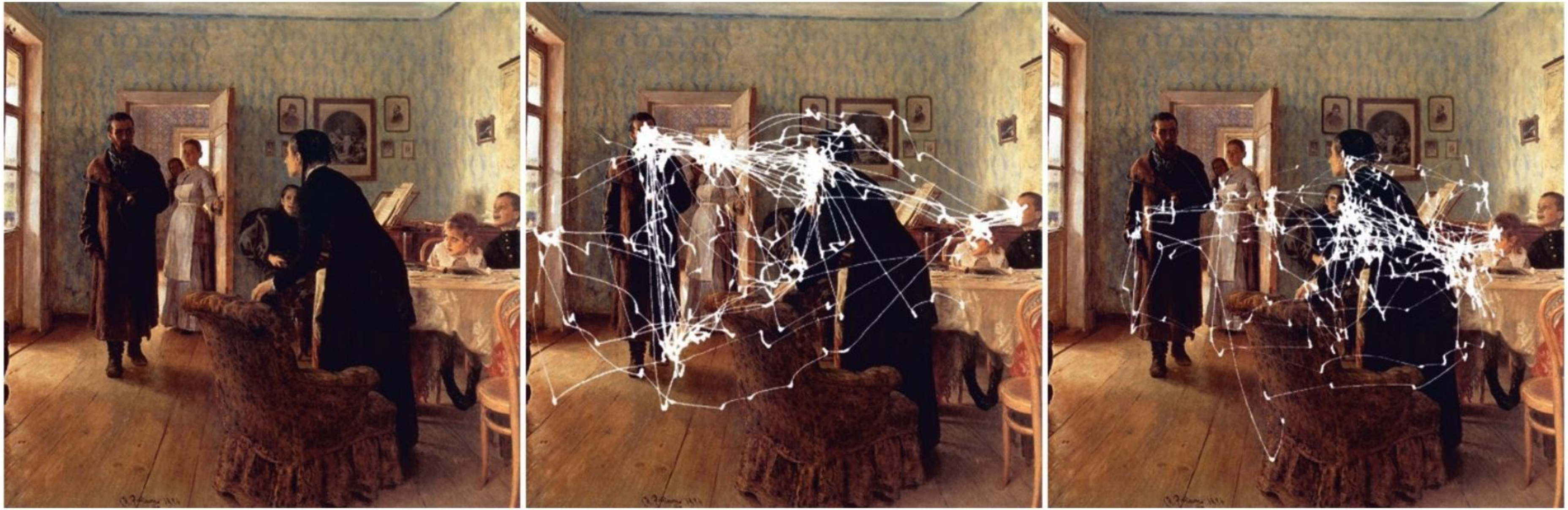
Explain the difference between the mean prediction and the actual prediction (of the model)

Calculate a mean marginal contribution to the prediction across all combinations of features

In modern ML/DL problems, computationally infeasible

SHAP (SHapley Additive exPlanations) Lundberg and Lee (2017)

# Saliency (as general concept)

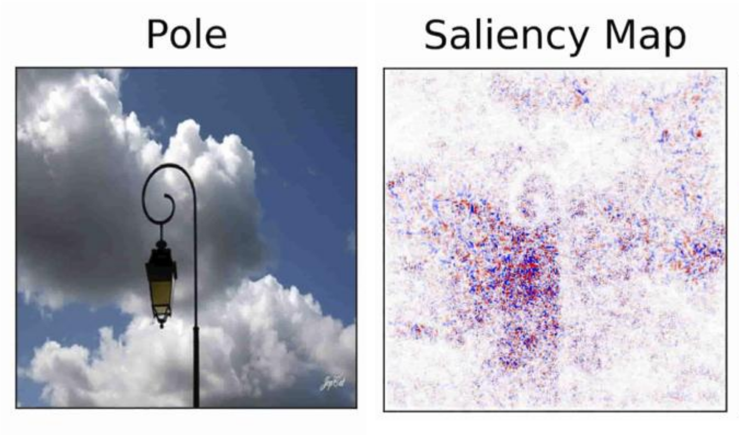


Yarbus, Eye Movements and Vision, 1967

# Saliency Maps aka, feature importance/relevance, pixel attribution

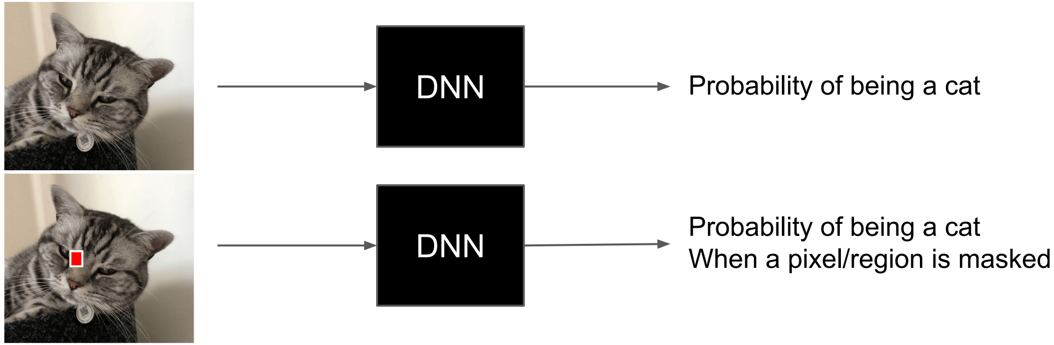
- Where does the model look at?
- We want to know importance of input features (pixels) for classification
- We can visualize ‘importance scores’ in the same dimensions as inputs

DNN trained on ImageNet
Image is classified as “Pole”
What pixels were important?



# Two Types of Saliency Maps

## Perturbation-based forward propagation methods



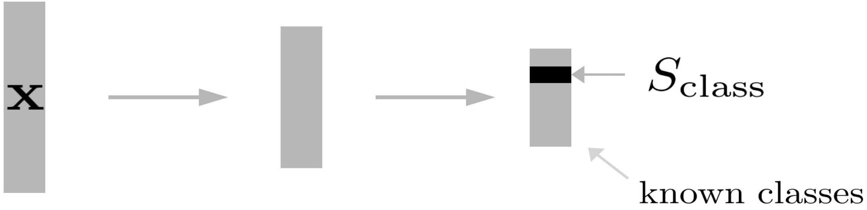
DNN

Probability of being a cat

DNN

Probability of being a cat
When a pixel/region is masked

## Gradient-based backpropagation methods

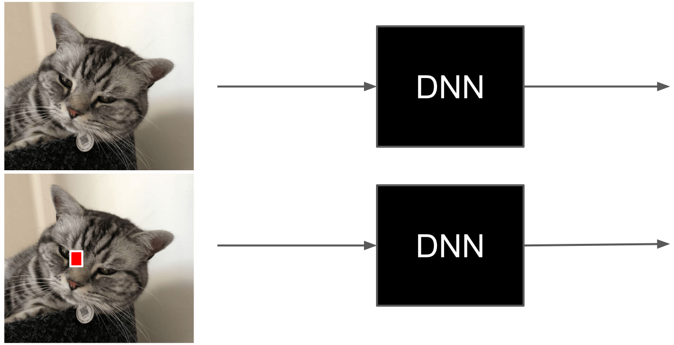


$$S_{ij} = \frac{\partial S_{\text{class}}}{\partial x_{ij}}$$

# Occlusion

Zeiler and Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014

## Perturbation-based forward propagation methods



DNN

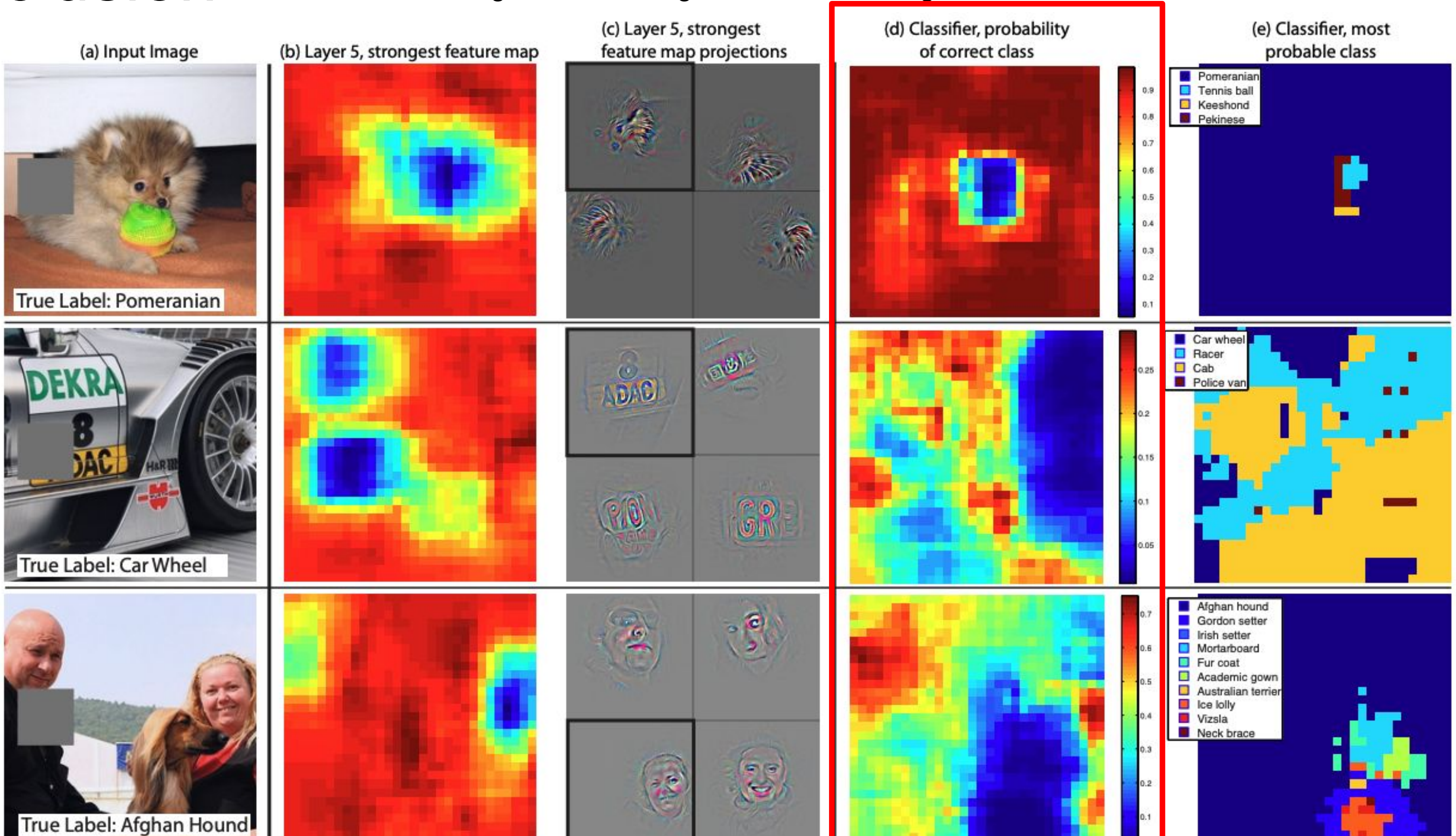
Probability of being a cat

DNN

Probability of being a cat
When a pixel/region is masked

# Occlusion

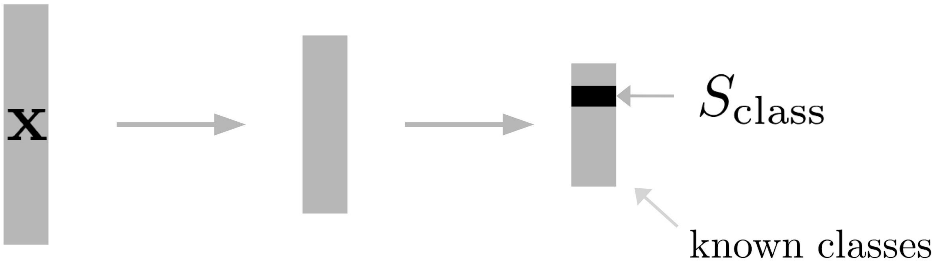
Zeiler and Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014



# Gradient with backpropagation

Simonyan, Vedaldi, and Zisserman, “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”, ICLR Workshop 2014.

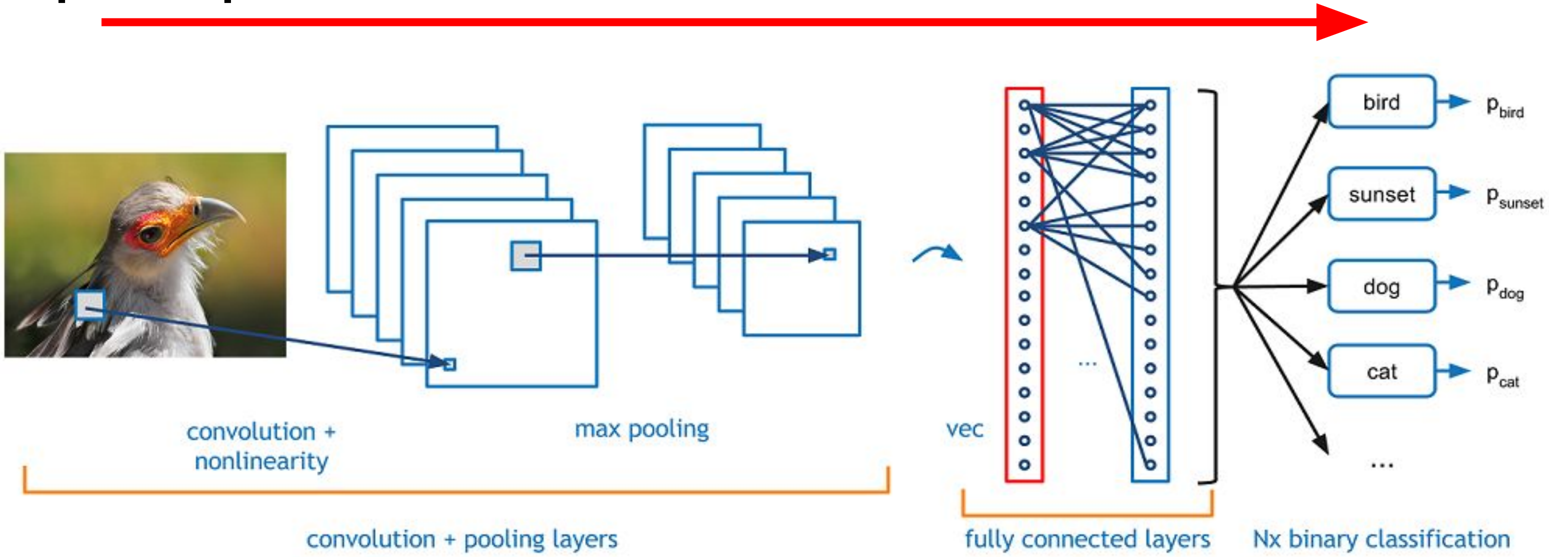
$$S_{ij} = \frac{\partial S_{\text{class}}}{\partial x_{ij}}$$



DNN is non-linear, thus Simonyan et al (2013) propose to estimate it via backpropagation.
Only the class of interest is being used, while all other classes are set to 0.

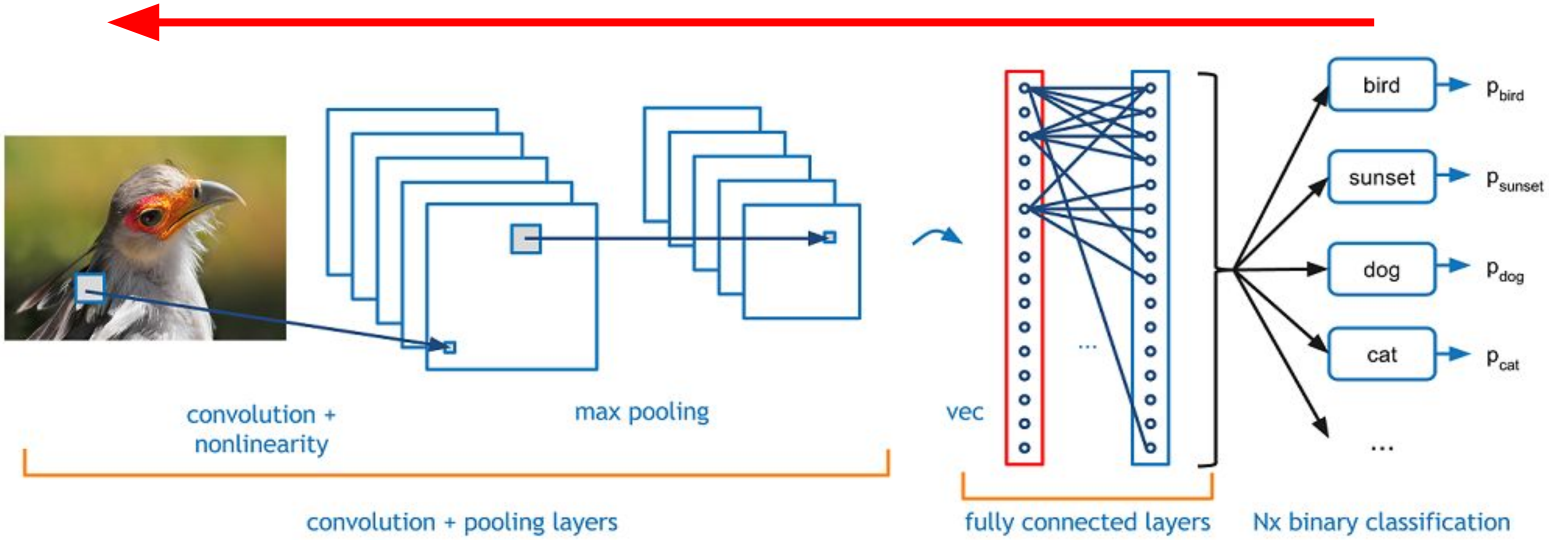
Results could be in 3 channels (RGB), collapsing into 2D
Some uses negative and positive values separately (also called ‘divergent’ in some old libraries)
Most often, people take absolute values

# Forward propagation
# Compute $p_{bird}$



# Backpropagation

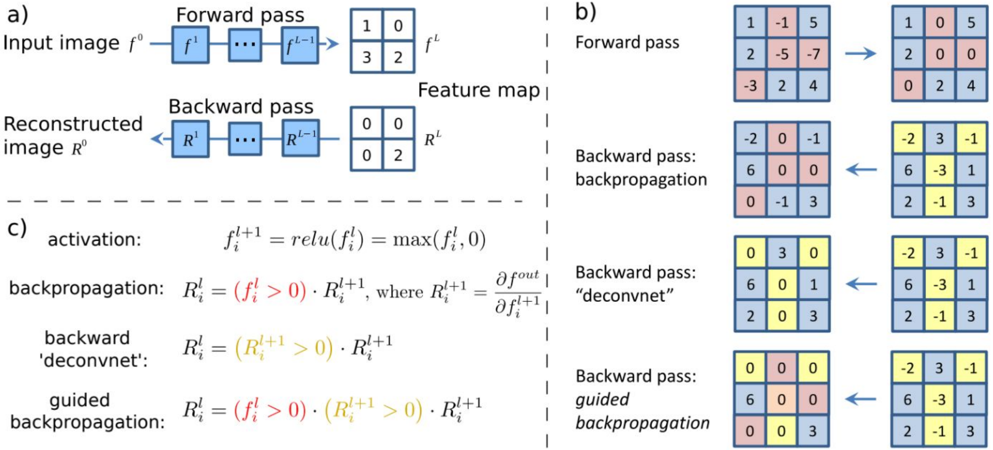
# Compute gradient of class score w.r.t. Image pixels



# Guided Backpropagation

Springenberg et al. (2014)

Emphasize positive contributions to the final outcome.



$$f^L = \begin{bmatrix} 1 & 0 \\ 3 & 2 \end{bmatrix} \qquad R^L = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}$$

c)

activation: $f_i^{l+1} = relu(f_i^l) = \max(f_i^l, 0)$

backpropagation: $R_i^l = (f_i^l > 0) \cdot R_i^{l+1}$, where $R_i^{l+1} = \frac{\partial f^{out}}{\partial f_i^{l+1}}$

backward 'deconvnet': $R_i^l = (R_i^{l+1} > 0) \cdot R_i^{l+1}$

guided backpropagation: $R_i^l = (f_i^l > 0) \cdot (R_i^{l+1} > 0) \cdot R_i^{l+1}$

b)

Forward pass
$$\begin{bmatrix} 1 & -1 & 5 \\ 2 & -5 & -7 \\ -3 & 2 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 5 \\ 2 & 0 & 0 \\ 0 & 2 & 4 \end{bmatrix}$$

Backward pass: backpropagation
$$\begin{bmatrix} -2 & 0 & -1 \\ 6 & 0 & 0 \\ 0 & -1 & 3 \end{bmatrix} \leftarrow \begin{bmatrix} -2 & 3 & -1 \\ 6 & -3 & 1 \\ 2 & -1 & 3 \end{bmatrix}$$

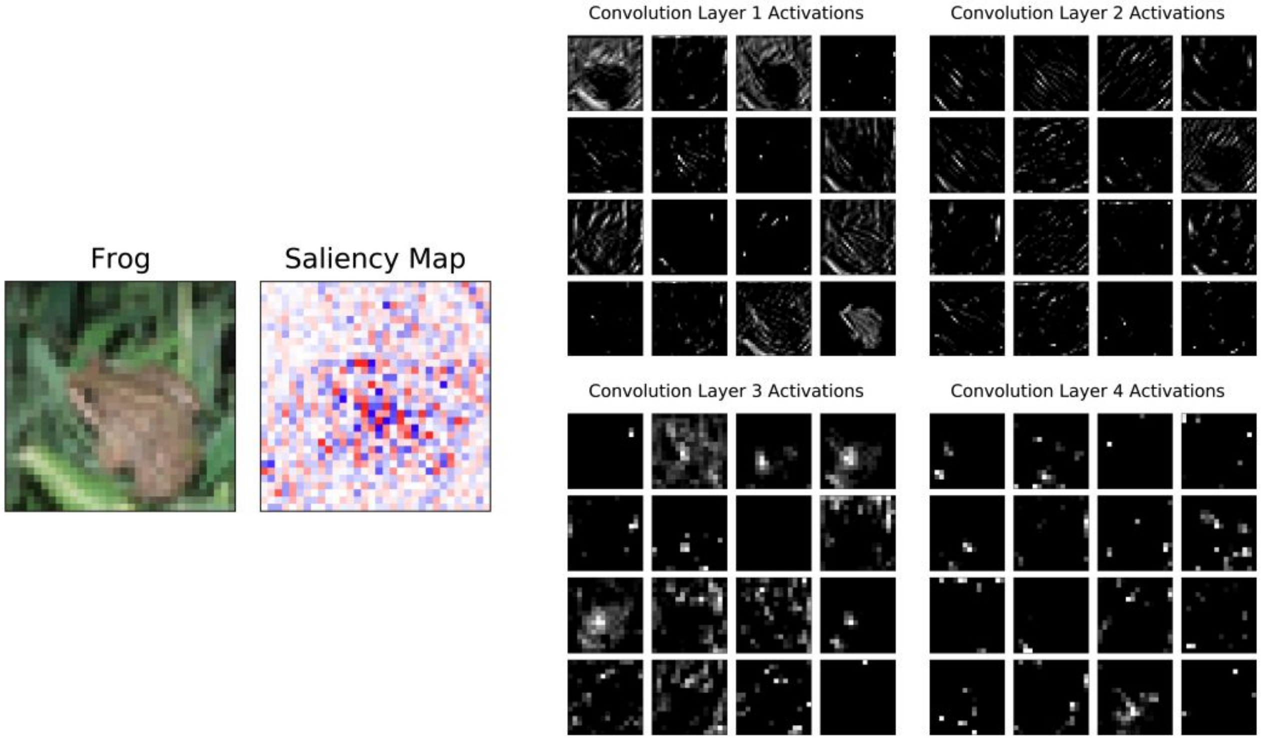
Backward pass: “deconvnet”
$$\begin{bmatrix} 0 & 3 & 0 \\ 6 & 0 & 1 \\ 2 & 0 & 3 \end{bmatrix} \leftarrow \begin{bmatrix} -2 & 3 & -1 \\ 6 & -3 & 1 \\ 2 & -1 & 3 \end{bmatrix}$$

Backward pass: *guided backpropagation*
$$\begin{bmatrix} 0 & 0 & 0 \\ 6 & 0 & 0 \\ 0 & 0 & 3 \end{bmatrix} \leftarrow \begin{bmatrix} -2 & 3 & -1 \\ 6 & -3 & 1 \\ 2 & -1 & 3 \end{bmatrix}$$

Figure 1: Schematic of visualizing the activations of high layer neurons. a) Given an input image, we perform the forward pass to the layer we are interested in, then set to zero all activations except one and propagate back to the image to get a reconstruction. b) Different methods of propagating back through a ReLU nonlinearity. c) Formal definition of different methods for propagating a output activation *out* back through a ReLU unit in layer $l$; note that the ’deconvnet’ approach and guided backpropagation do not compute a true gradient but rather an imputed version.

# Rectified Gradients

Kim et al. (2020)


Frog
Saliency Map
Convolution Layer 1 Activations
Convolution Layer 2 Activations
Convolution Layer 3 Activations
Convolution Layer 4 Activations

(a) Sample image and its saliency map.

(b) Intermediate layer activations.

Figure 2: Feature map visualization for an image with a noisy saliency map.

# Rectified Gradients

Kim et al. (2020) introduces an arbitrary thresholding to Guided Backpropagagtion

Instead of thresholding at 0, we introduce τ

guided backpropagation: $R_i^l = (f_i^l > 0) \cdot (R_i^{l+1} > 0) \cdot R_i^{l+1}$

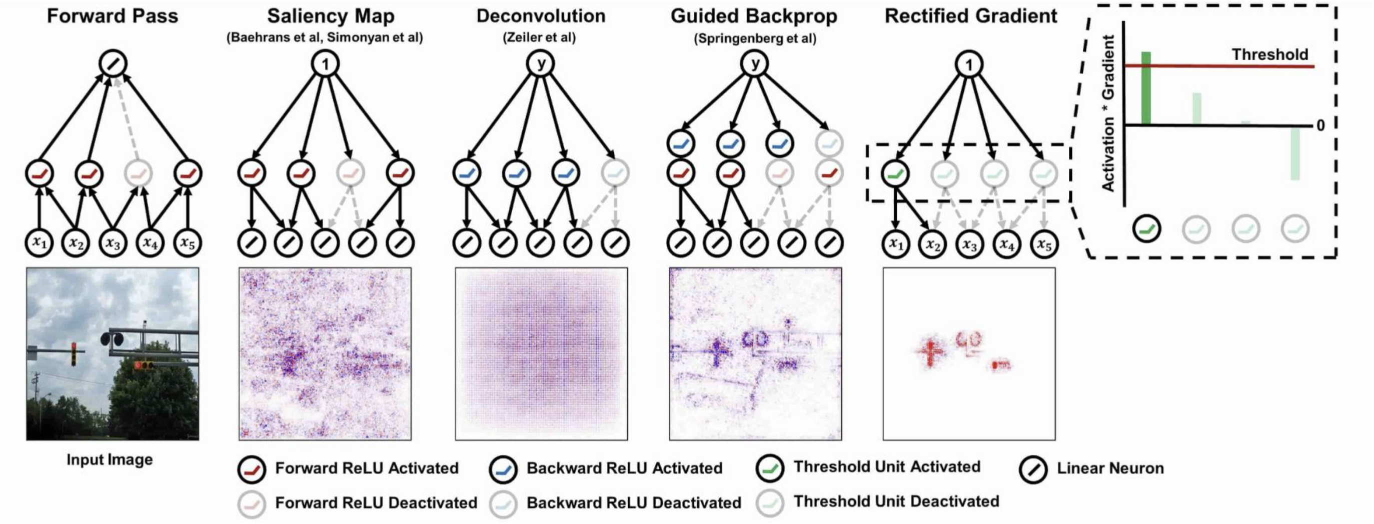
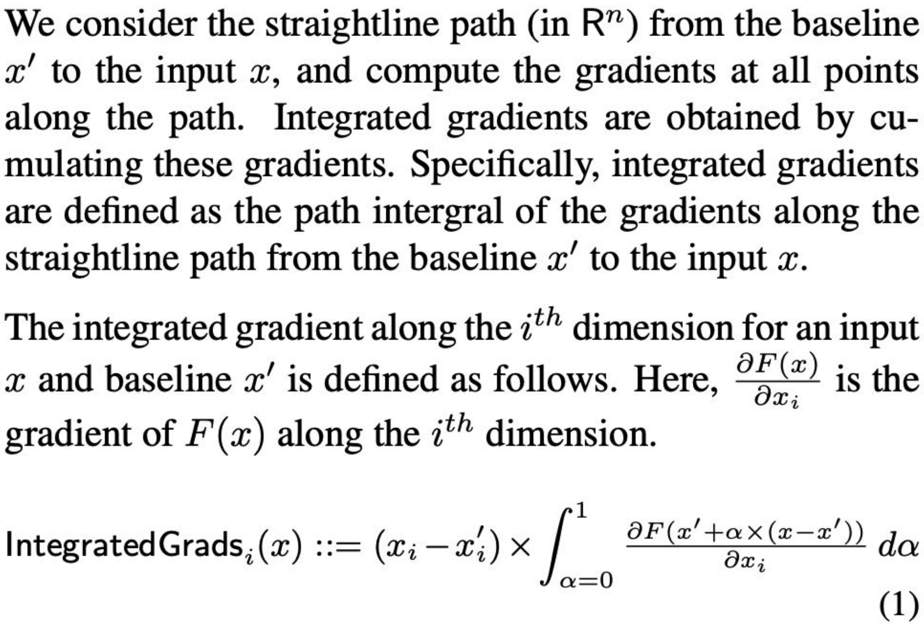


Figure 1: Comparison of attribution methods. See Appendix F.1 for details on the visualization.

# Integrated Gradients

Sundararajan et al. (2017)



We consider the straightline path (in R$^n$) from the baseline $x'$ to the input $x$, and compute the gradients at all points along the path. Integrated gradients are obtained by cumulating these gradients. Specifically, integrated gradients are defined as the path intergral of the gradients along the straightline path from the baseline $x'$ to the input $x$.

The integrated gradient along the $i^{th}$ dimension for an input $x$ and baseline $x'$ is defined as follows. Here, $\frac{\partial F(x)}{\partial x_i}$ is the gradient of $F(x)$ along the $i^{th}$ dimension.

$$\mathsf{IntegratedGrads}_i(x) ::= (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i}\, d\alpha \qquad (1)$$

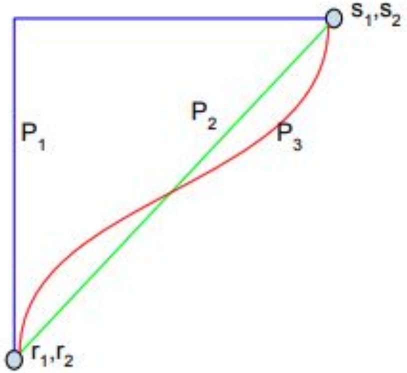


*Figure 1.* Three paths between an a baseline $(r_1, r_2)$ and an input $(s_1, s_2)$. Each path corresponds to a different attribution method. The path $P_2$ corresponds to the path used by integrated gradients.

# Integrated Gradients

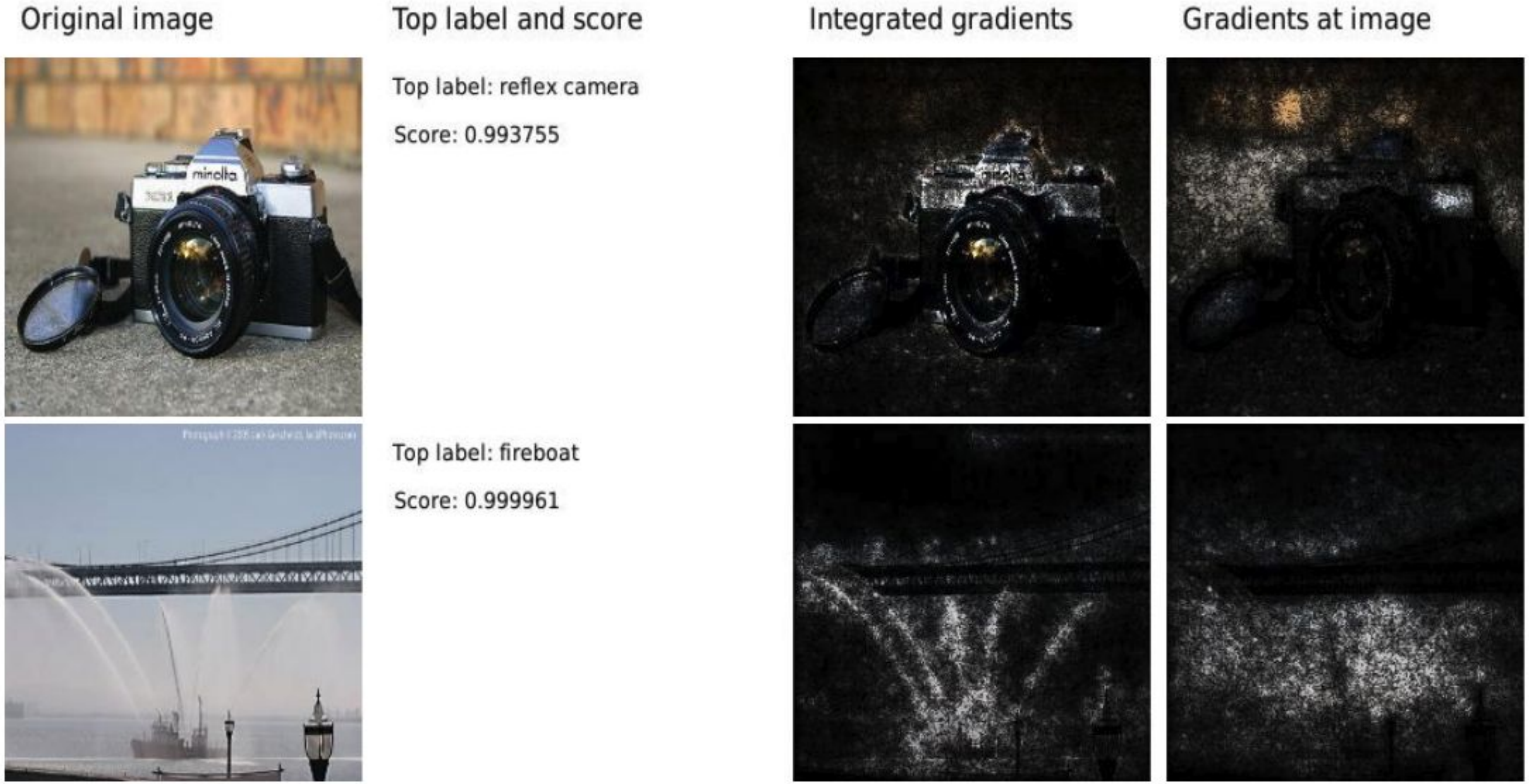
Sundararajan et al. (2017)

1. Obtain a series of images between the original image and the baseline image (e.g., black)
2. Calculate importance scores of those multiple images
3. Calculate average importance scores

$$\phi_i^{IG}(f, x, x') = \overbrace{(x_i - x'_i)}^{\text{Difference from baseline}} \times \underbrace{\int_{\alpha=0}^{1}}_{\text{From baseline to input}\ldots} \overbrace{\frac{\delta f(x' + \alpha(x - x'))}{\delta x_i} d\alpha}^{\ldots\text{accumulate local gradients}}$$

# Integrated Gradients

Sundararajan et al. (2017)



# Smooth Gradients

Smilkov et al. (2017) “SmoothGrad: removing noise by adding noise”

1. Add a slight (i.i.d. Gaussian) noise to the original image, creating multiple versions

2. Calculate importance scores (e.g., vanilla gradients) of those multiple ‘noisy’ images

3. Calculate average importance scores

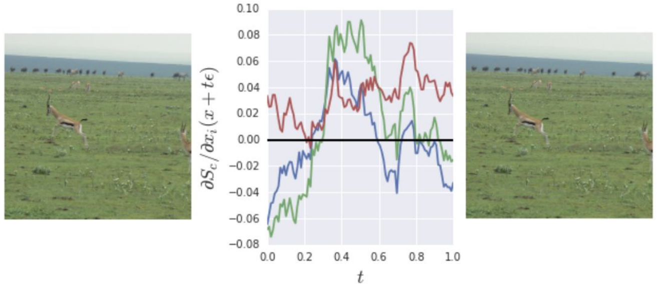
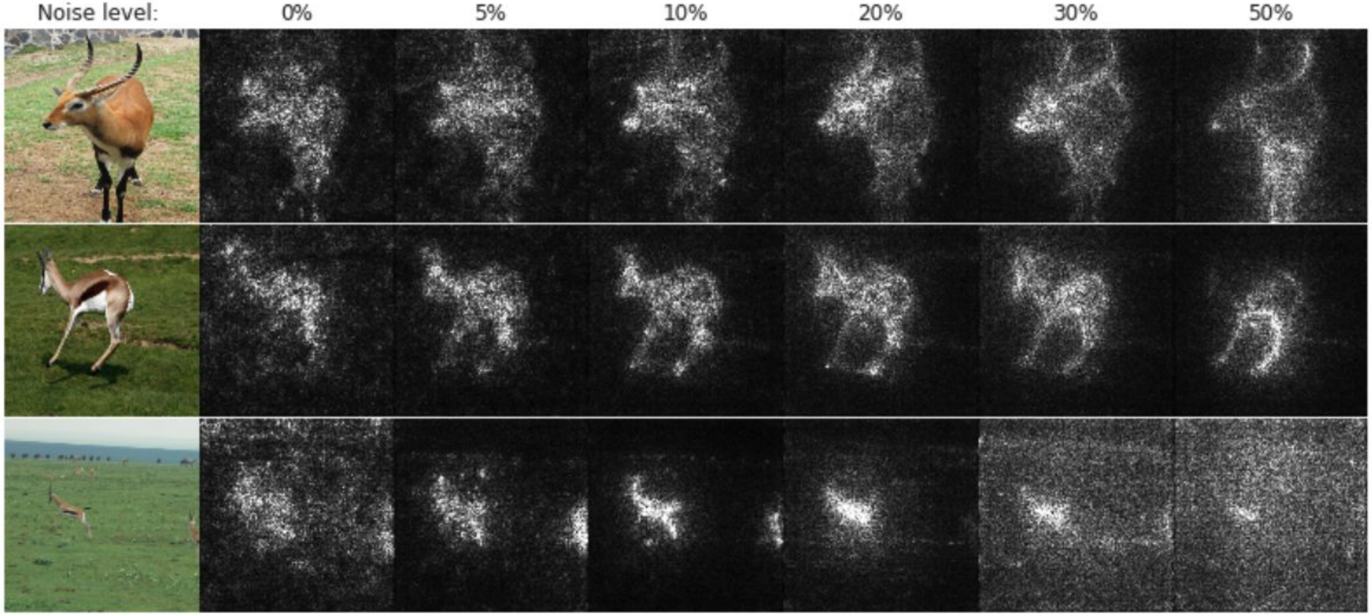


*Figure 2.* The partial derivative of $S_c$ with respect to the RGB values of a single pixel as a fraction of the maximum entry in the gradient vector, $\max_i \frac{\partial S_c}{\partial x_i}(t)$, (middle plot) as one slowly moves away from a baseline image $x$ (left plot) to a fixed location $x+\epsilon$ (right plot). $\epsilon$ is one random sample from $\mathcal{N}(0,\, 0.01^2)$. The final image $(x+\epsilon)$ is indistinguishable to a human from the origin image $x$.

# Smooth Gradients

Smilkov et al. (2017) “SmoothGrad: removing noise by adding noise”



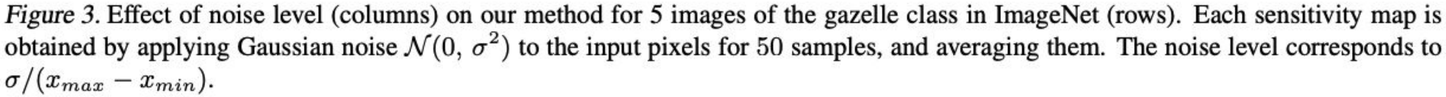
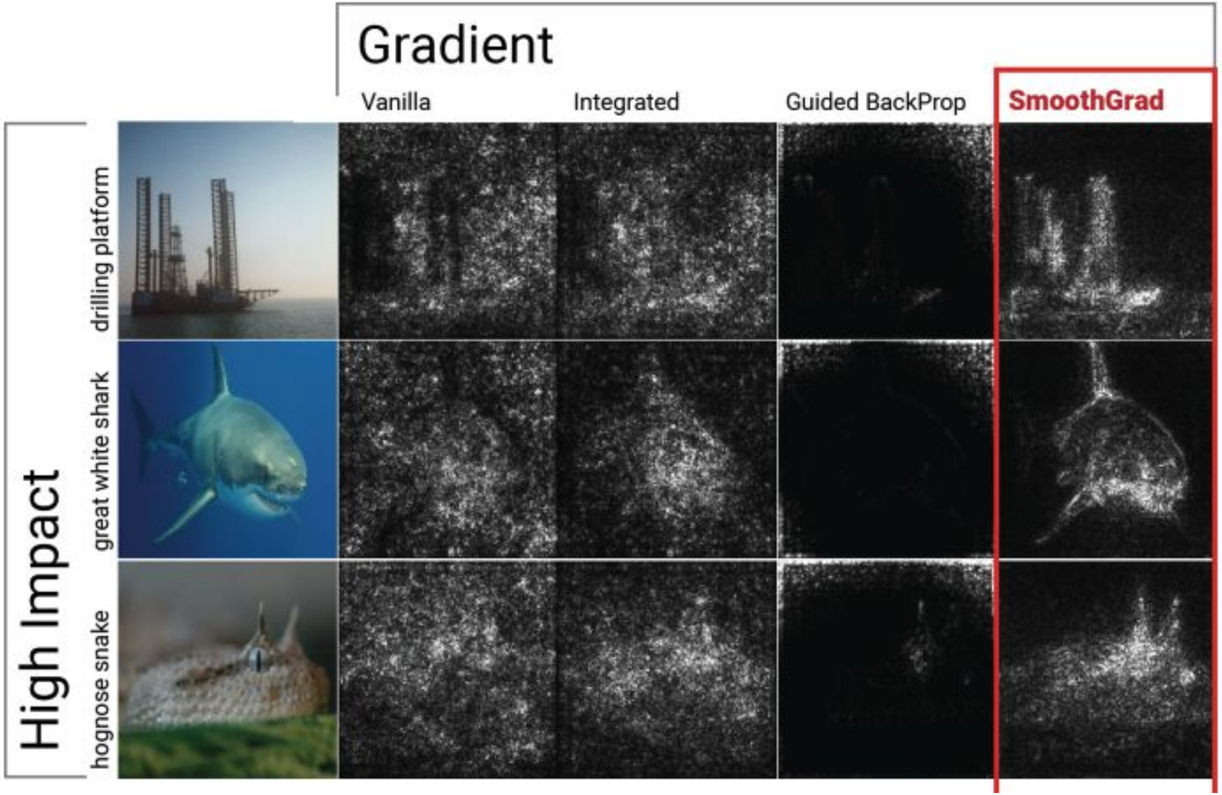


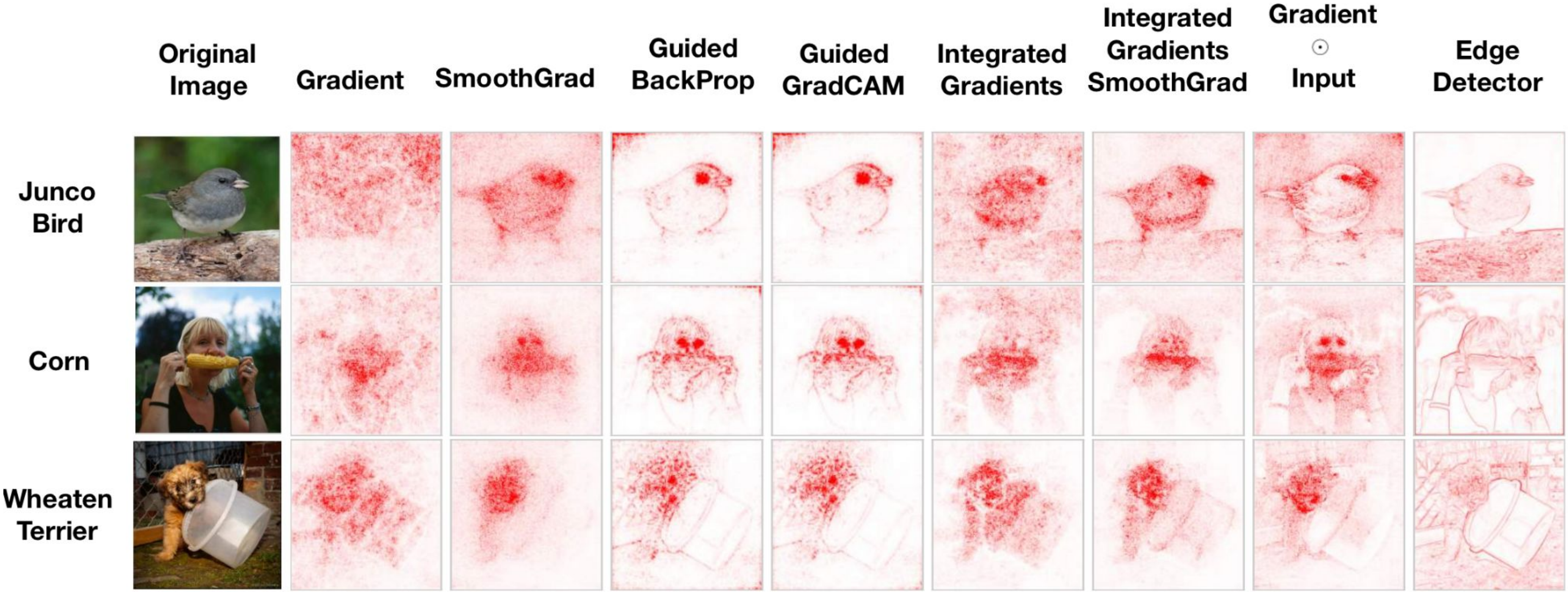
*Figure 3*. Effect of noise level (columns) on our method for 5 images of the gazelle class in ImageNet (rows). Each sensitivity map is obtained by applying Gaussian noise $\mathcal{N}(0, \sigma^2)$ to the input pixels for 50 samples, and averaging them. The noise level corresponds to $\sigma/(x_{max}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

# Smooth Gradients

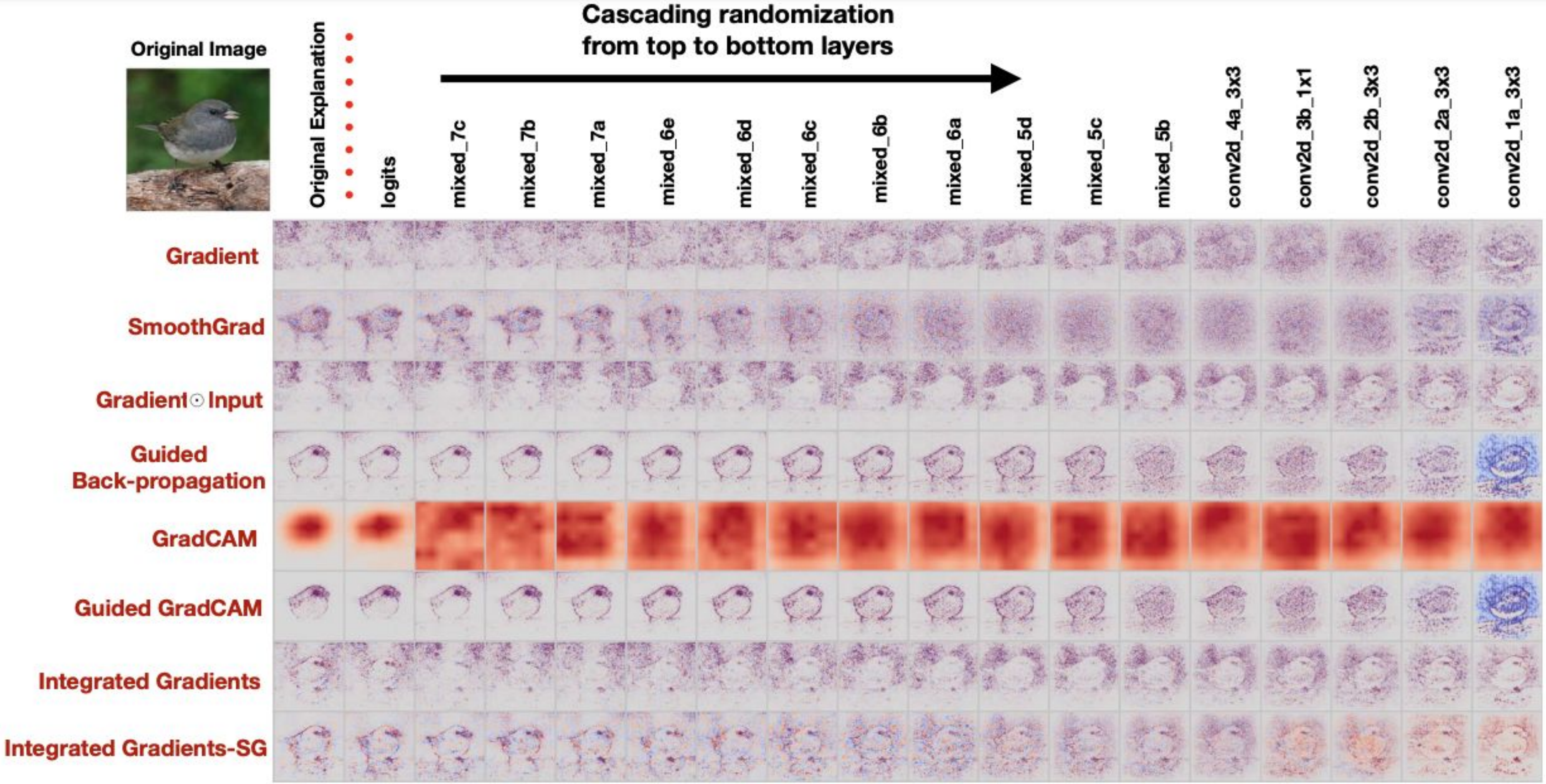
Smilkov et al. (2017) “SmoothGrad: removing noise by adding noise”



# Sanity Checks for Saliency Map Adebayo et al (2018)

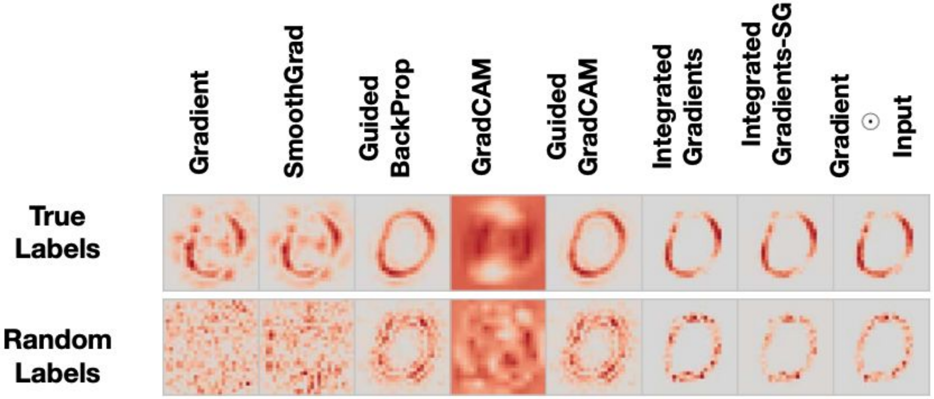


# Parameter randomization on Inception

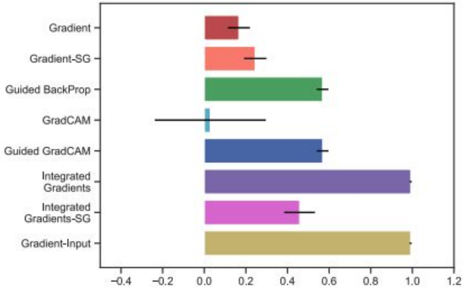


# Data (label) randomization on CNN for MNIST

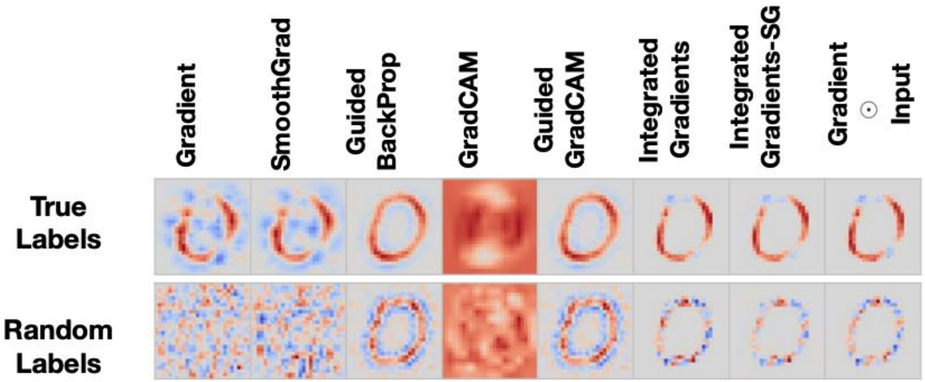
## Absolute-Value Visualization



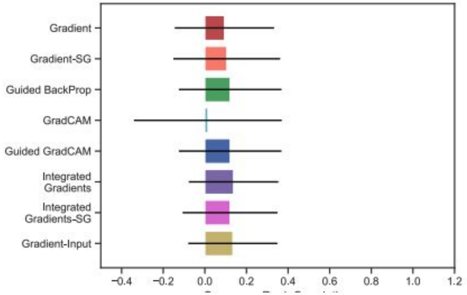
Rank Correlation - Abs



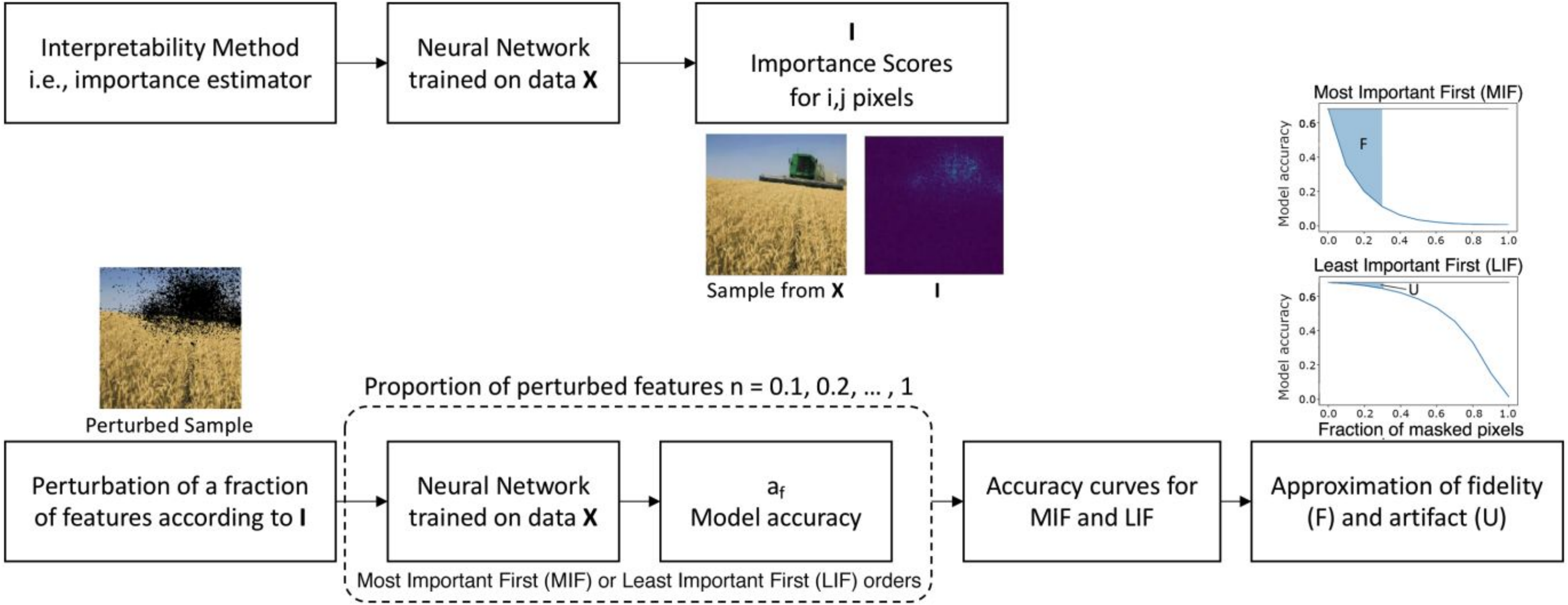
## Diverging Visualization



Rank Correlation - No Abs



# Fidelity (faithfulness) evaluation Brocki and Chung 2023



Interpretability Method i.e., importance estimator

Neural Network trained on data **X**

**I** Importance Scores for i,j pixels

Sample from **X**

**I**

Perturbed Sample

Proportion of perturbed features n = 0.1, 0.2, ... , 1

Perturbation of a fraction of features according to **I**

Neural Network trained on data **X**

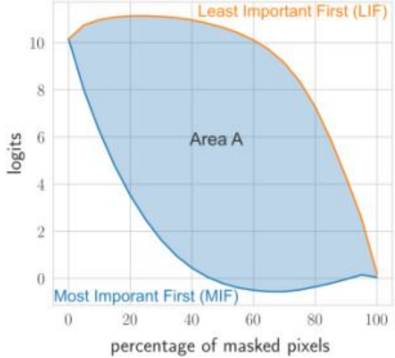
$a_f$ Model accuracy

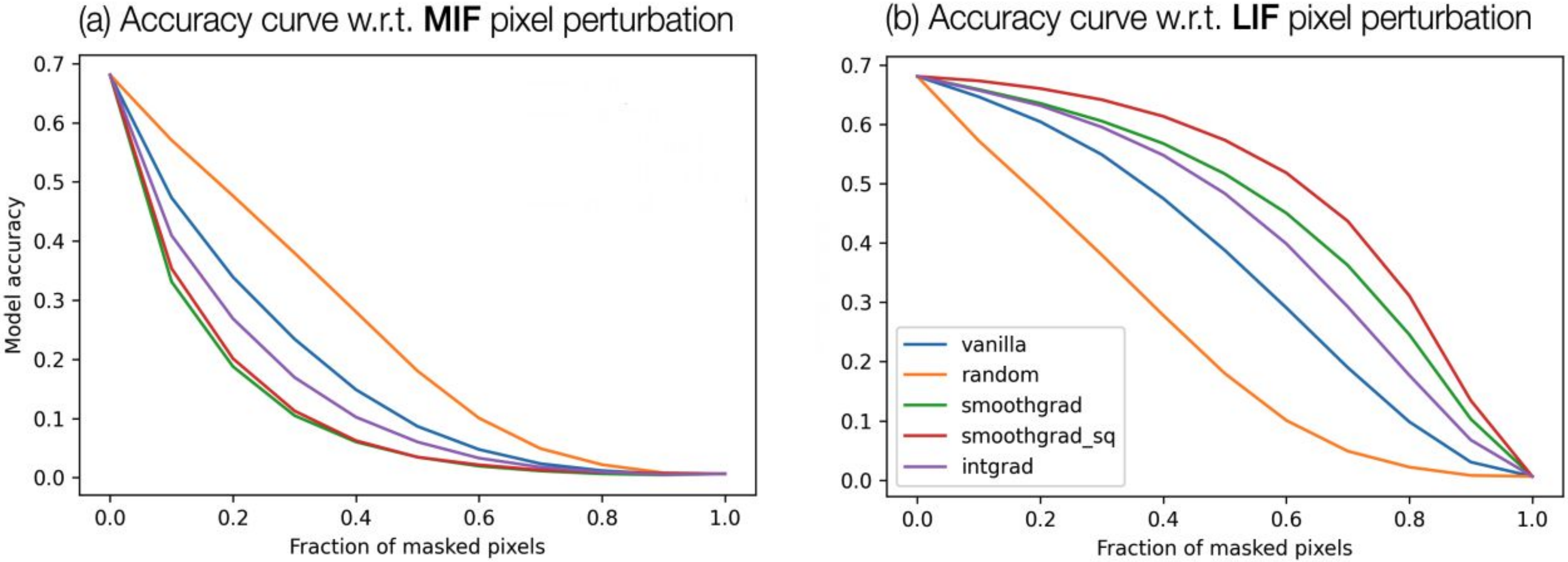
Most Important First (MIF) or Least Important First (LIF) orders

Accuracy curves for MIF and LIF

Approximation of fidelity (F) and artifact (U)

# Fidelity (faithfulness) evaluation




(a) Accuracy curve w.r.t. **MIF** pixel perturbation

(b) Accuracy curve w.r.t. **LIF** pixel perturbation

# Considerations & Pitfalls

- What humans think as important are independent of pixels that the model considers to be important
- Don’t evaluate based on visually appealing characteristics
- The samples in a dataset (e.g., ImageNet) have structures
- Complex methods to tackle this problem exacerbate/hide the problem
- How do we evaluate saliency maps when we don’t have a ground truth?