Introduction on Interpretable Deep Learning

M. Soleymani

Deep Learning

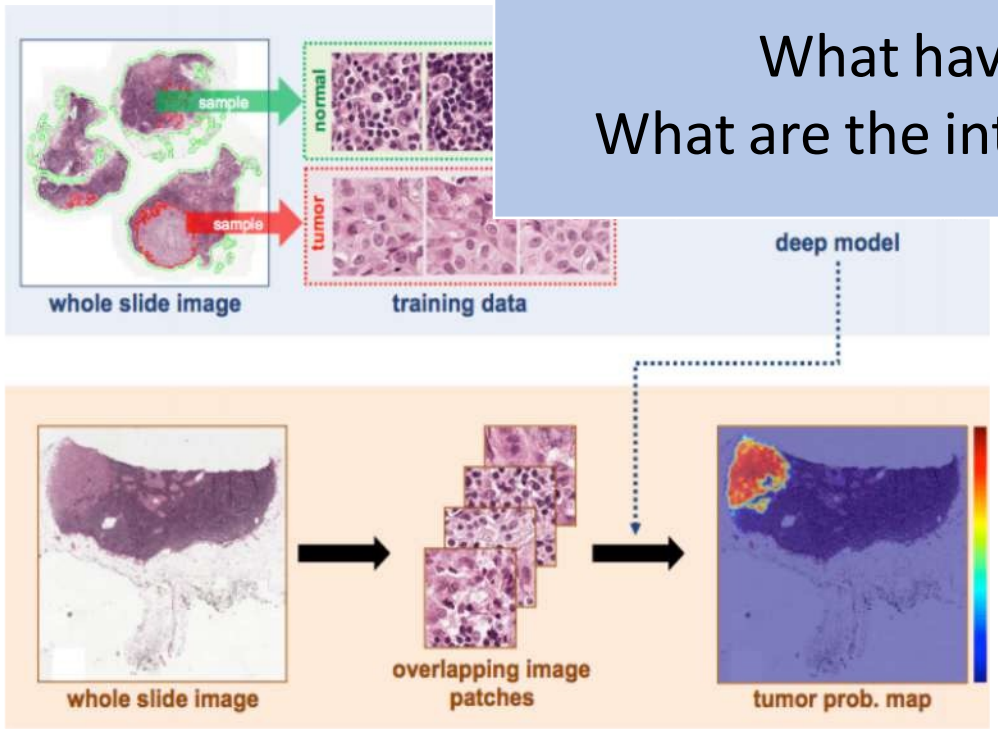
Sharif University of Technology

Spring 2024

Some slides are based on Fei Fei Li and colleagues lectures, cs231n, Stanford
and some slides are adopted from Yiyou Sun, CS839, UWM

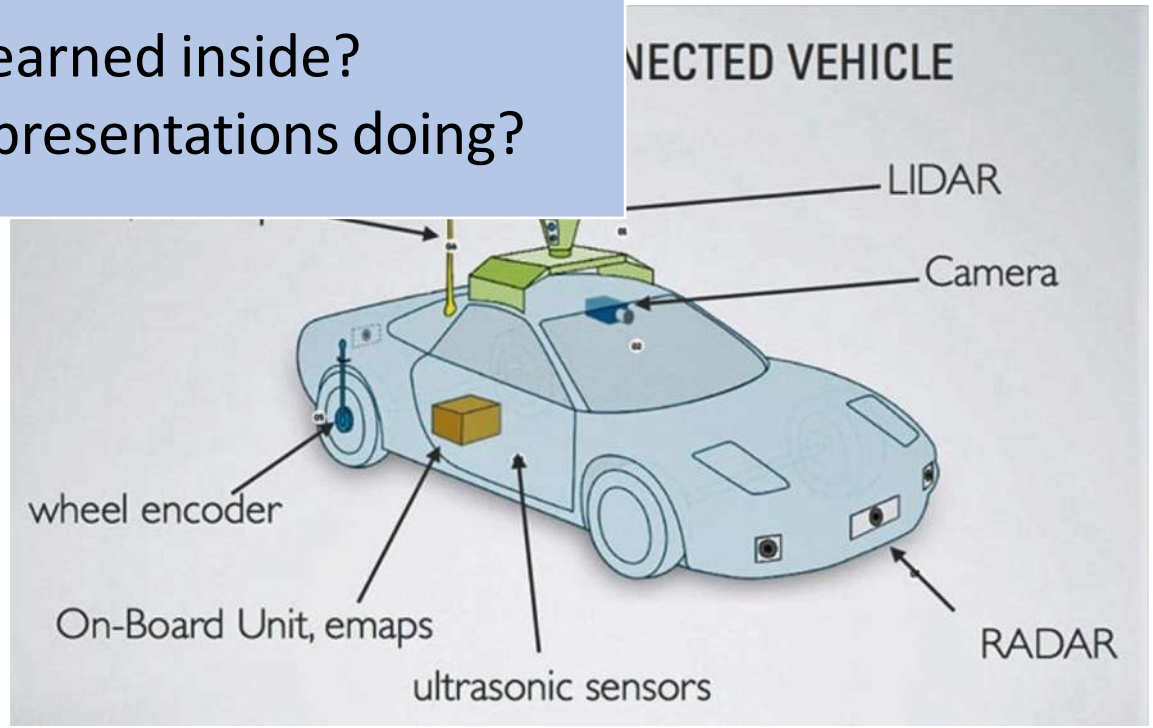
Deep Neural Networks are Everywhere

Making Medical Decision



What have been learned inside?
What are the internal representations doing?

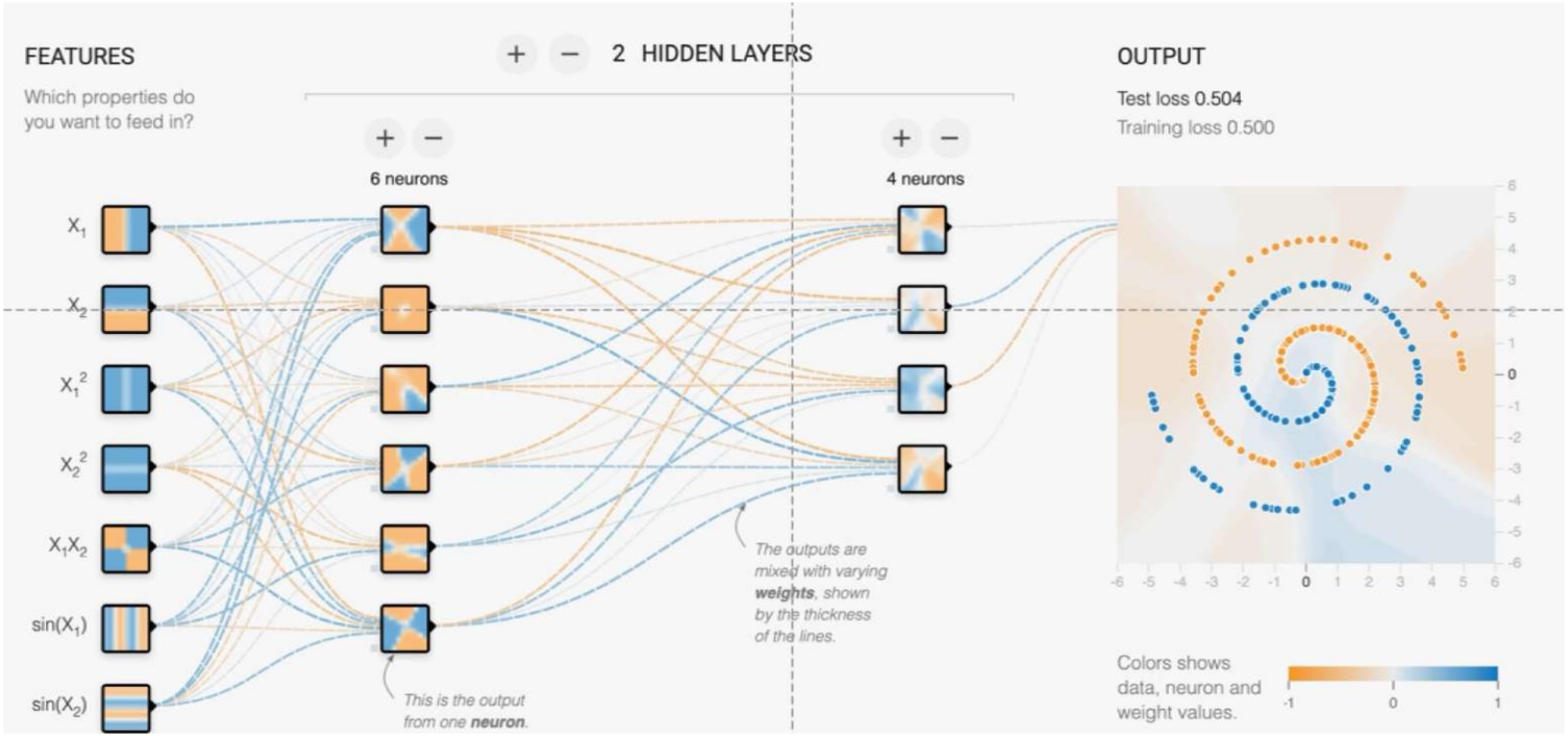
Self-Driving Car



Outline

- **Visualization Methods**
  - Feature interpretability
  - Activation maximization
- Attribution Methods
- Interpretable Models

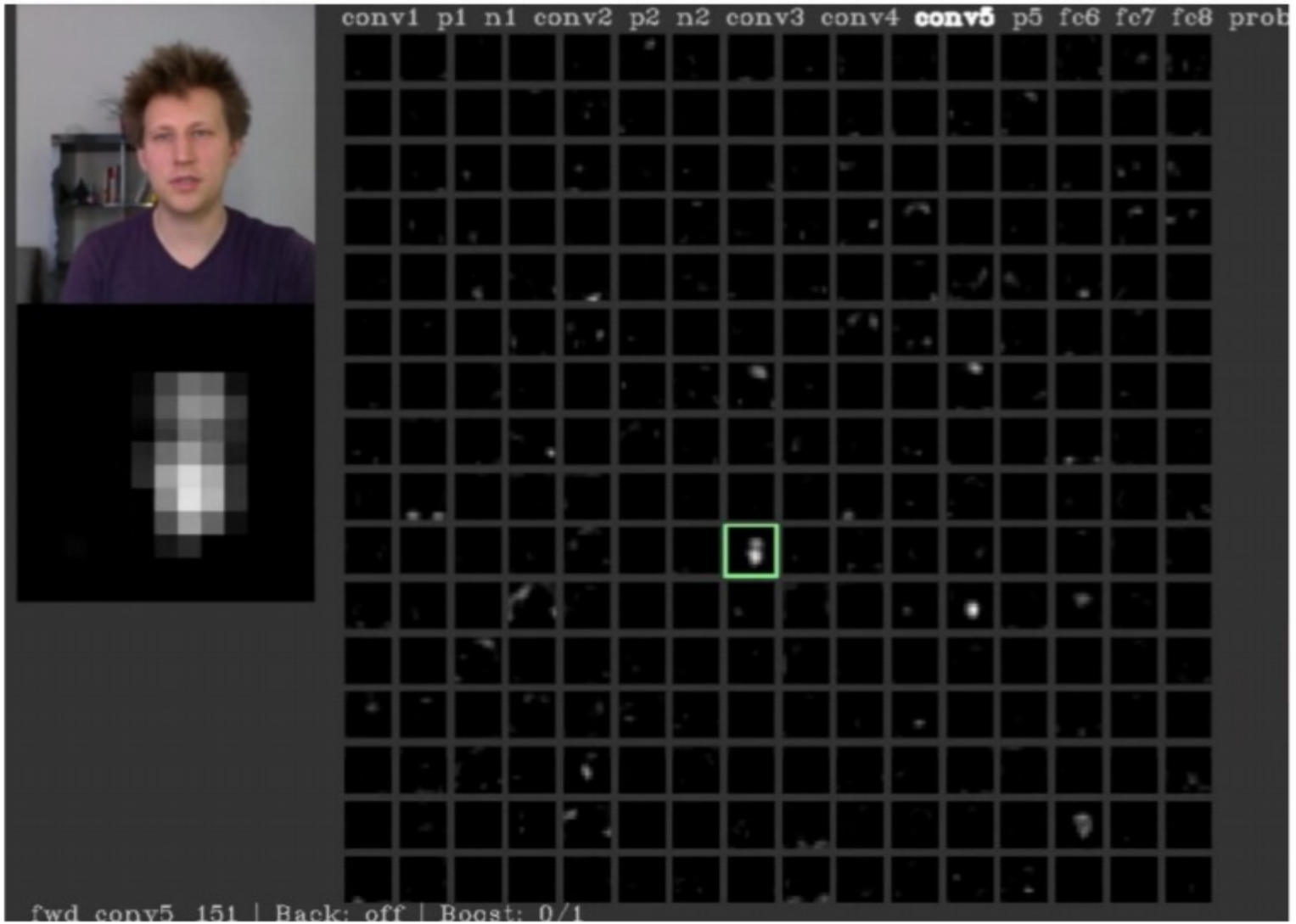
Let’s Start with a toy demo!



Source: https://playground.tensorflow.org/

Visualizing Activations

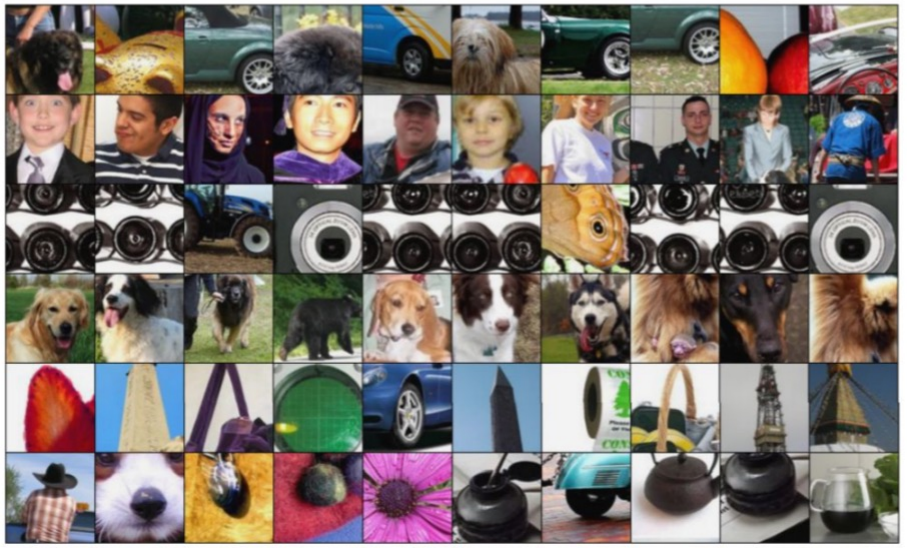
- conv5 feature map is 128x13x13;
  - visualize as 128 13x13 grayscale images

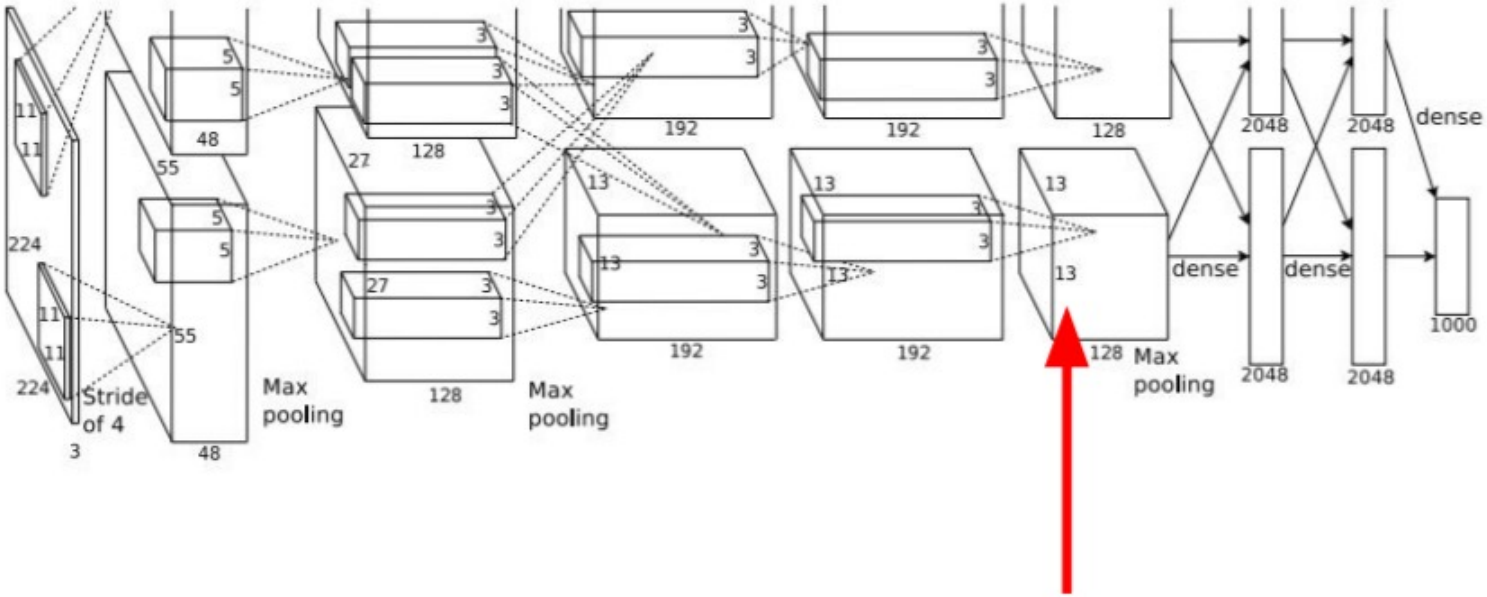


Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014.
Figure copyright Jason Yosinski, 2014. Reproduced with permission.

Maximally Activating Patches

- Pick a layer and a channel;
  - e.g. conv5 is 128 x 13 x 13, pick channel 17/128
  - Run many images through the network, record values of chosen channel
  - Visualize image patches that correspond to maximal activations





Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015
Figure copyright Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller, 2015; reproduced with permission.

Broadly and Densely (Broden) Annotated Dataset

ADE20K
Zhou et al, CVPR'17

Pascal Context
Mottaghi et al, CVPR'14

Pascal Part
Chen et al, CVPR'14

Open-Surfaces
Bell et al, SIGGRAPH'14

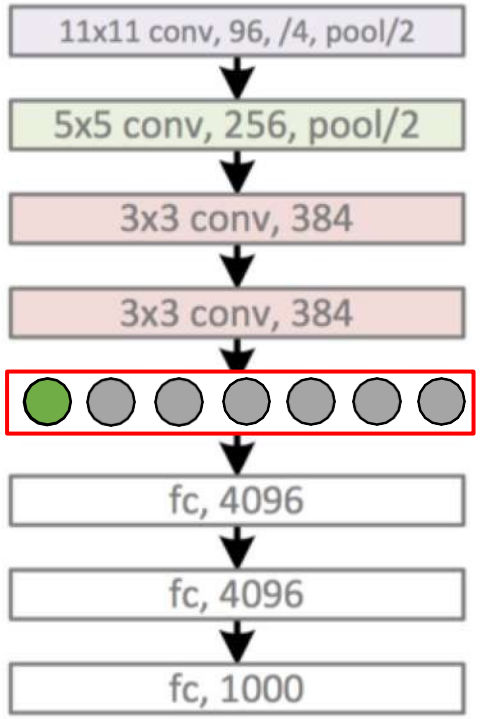
Describable Textures
Cimpoi et al, CVPR'14

Colors

Total = 63,305 images
1,197 visual concepts

Zhou et al., Network Dissection. PAMI 2018

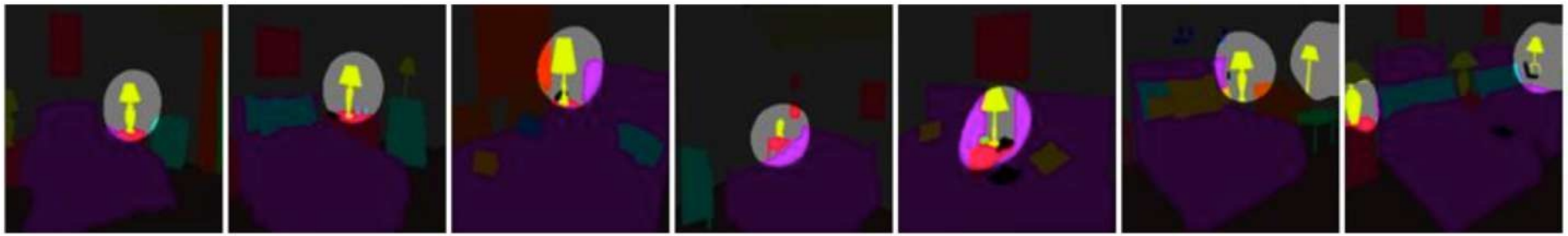
From Visualization to Interpretation



Top Activated Images

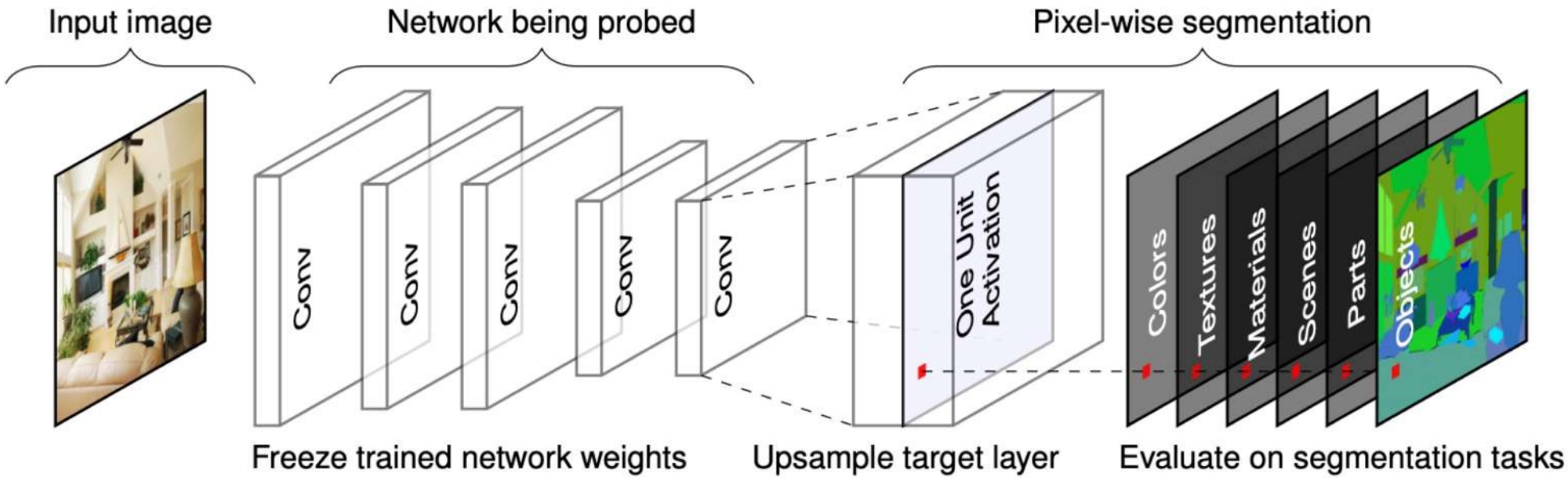
Unit 1

Lamp Intersection over Union (IoU)= 0.12



Zhou et al., Network Dissection. PAMI 2018

IoU Score Calculation



$$IoU(Unit, Concept) = \frac{\sum|Area(Unit) \cap Area(Concept)|}{\sum|Area(Unit) \cup Area(Concept)|}$$

Zhou et al., Network Dissection. PAMI 2018

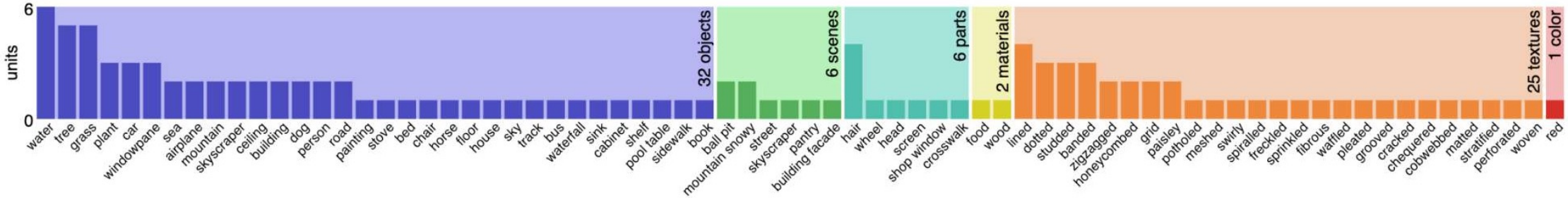
Network Dissection

AlexNet-Places205 conv5 unit 138: heads

AlexNet-Places205 conv5 unit 215: castles

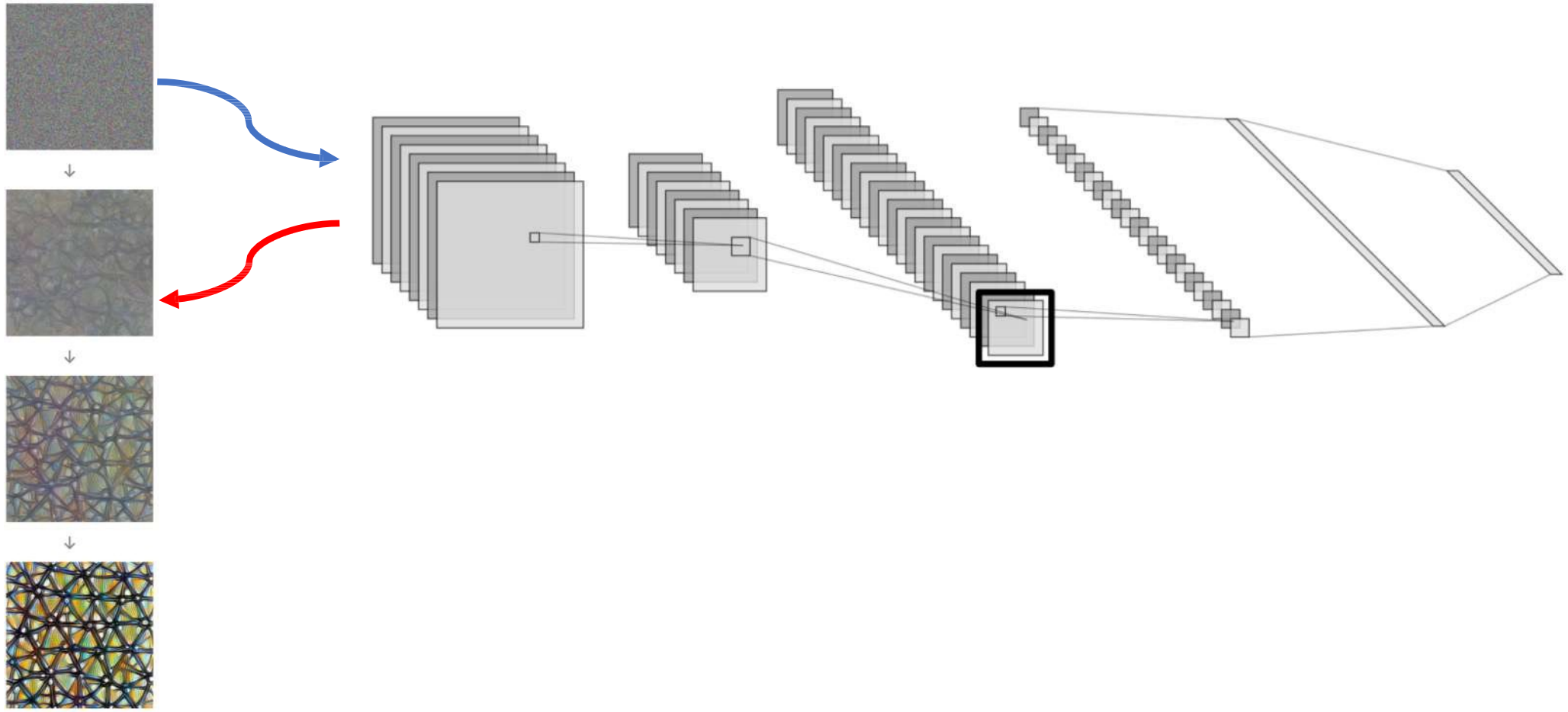
AlexNet-Places205 conv5 unit 13: lamps

AlexNet-Places205 conv5 unit 53: stairways



Zhou et al., Network Dissection. PAMI 2018

Deep Feature Visualization by Gradient Ascent



https://distill.pub/2017/feature-visualization/

Tricks to Make Natural Visualization

- **Frequency penalization**
  - Penalize variance between neighboring pixels
  - Penalize high-frequency noise by blurring the image each optimization step
- **Transformation robustness**
  - Generate images that still activate the optimization target even with **jitter**, **rotation** or **scaling**
- **Learned priors**
  - learn a generative model of the real data to generate photorealistic visualizations

Visualizing CNN features: Activation maximization

- Activation Maximization: synthesizes an image that highly activates a neuron

- Gradient ascent: Generate a synthetic image that maximally activates a neuron

$$I^* = \arg\max_I f(I) + R(I)$$

Neuron value

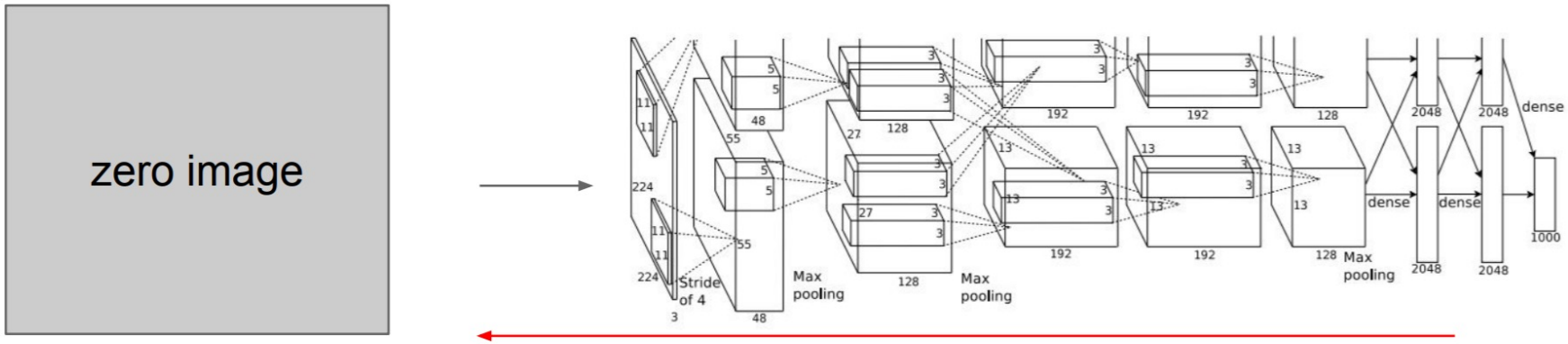
Natural image regularizer

Visualizing features: Activation maximization

1. Initialize image to zeros

$$\arg\max_I S_c(I) - \lambda \|I\|_2^2$$

score for class c (before Softmax)



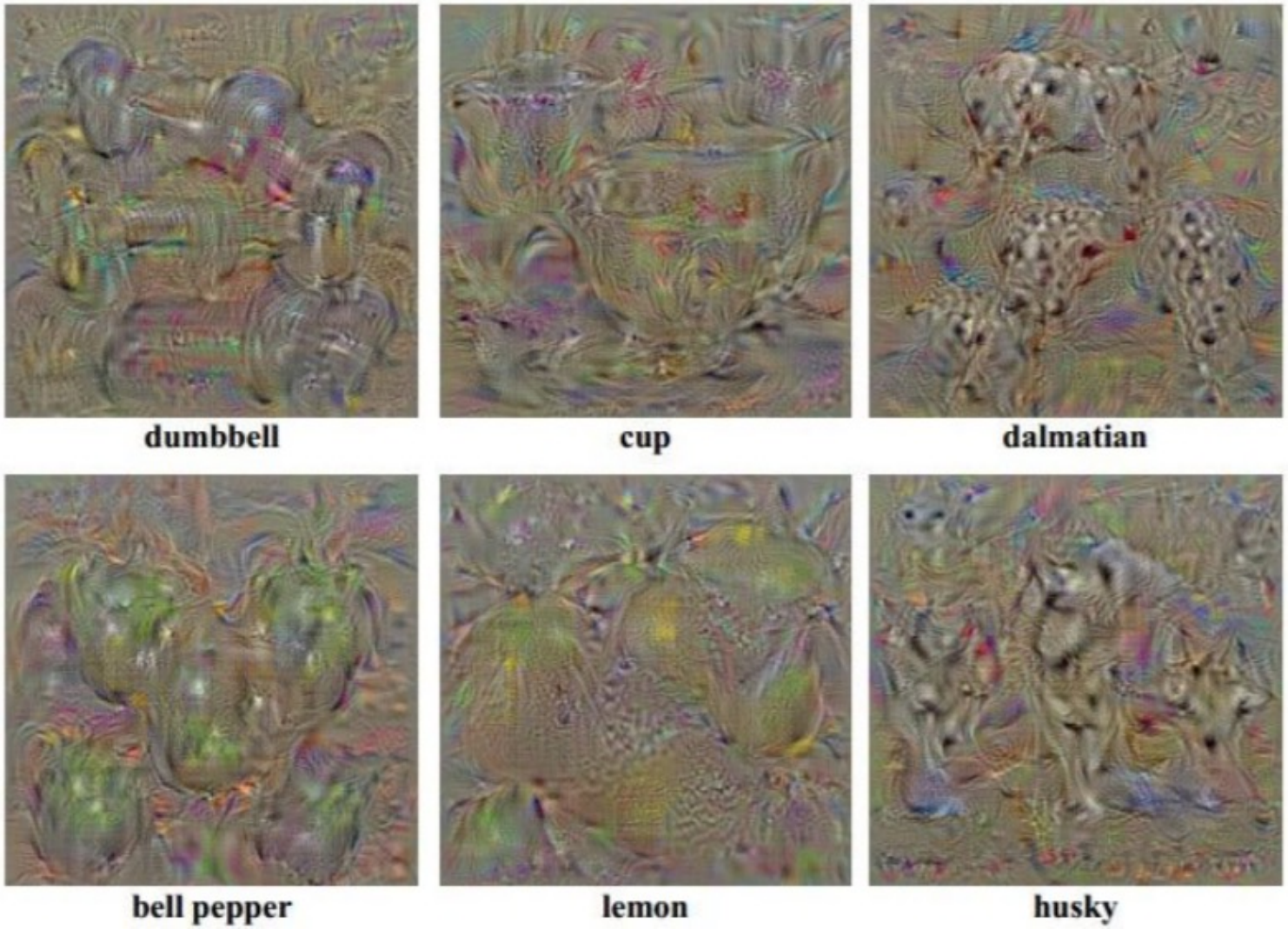
Repeat:

2. Forward image to compute current scores
3. Backprop to get gradient of neuron value with respect to image pixels
4. Make a small update to the image

Visualizing CNN features: Activation maximization

$$\arg\max_I S_c(I) - \lambda \|I\|_2^2$$

Simple regularizer: Penalize L2 norm of generated image

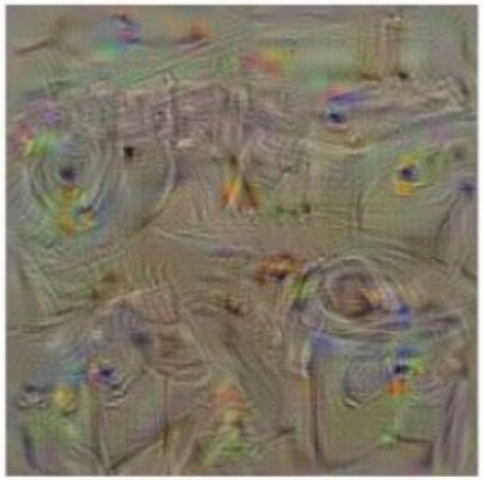


Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

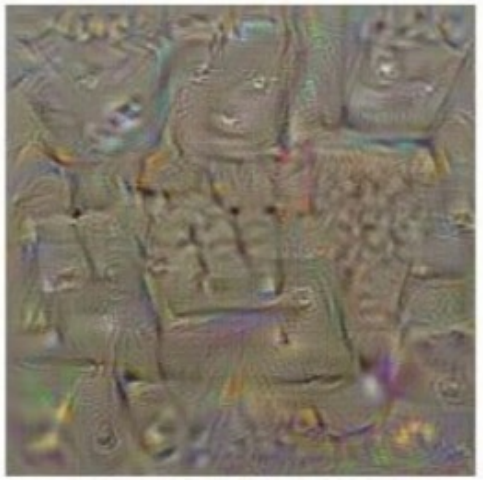
Visualizing CNN features: Activation maximization

$$\arg\max_I S_c(I) - \lambda \|I\|_2^2$$

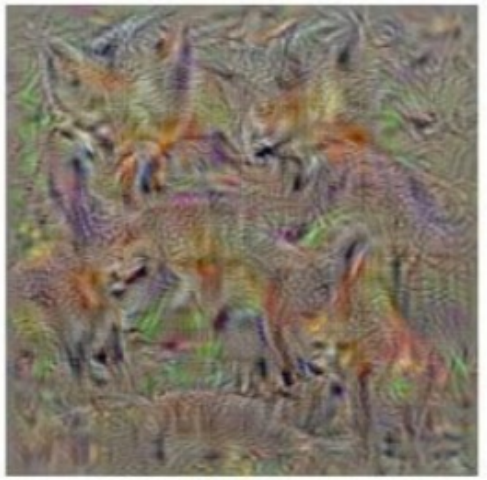
Simple regularizer: Penalize L2 norm of generated image



washing machine



computer keyboard



kit fox

goose

ostrich

limousine

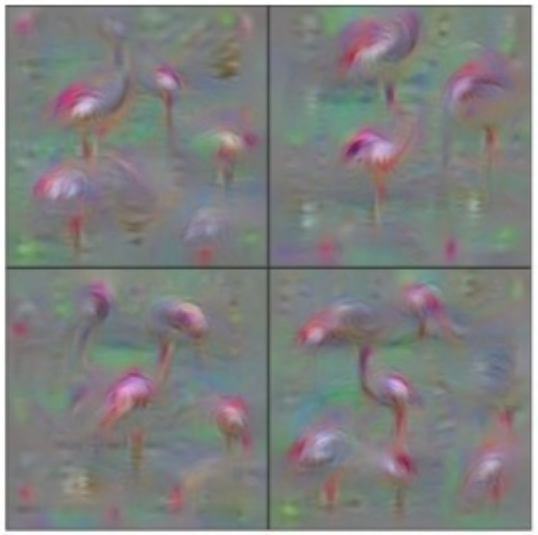
Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Im
Models and Saliency Maps", ICLR Workshop 2014.

Visualizing CNN features: Activation maximization

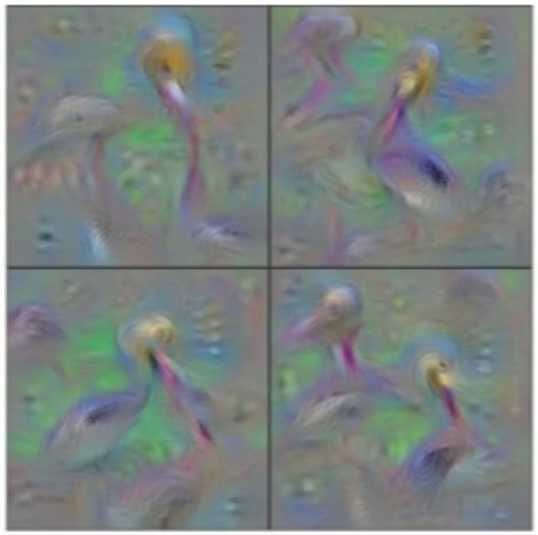
$$\arg\max_I S_c(I) - \lambda \|I\|_2^2$$

Better regularizer: Penalize L2 norm of image; also during optimization periodically

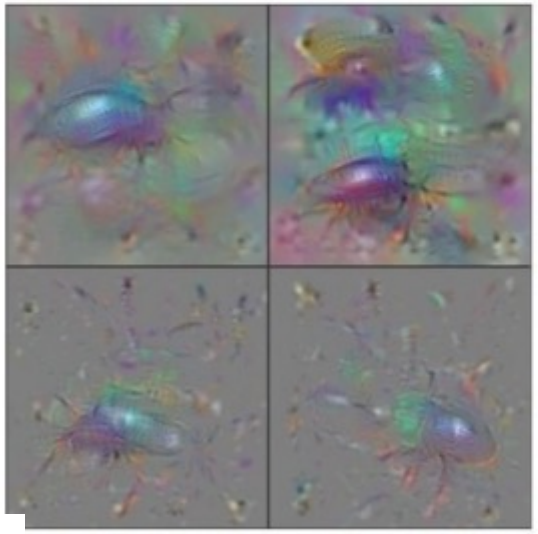
(1) Gaussian blur image
(2) Clip pixels with small values to 0
(3) Clip pixels with small gradients to 0



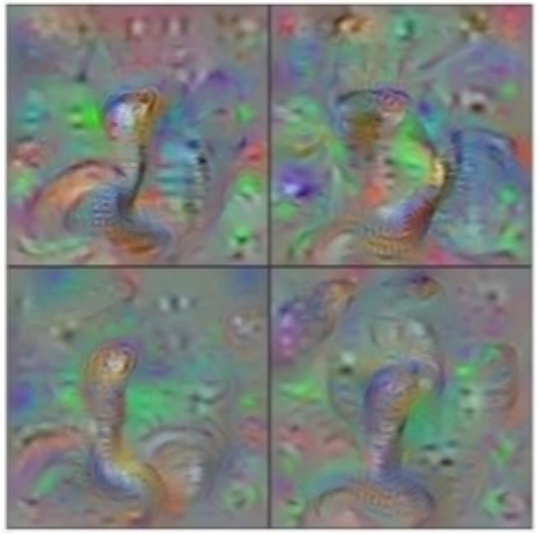
Flamingo



Pelican



Ground Beetle



Indian Cobra

Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014.

Interpretability Approaches

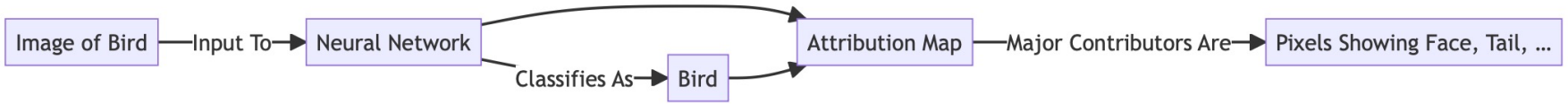
- **Visualization**
  - **Feature Interpretability**
    - visualizing the features that the model has learned to understand what it is learning
  - **Activation Maximization**
    - maximizing the activation of a neuron or layer to understand what the model is learning

- **Attribution (Saliency Maps)**
  - which parts of the input are most important

- Interpretable Models

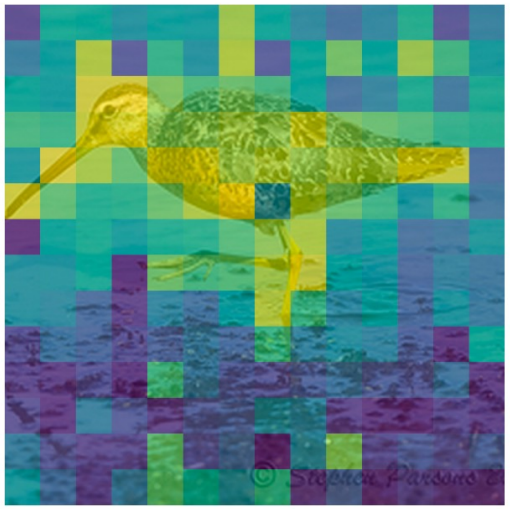
Attribution

- Attribution: determining how **individual features** (or groups of features) in the **input data** contribute to a neural network's output decision

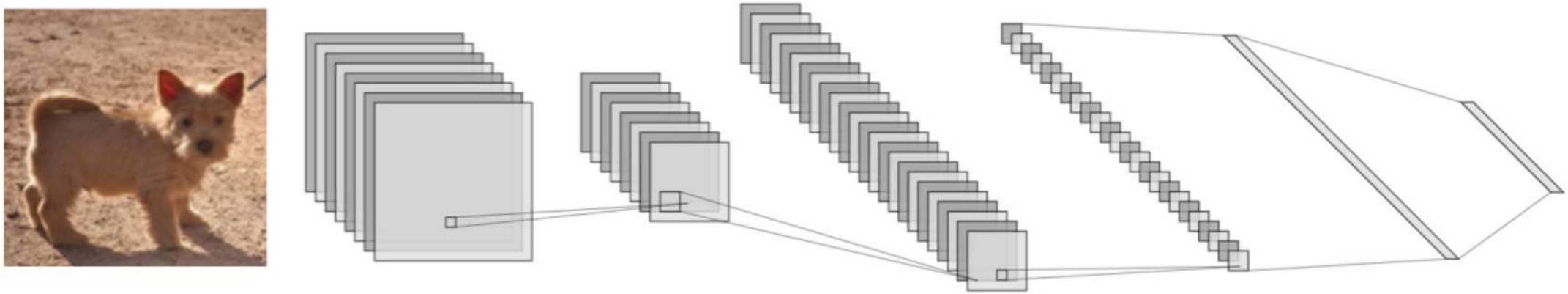


- Input

- Attribution Map



Which pixels matter for classification?



Simple Gradient: The Most Obvious Attribution Method

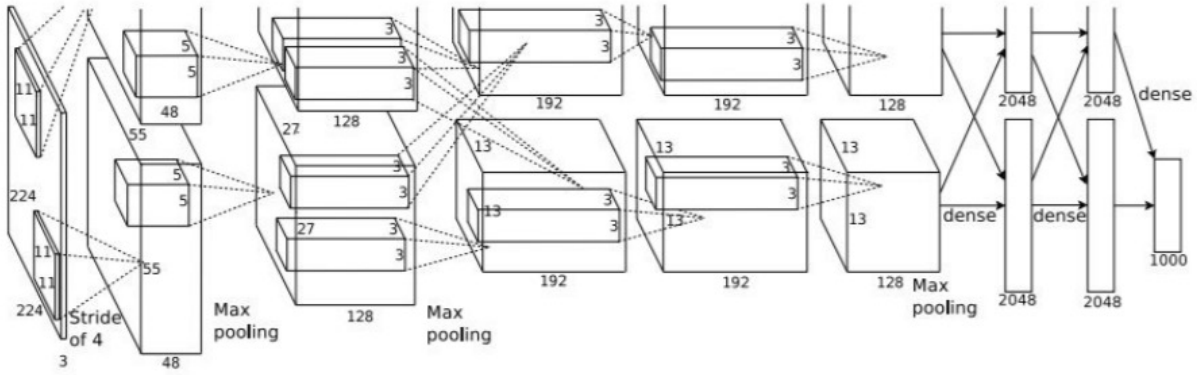
- Definition: Explains the change in the output of a model with respect to changes in the input.

$$\nabla_x f_\theta(\boldsymbol{x})$$

- Limitations
  - It measures the sensitivity of the output to each input feature, rather than the direct contribution of that feature to the current output.
  - The underlying network needs to be differentiable.

Saliency Maps

How to tell which pixels matter for classification?

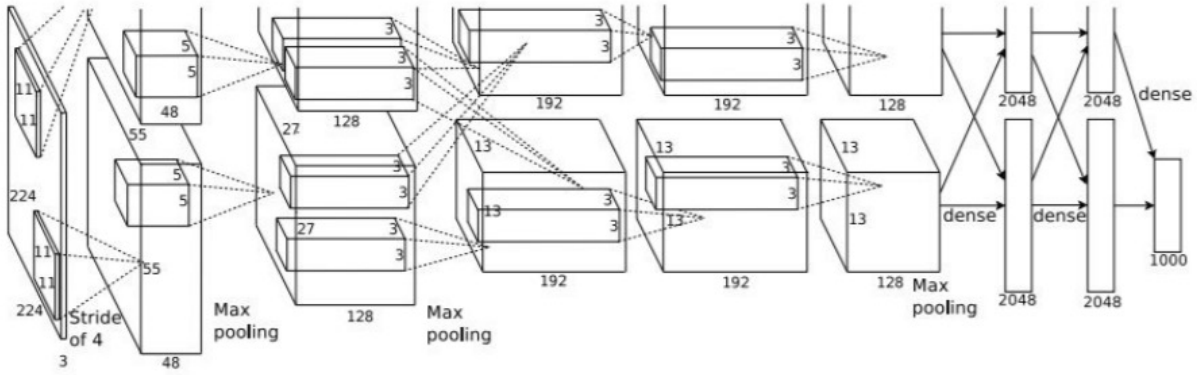


Dog

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

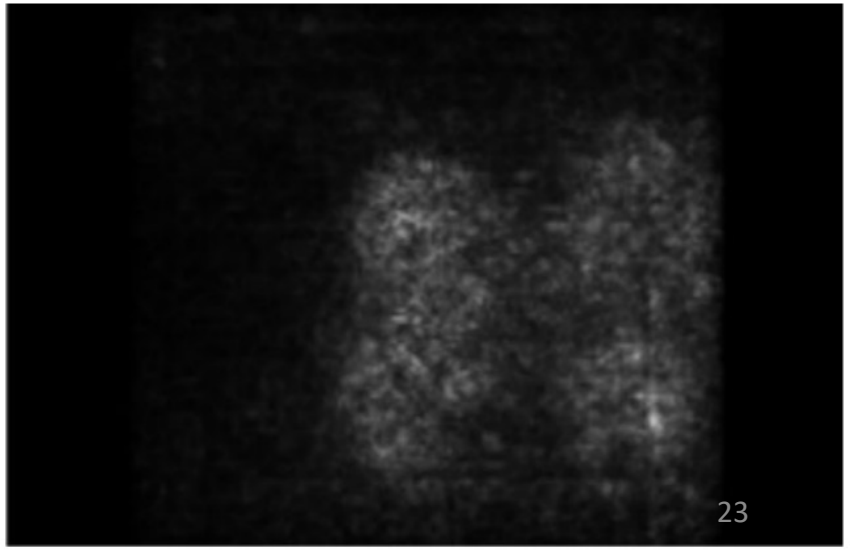
Which pixels matter: Saliency via Backprop

How to tell which pixels matter for classification?



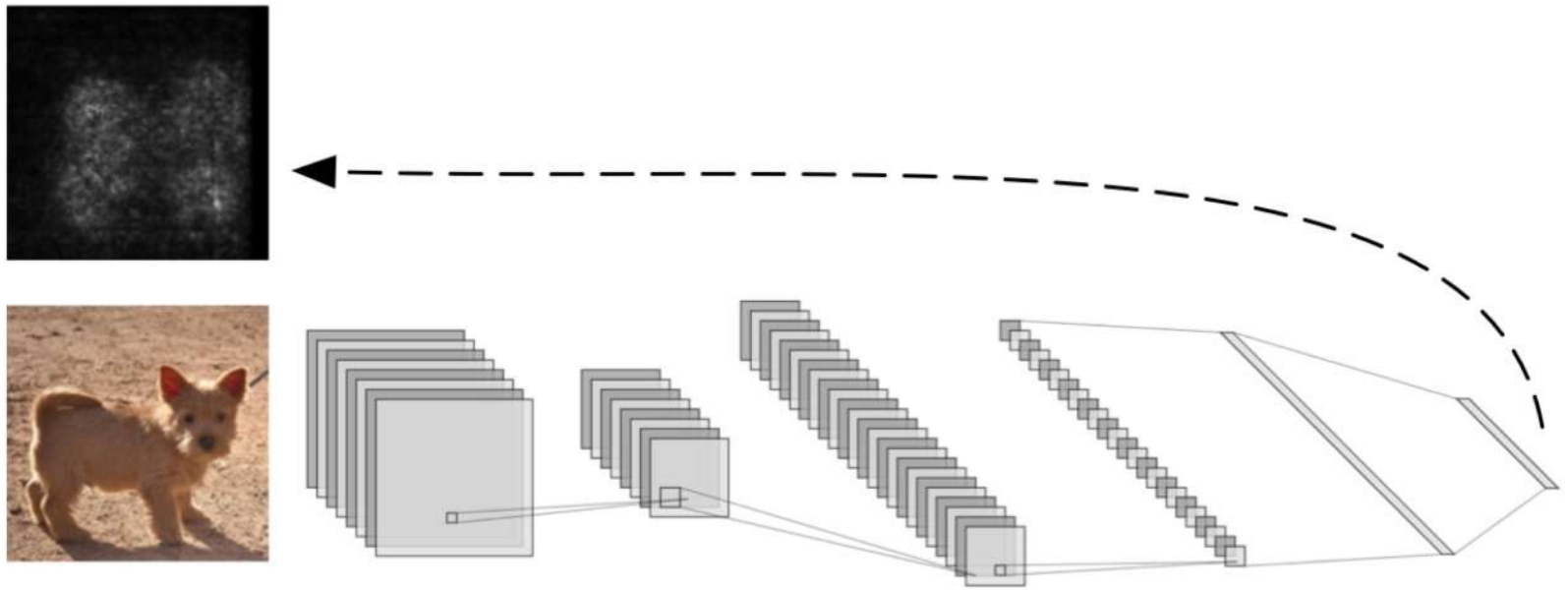
Dog

- Compute gradient of (unnormalized) class score with respect to image pixels,
  - It is computed for a pair of class and image
  - take absolute value and max over RGB channels



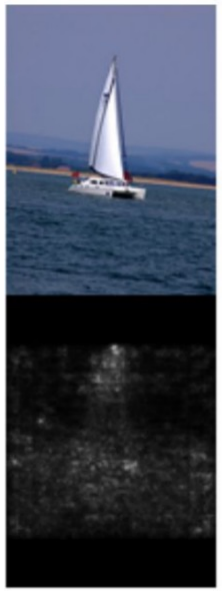
Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

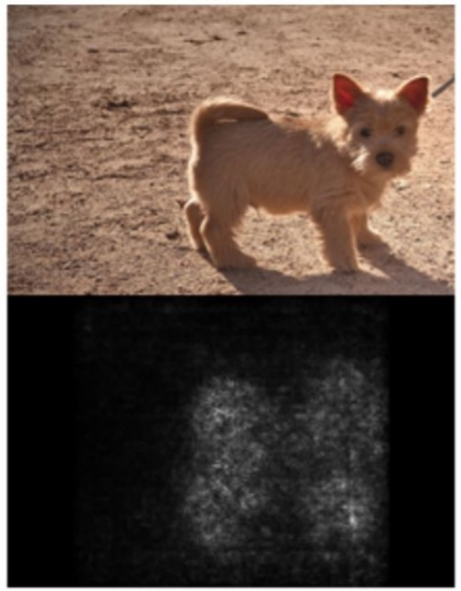
Saliency Map: Gradient Visualization

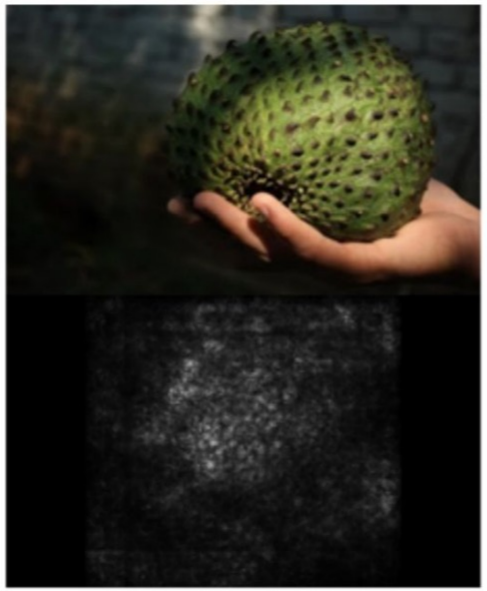


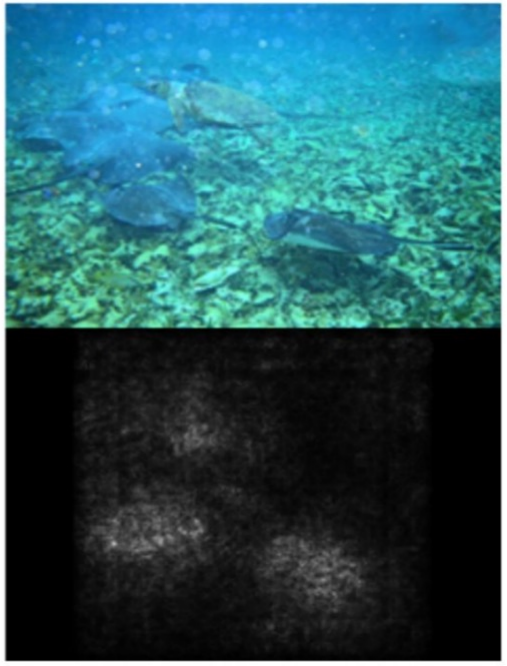
Simonyan et al., “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”, ICLR Workshop 2014

Saliency Maps





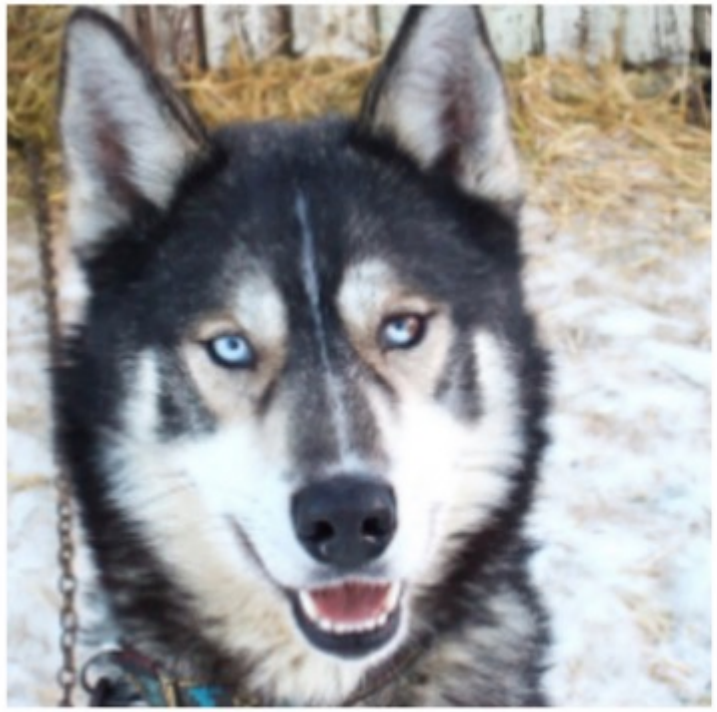




Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

Saliency maps: Uncovers biases

- Such methods also find biases
- wolf vs dog classifier looks is actually a snow vs no-snow classifier

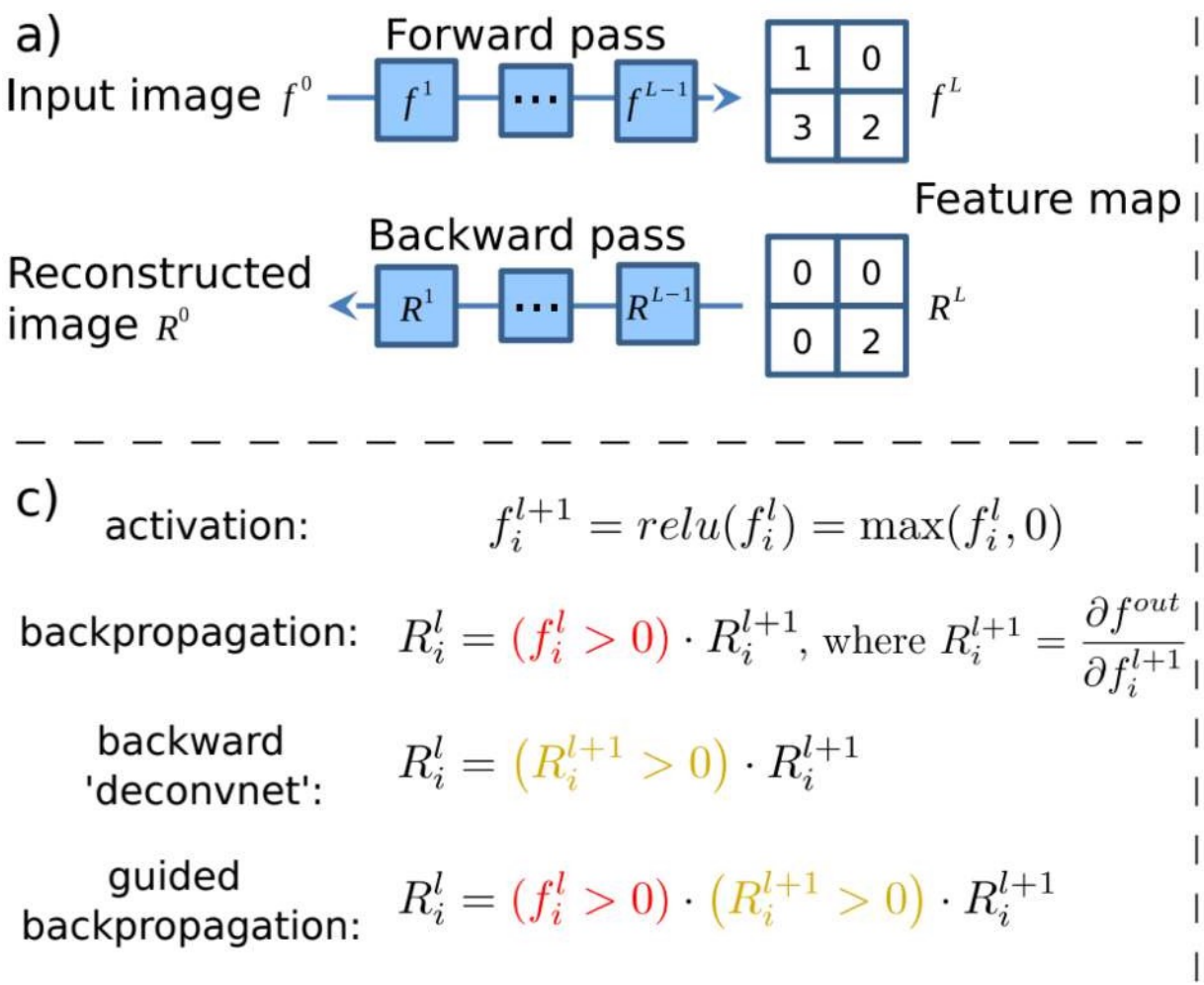


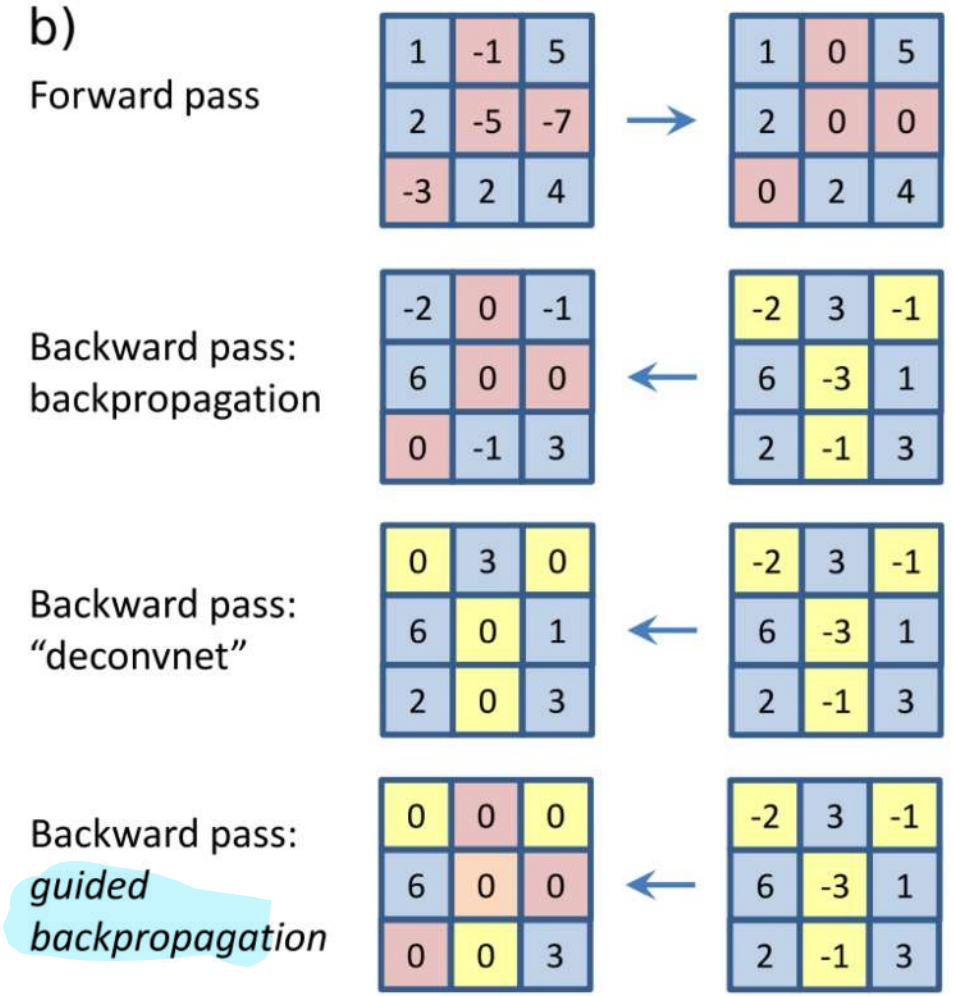
(a) Husky classified as wolf

(b) Explanation

Figures copyright Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, 2016; reproduced with permission.
Ribeiro et al, ""Why Should I Trust You?" Explaining the Predictions of Any Classifier", ACM KDD 2016

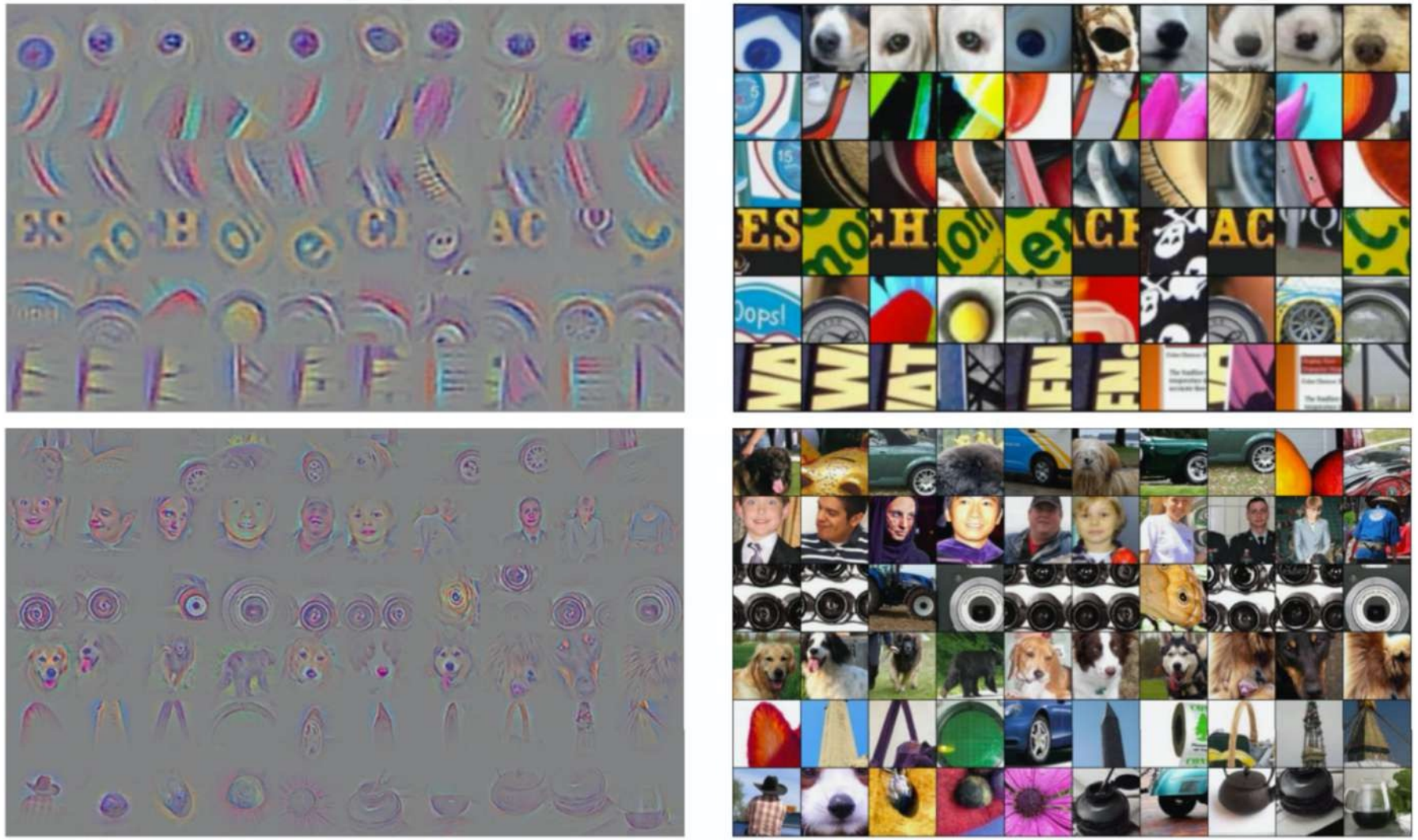
Guided Backprop





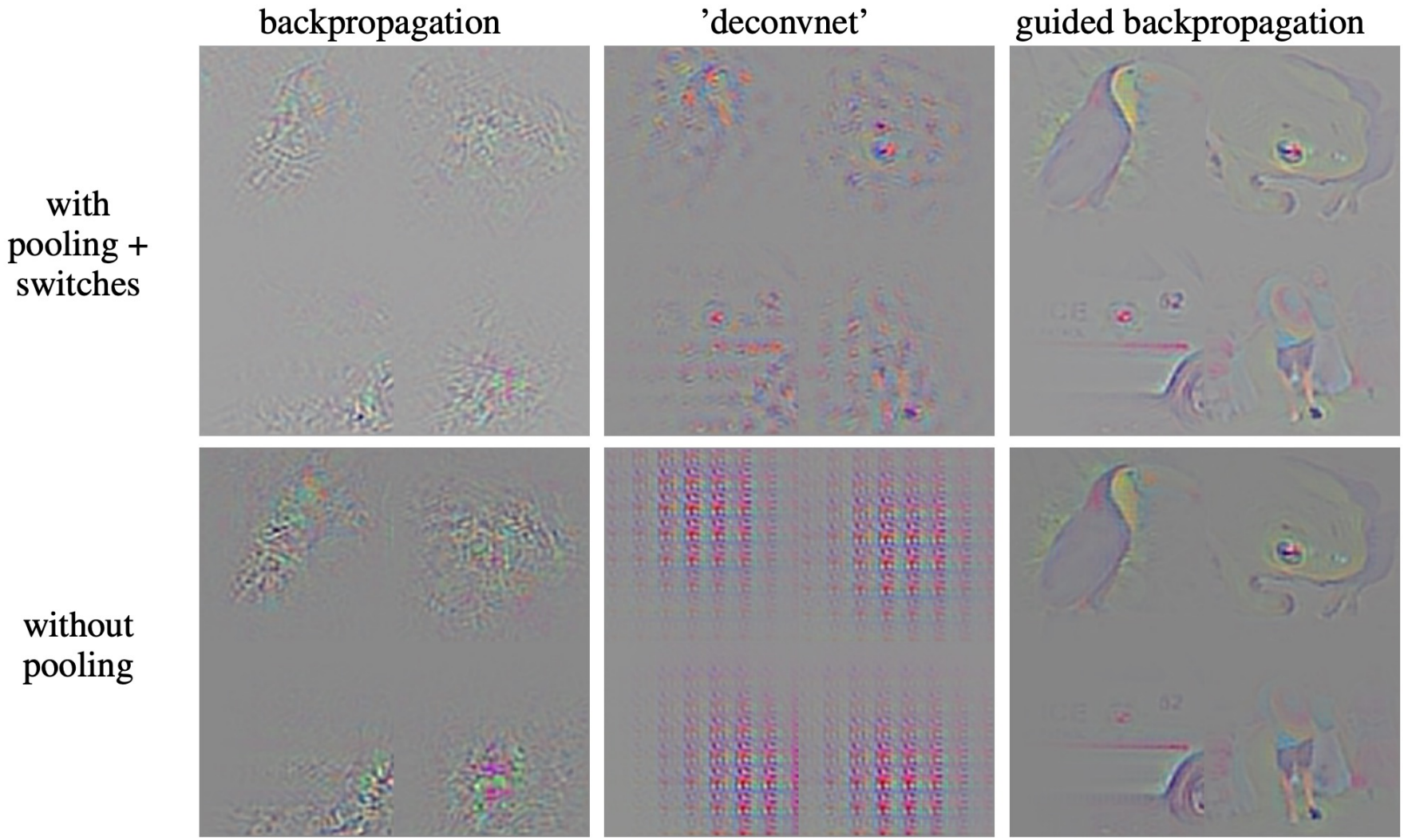
Springenberg et al., "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015

Guided Backprop



Springenberg et al., “Striving for Simplicity: The All Convolutional Net”, ICLR Workshop 2015

Guided Backprop



Input × Gradient

- Definition: Multiplies each input feature with its corresponding gradient (elementwise):

$$\boldsymbol{x} \odot \nabla f_{\boldsymbol{\theta}}(\boldsymbol{x})$$

- Advantages over Simple Gradient:
  - It takes into account the sign and magnitude of the input features.
  - Provides a more global estimate of feature contribution, moving beyond just local sensitivity.

- Limitations:
  - The model needs to be differentiable.
  - Assumes a constant gradient across the input range from 0 to $\boldsymbol{x}$. I.e., it assumes the model is linear.

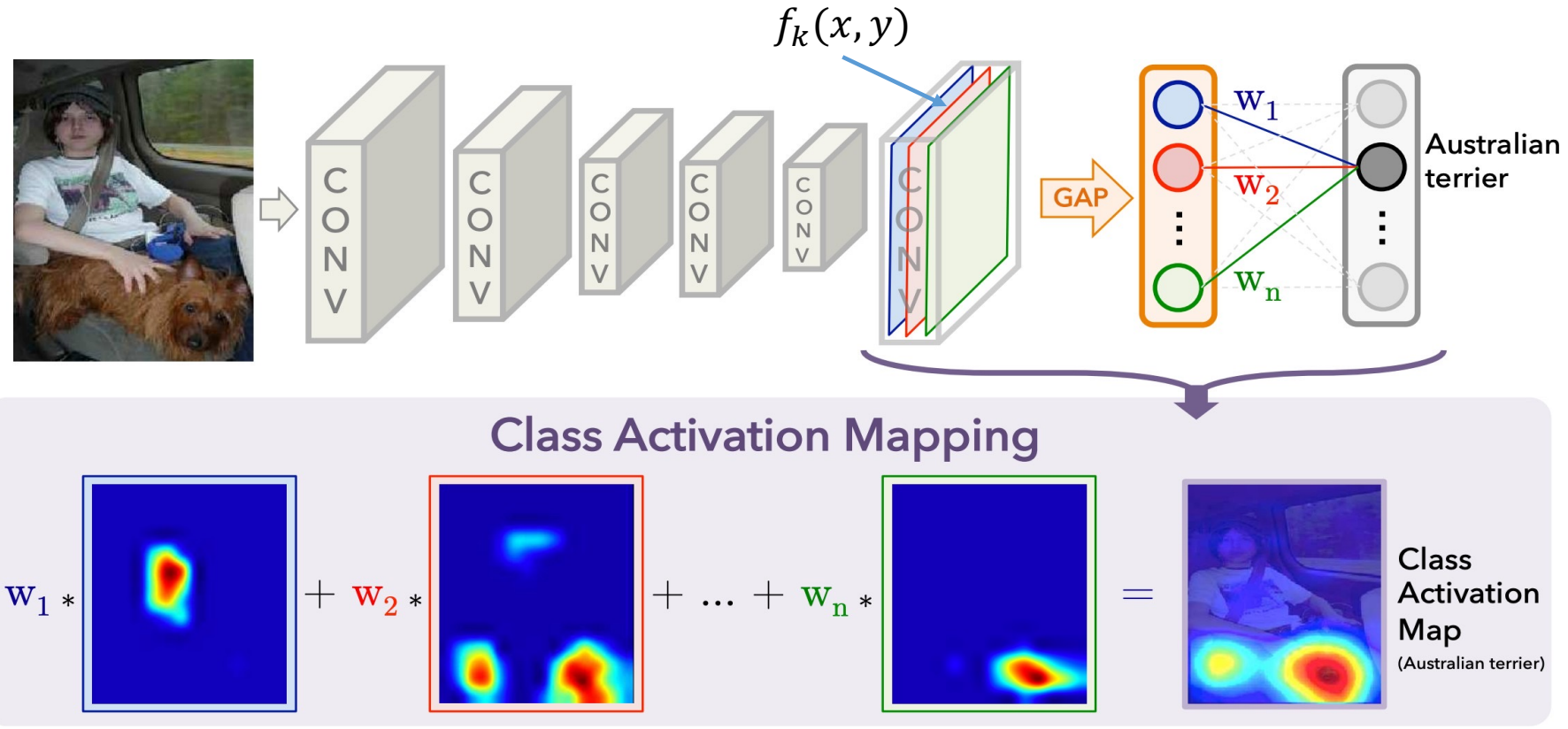
Shrikumar et al., Not Just a Black Box: Learning Important Features Through Propagating Activation Differences", 2016

Class Activation Mapping (CAM)

- Discriminative image regions used by the CNN to **identify the specified category**

- Replace fully connected layers with Global Average Pooling (GAP)
  - Minimize the number of parameters
  - Acts as a regularizer preventing overfitting

Class Activation Mapping (CAM)

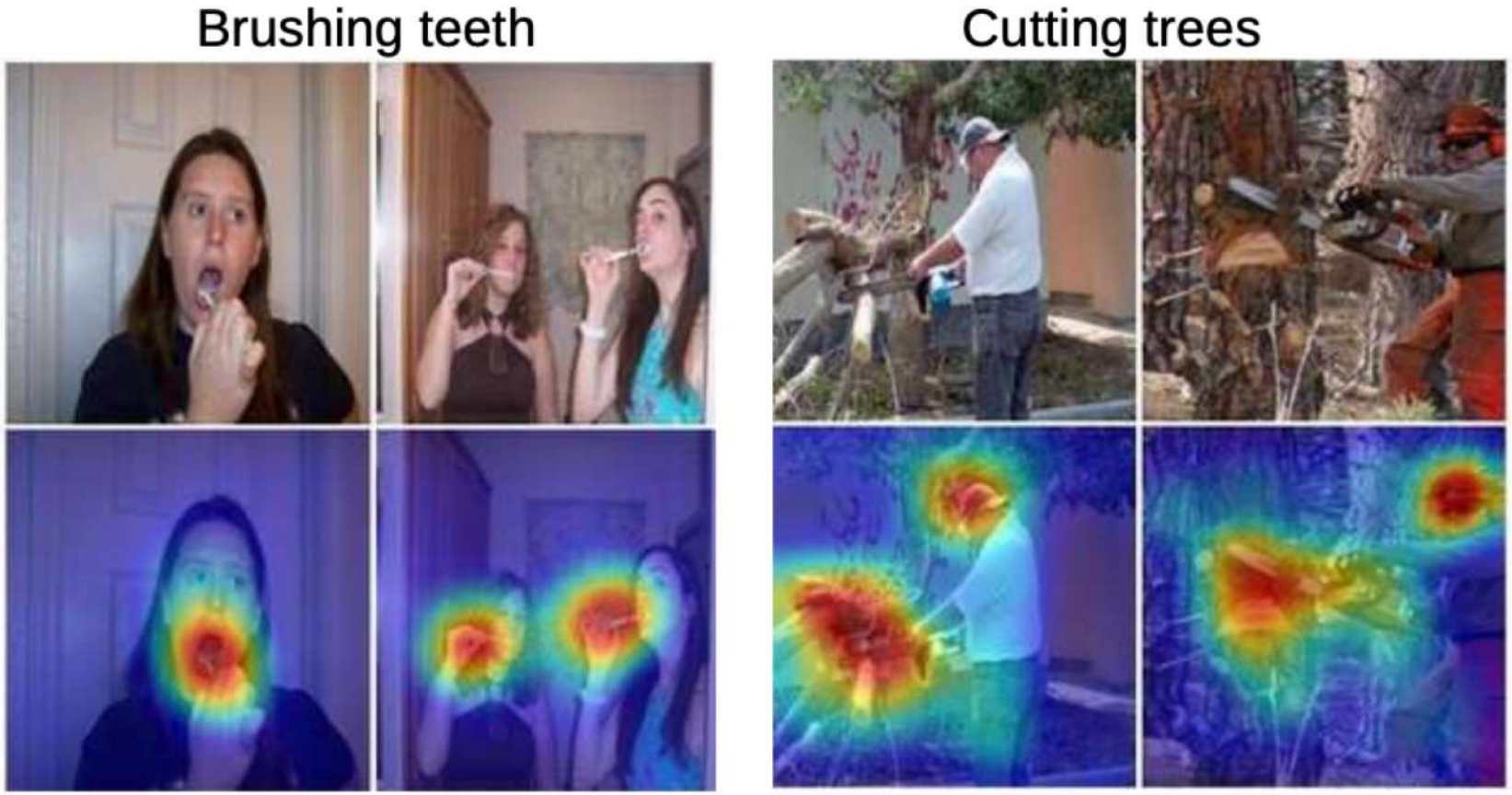


- $f_k(x, y)$: ***k*-th feature map** (activation of unit k in spatial location (x, y))
- $F^k = \sum_{x,y} f_k(x, y)$: Result of Global Average Pooling (**GAP**)
- $S_c = \sum_k w_k^c F_k$: logit (input to Softmax layer for class c)
- $M_c(x, y) = \sum_k w_k^c f_k(x, y)$: **CAM** for class c

$$M_c(x, y) = \sum_k w_k^c f_k(x, y)$$

Bolei et al., "Learning Deep Features for Discriminative Localization", CVPR 2016

CAM: Can we use last feature map as attribution?



Bolei et al., “Learning Deep Features for Discriminative Localization”, CVPR 2016

CAM: What is it?

- Enables Classification CNNs to learn to perform localization
- CAM indicates the discriminative regions used to identify that category
- No explicit bounding box annotations required

CAM: Properties

- CAM trades of model complexity and performance

- Shotcoming:
  - Must put GAP between the last conv layer and the output layer
    - Needs retraining
  - Requires feature maps to directly precede softmax layers
    - Such architectures may achieve inferior accuracies compared to general networks on other tasks
  - Inapplicable to other tasks like VQA, Image Captioning

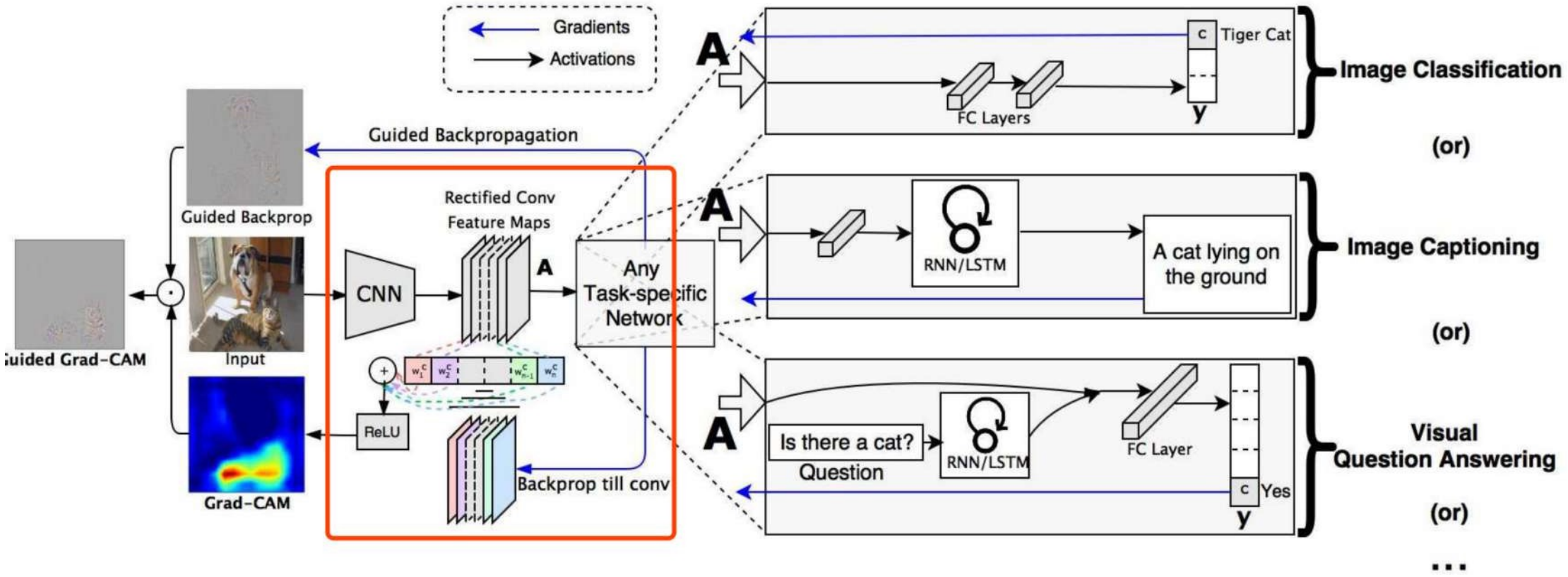
Gradient-weighted Class Activation Mapping (Grad-CAM)

- A method that doesn't need any modification to the existing CNN architecture
- Generalizes CAM to a wide range CNN-based architectures
  - Don't need architectural changes or re-training
- How to find the weights of each feature map?
  - Use the gradient of the target concept c w.r.t. the feature map

Grad-CAM: Motivation

- Deeper representations in a CNN capture higher-level visual constructs
- Convolutional layers retain spatial information, which is lost in fully connected layers
- Grad-CAM uses gradient information flowing from the last layer to understand the importance of each neuron for a decision of interest

Grad-CAM

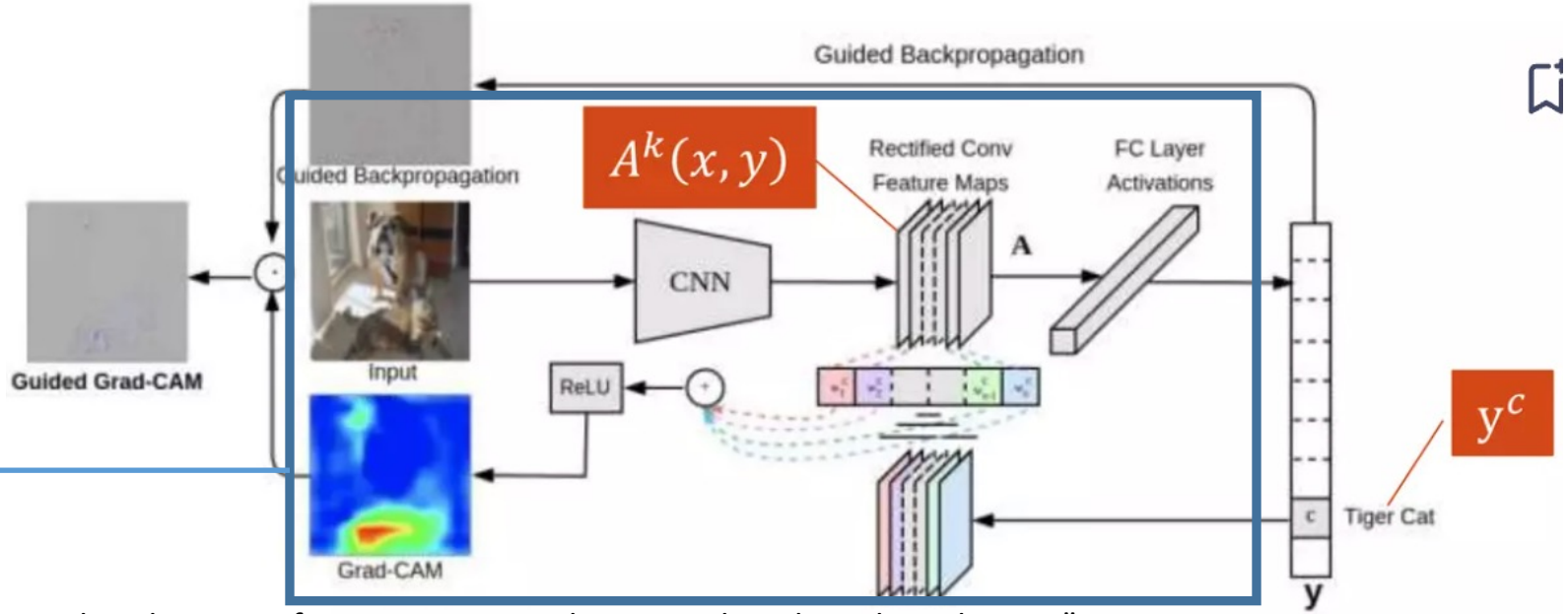


Ramprasaath et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”, ICCV 2017

Grad-CAM

- For a **class** c, compute the gradient of its score y^c (before the softmax), w.r.t. each feature map activations $A^k_{x,y}$ $(k = 1, \dots, n)$
- Infulence of $A^k_{x,y}$ on y^c : $\frac{\partial y^c}{\partial A^k_{x,y}}$
- Define the importance weights of feature map k via GAP: $\alpha^c_k = \frac{1}{Z}\sum_x \sum_y \frac{\partial y^c}{\partial A^k_{x,y}}$

$$ReLU\left(\sum_{k=1}^{n} \alpha^c_k A^k\right)$$



Ramprasaath et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", ICCV 2017

Grad-CAM: How it works

- Compute $\frac{\partial y^c}{\partial A^k}$: gradient of score y^c wrt feature maps A^k
- Global average pool these gradients to obtain neuron importance weights:

$$a_k^c = \frac{1}{z}\sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$

Weight (importance) of kth neuron for predicting class c

- Perform weighted combination of forward activation maps and follow it by ReLU to obtain:

$$L_{Grad-CAM}^c = ReLU(\sum_k a_k^c A^k)$$

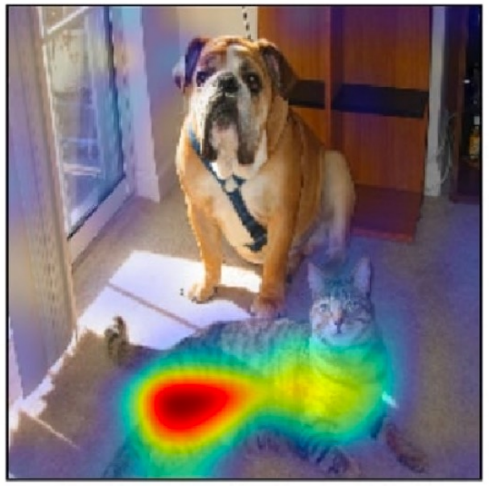
We are only interested in features having positive impact on the score of the target class

Ramprasaath et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", ICCV 2017

GradCAM: Pros and Cons

- Advantages:
  - High class-discriminativity
  - Starting from the final layer allows the network to escape from the noisy low-level processing in the initial layers.

- Shortcomings:
  - Produces coarse (not pixel-level) attribution maps.
  - Starting from the final layer is difficult to justify theoretically.
    - Assumes that neurons keep their spatial identity throughout the network.



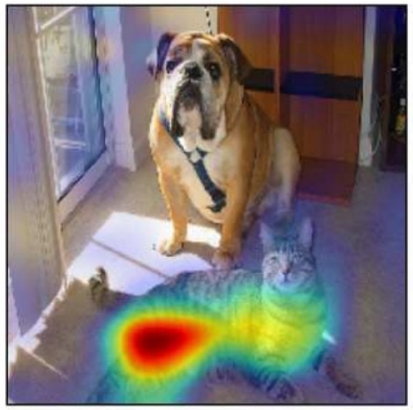
(c) Grad-CAM 'Cat'

(i) Grad-CAM 'Dog'

Ramprasaath et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", ICCV 2017

A Grad-CAM Shortcoming

- Grad-CAM provides good localization, but it lacks fine-grained detail
- In this example, it can easily localize cat
- However, it doesn't explain why the cat is labeled as 'tiger cat'
- Point-wise multiplying guided backpropagation and Grad- CAM visualizations solves the issue



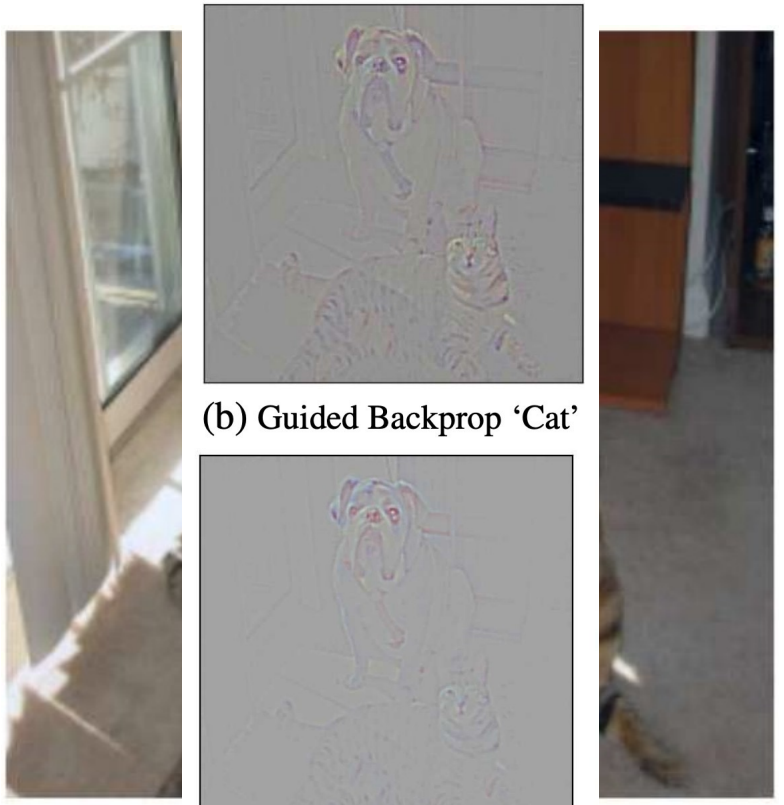
(c) Grad-CAM 'Cat'

(i) Grad-CAM 'Dog'

Ramprasaath et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", ICCV 2017

Good visual explanation

- Class discriminative - localize the category in the image
  - CAM
  - Grad-CAM
- High resolution - capture fine-grained detail
  - Guided Backpropagation
  - Deconvolution



(b) Guided Backprop 'Cat'

(h) Guided Backprop 'Dog'

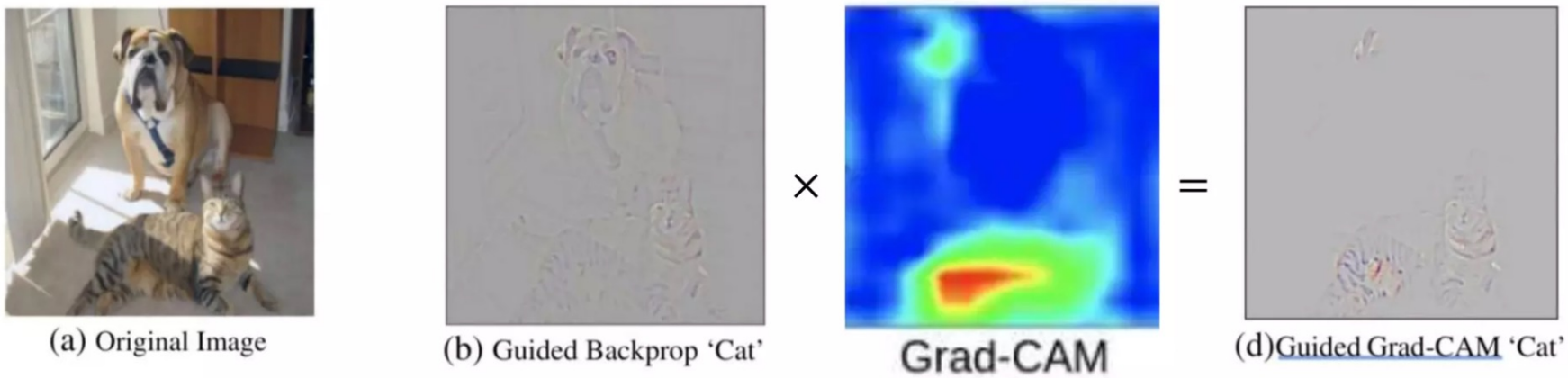
Ramprasaath et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", ICCV 2017

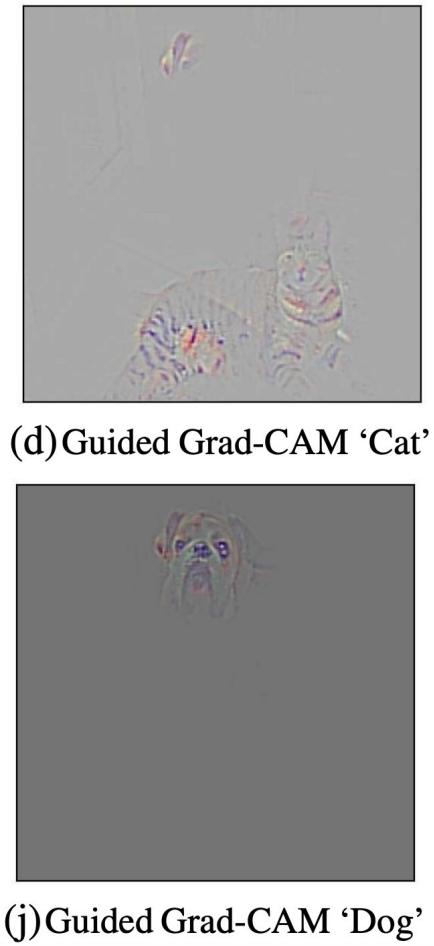
Guided Backpropagation

- It is high resolution
  - Since it derives gradients w.r.t. the input image instead of the last convolutional layer similar to Grad-CAM
- Disadvantages:
  - Not class discriminative

Guided Grad-CAM

- Combines Guided Backpropagation and Grad-CAM to provide a high-resolution class discriminative technique





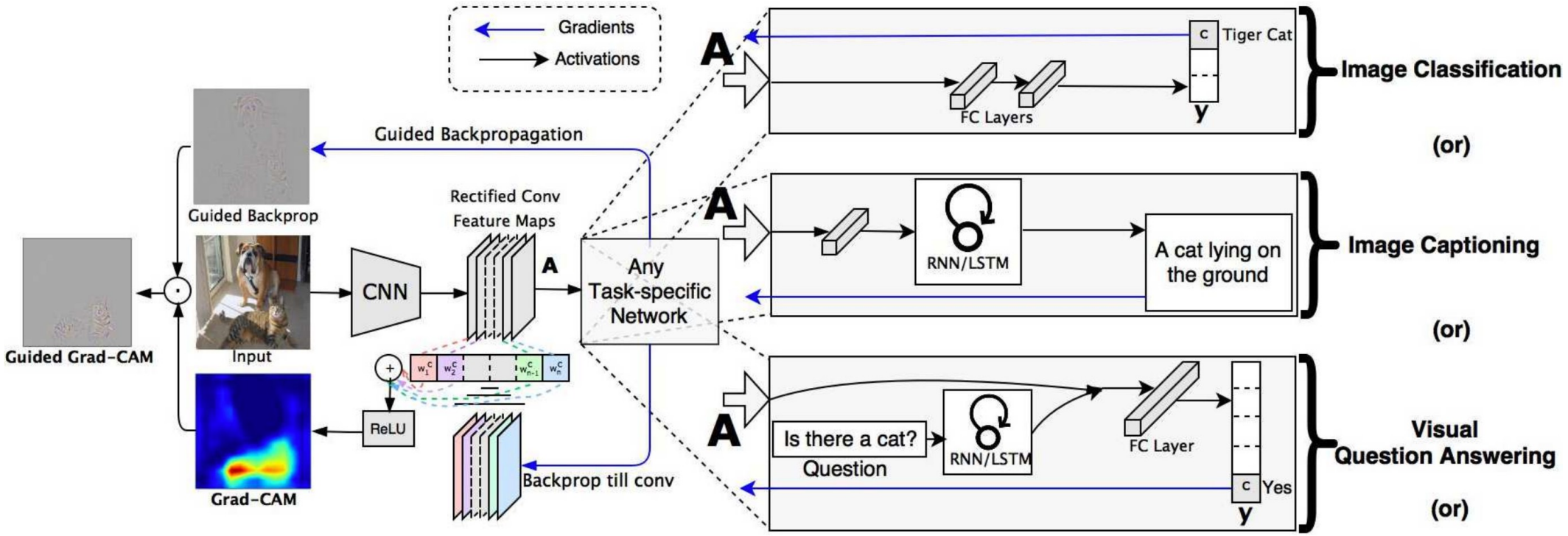
- With Guided Grad-CAM, it becomes easier to see which details went into decision making
  - For example, we can now see the stripes and pointed ears by using the model predicted it as 'tiger cat'

Ramprasaath et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", ICCV 2017

Grad-CAM: How it works

GradCAM in a glance:

$$L^{class}_{GradCAM} = (E_{Channel}[Input_{Last\ Layer} \times E_{Spatial}[\nabla Y^{class}]])^{+}$$



Ramprasaath et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”, ICCV 2017

Evaluations: Localization

- Generate Grad-CAM maps for each of the predicted classes
- Binarize with threshold of 15% of max intensity
- Draw bounding box around single largest connected segment of pixels

Method	Top-1 loc error	Top-5 loc error	Top-1 cls error	Top-5 cls error
Backprop on VGG-16 [40]	61.12	51.46	30.38	10.89
c-MWP on VGG-16 [46]	70.92	63.04	30.38	10.89
Grad-CAM on VGG-16 (ours)	56.51	46.41	30.38	10.89
VGG-16-GAP (CAM) [47]	57.20	45.14	33.40	12.20

Table 1: Classification and Localization on ILSVRC-15 val (lower is better).

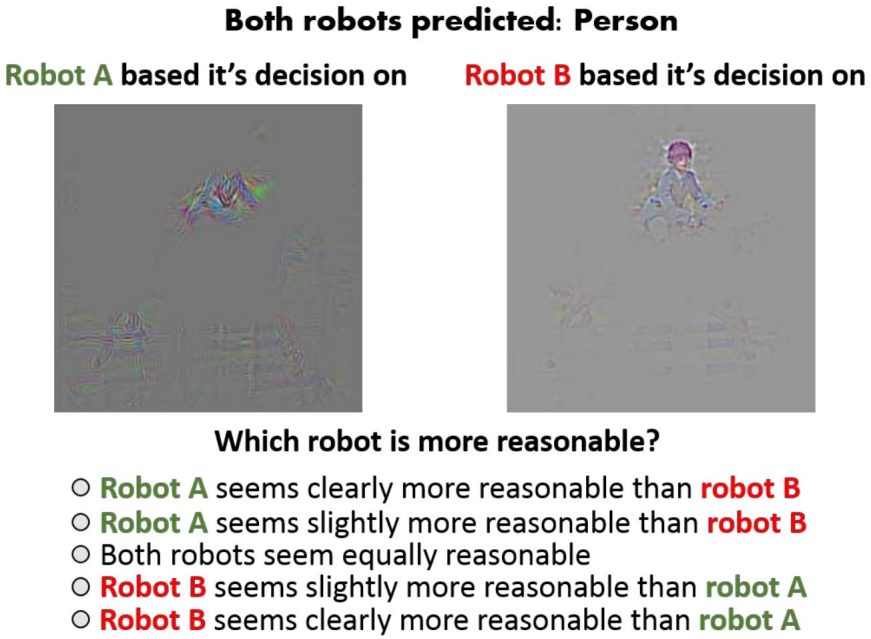
Ramprasaath et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”, ICCV 2017

Evaluations: Trust - Why is it needed?

- Given two models with the same predictions, which model is more trustworthy?
- Visualize the results to see which parts of the image are being used to make the decision!

Evaluations: Trust - Experimental Setup

- Given visualizations from both models, 54 AMT workers were asked were asked to rate reliability of the two models as follows
  - More/less reliable (+/-2)
  - Slightly more/less reliable (+/-1)
  - Equally reliable (0)



Ramprasaath et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", ICCV 2017

Evaluations: Trust - Result

- AlexNet and VGG-16 to compare Guided Backprop and Guided Grad-CAM visualizations
  - Note that VGG-16 is more accurate (79.09mAP vs 69.20)
  - Only those instances considered where both models make same prediction as ground truth

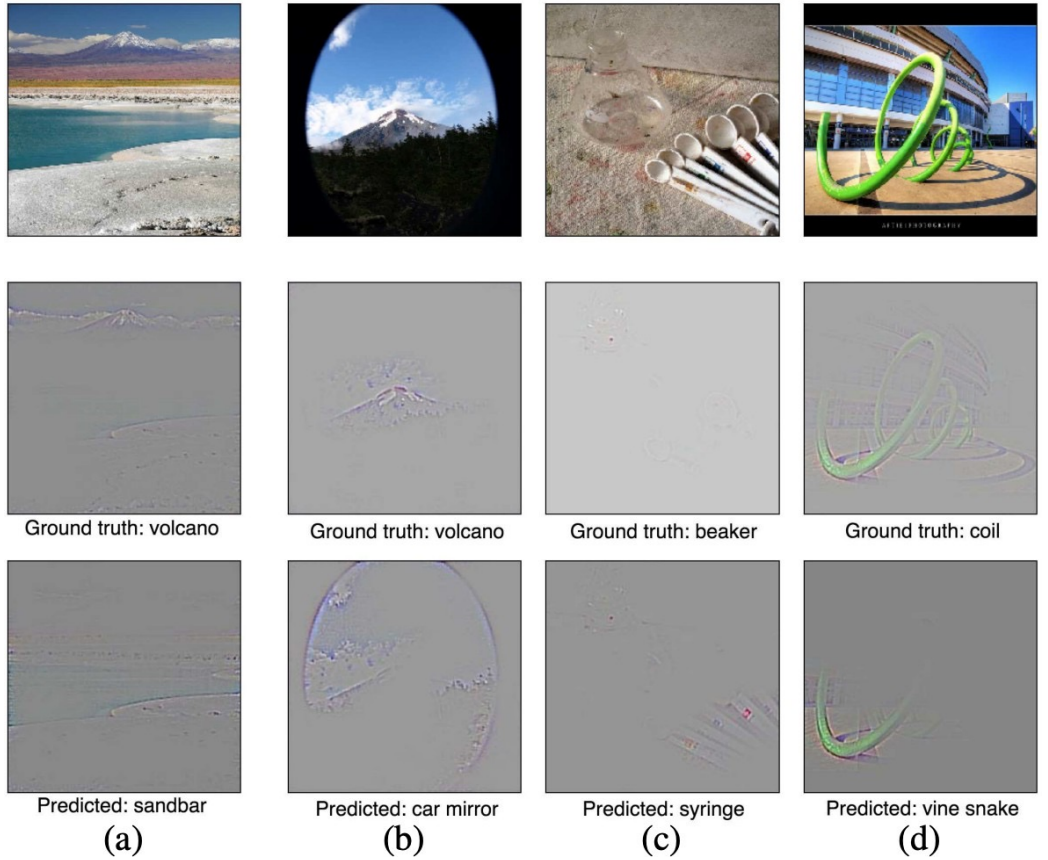
- Humans are able to identify the more accurate classifier, despite identical class predictions
  - With Guided Backpropagation, VGG was assigned a score of 1.0
  - With Guided Grad-CAM, it achieved a higher score of 1.27

- Thus, the visualization can help place trust in a model which will generalize better, just based on individual predictions

Ramprasaath et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”, ICCV 2017

Analyzing Failure Modes for VGG-16

- In order to see what mistakes a network is making, first collect the misclassified examples
- Visualize both the ground truth class as well as the predicted class
- Some failures are due to ambiguities inherent in the dataset
- Seemingly unreasonable predictions have reasonable explanations



Ramprasaath et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”, ICCV 2017

Image Captioning

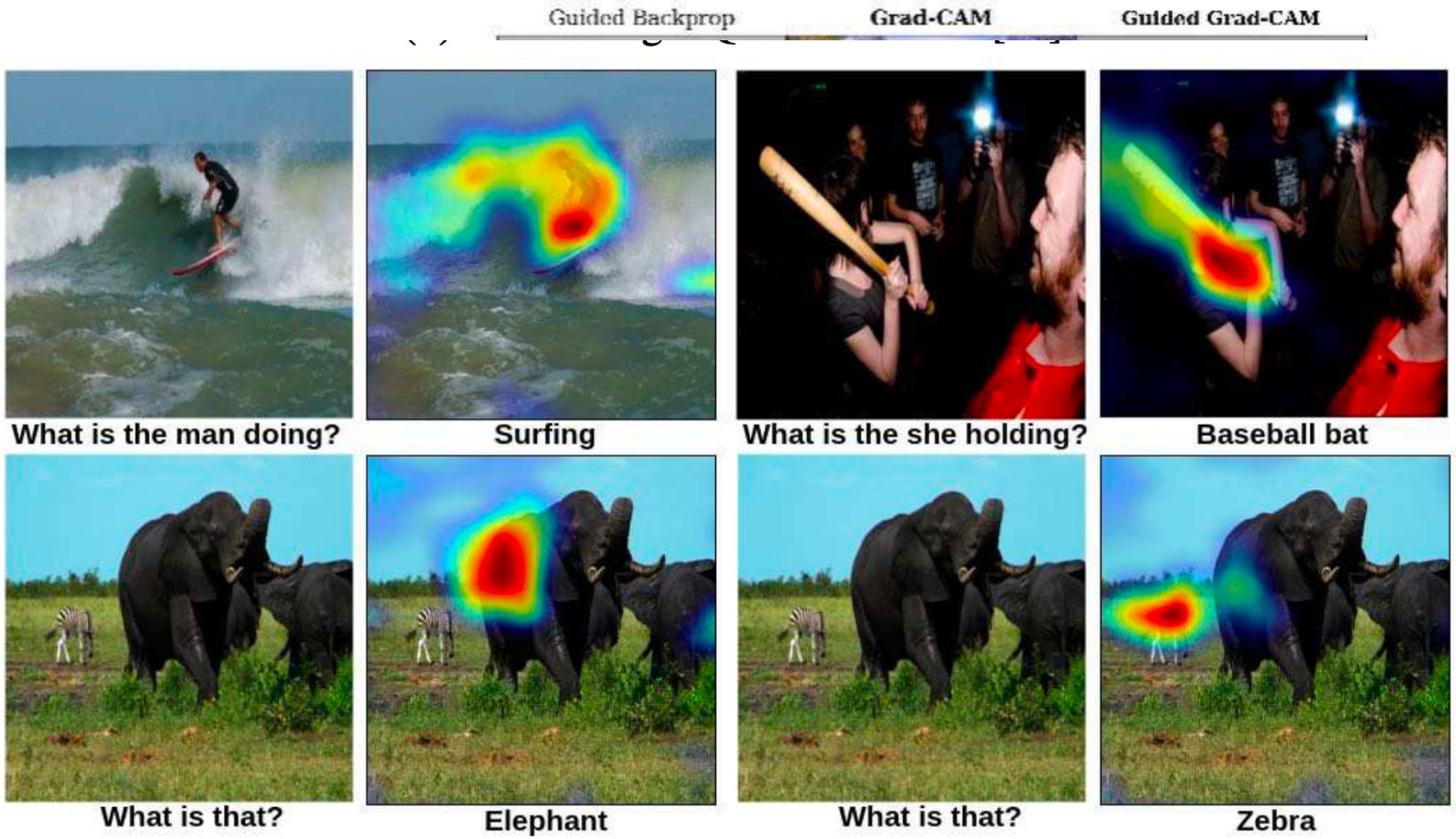
(a) Image captioning explanations

Neuraltalk2: VGG-16 CNN for images and an LSTM-based language model

Given a caption, compute gradient of its log-probability wrt units in the last convolutional layer of the CNN

Ramprasaath et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”, ICCV 2017

Visual Question Answering



(b) Visualizing ResNet based Hierarchical co-attention VQA model from [29]

(a) Visualizing VQA model from [28]

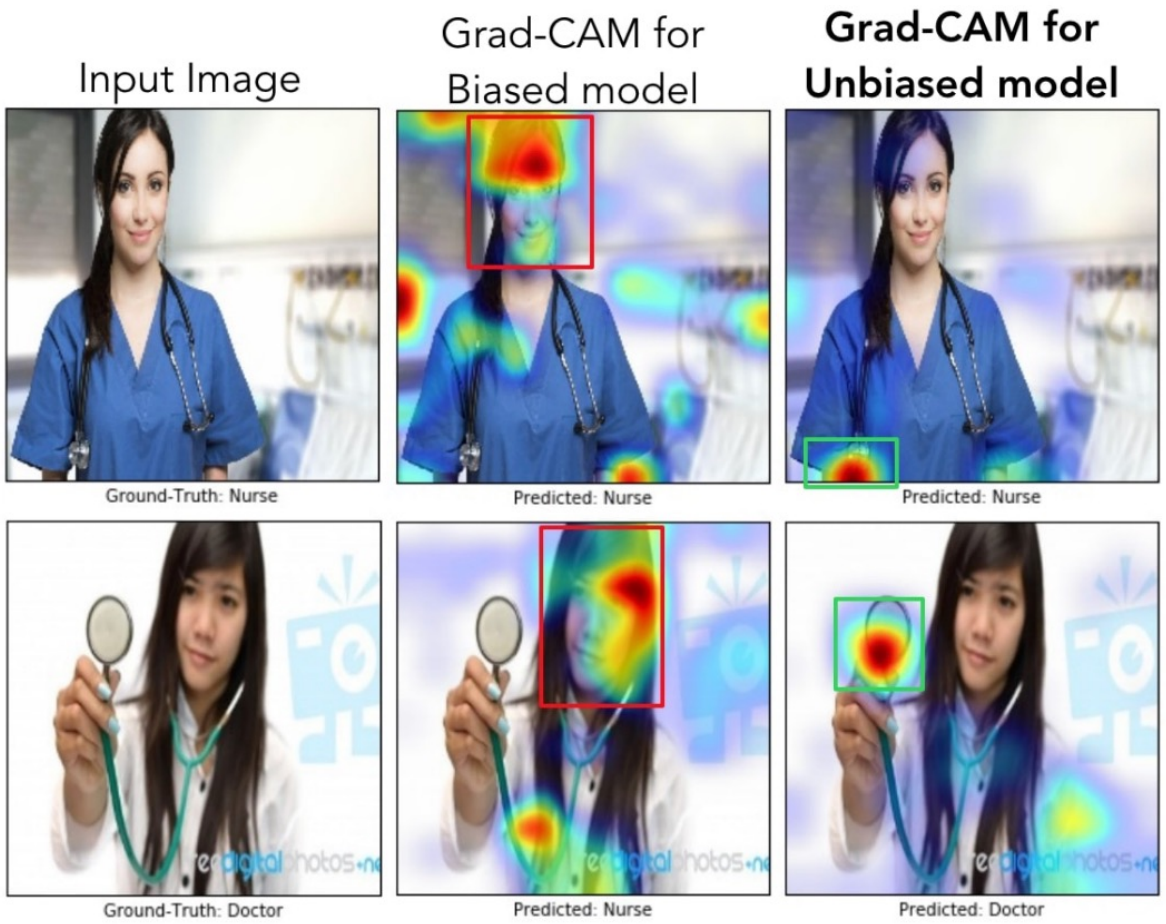
Ramprasaath et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", ICCV 2017

Identifying Bias in Dataset

- Fine-tuned an ImageNet trained VGG-16 model for the task of classifying “Doctors” vs “Nurses”
- Used top 250 relevant images from a popular image search engine
- Trained model achieved good validation accuracy, but didn’t generalize well (82%)
- Visualizations helped to see that the model had learnt to look at the person’s face/hairstyle to make the predictions, thus learning gender stereotypes

Ramprasaath et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”, ICCV 2017

Identifying Bias in Dataset



Ramprasaath et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”, ICCV 2017

Layer-Wise IxG (CAT): The Algorithm

- Definition: Calculates Input×Gradient for each layer in the network, and averages the obtained attribution maps.

- Also known as Class Activation Tokens (CAT).
- For each layer l in the network:
  - Compute the input X(l) and its gradient $\nabla_{\boldsymbol{x}^{(l)}} f_{\boldsymbol{\theta}}(\boldsymbol{x})$
  - Calculate the Layer-wise IxG as: $\boldsymbol{x}^{(l)} \odot \nabla_{\boldsymbol{x}^{(l)}} f_{\boldsymbol{\theta}}(\boldsymbol{x})$
- It might be beneficial to skip the first few layers.
  - skip the very first layer which is equivalent to normal Input×Gradient.

Yao Qiang et al. “AttCAT: Explaining Transformers via Attentive Class Activation Tokens”, NeurIPS 2022

Layer-Wise IxG (CAT): Pros and Cons

- Advantages: ViT
  - SOTA in identifying the patches most contributing to the counterfactual of the target class

Evaluations: Faithfulness vs Interpretability

- Faithfulness of a visualization to a model is defined as its ability to explain the function learned by the model
- There exists a trade-off between faithfulness and interpretability
- A fully faithful explanation is the entire description of the model, which would make it not interpretable/easy to visualize

Evaluations: Faithfulness vs Interpretability

- Explanations should be locally accurate
- For reference explanation, one choice is image occlusion
- CNN scores are measured when patches of the input image are masked
- Patches which change CNN scores are also patches which are assigned high intensity by Grad-CAM and Guided Grad-CAM
- Rank correlation of 0.261 achieved over 2510 images in PASCAL 2007 val set

Insertion: Area Over Perturbation Curve (AOPC)

- Features are inserted into an empty image in order of their importance in the attribution map of the target class.
- The more faithful method is, the faster the probability of the target class should increase.
- We denote the change in the probability of the target class with the AOPC metric:

$$\mathrm{AOPC}(k) = \frac{1}{N}\sum_{i=1}^{N}\left(p(y_i|x_i) - p\left(y_i\middle|\tilde{x}_i^k\right)\right)$$

  - $\tilde{x}_i^k$ is an input where the top k features are present.
  - y_i is the target class.
- Lower AOPC indicates higher faithfulness.

Insertion: Accuracy

- Similarly, if we define the accuracy of the model M as its output matching the target class, we have the following metric:

$$Accuracy(k) = \frac{1}{N}\sum_{i=1}^{N} I\left(M\left(\tilde{x}_i^k\right) = y_i\right)$$

- Higher accuracy indicate higher faithfulness.

Deletion Metrics

- Features are removed from the image in order of their importance.
- The more faithful the attribution method is, the faster the probability of the target class should decrease.
- The measurement of deletion metrics is similar to the insertion metrics but the direction for better is reversed;
- Higher AOPC and lower accuracy indicate higher faithfulness.
- Deletion metrics are more accepted in the literature

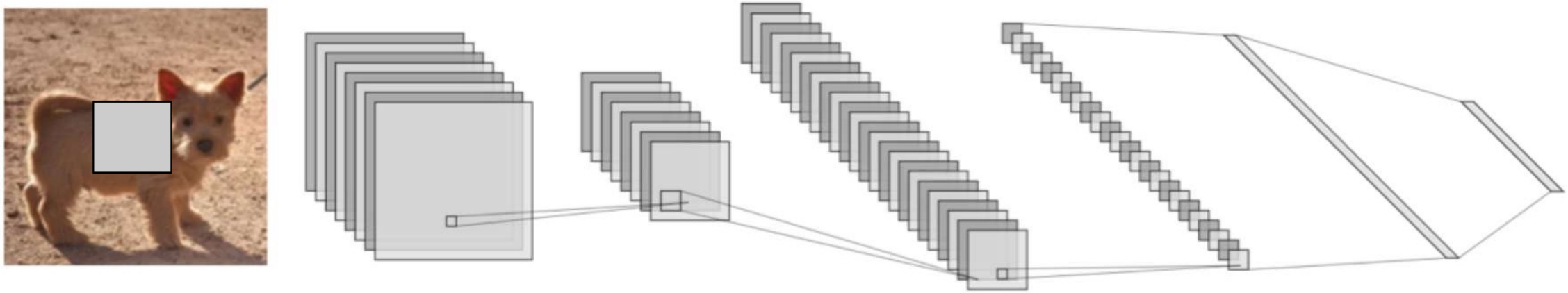
Black-Box Methods

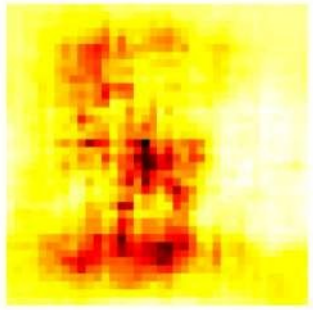
- Black-box methods try to optimize the faithfulness metrics directly.

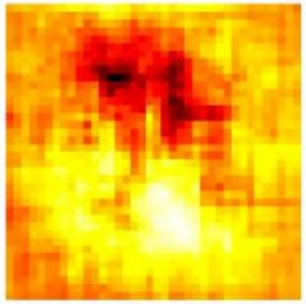
- They usually need to evaluate the model at many inputs, which makes them expensive.

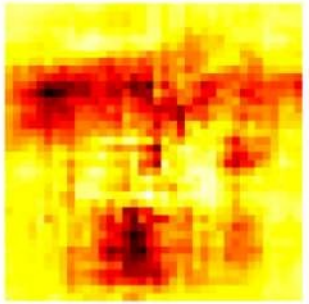
- These two reasons make them unsuitable for an apples-to-apples comparison with white-box methods.
  - Mask-Based: They mask some features and assign importance scores based on the changes observed in the target output.
    - ViT-CX [22]
  - LIME: trains a linear model to predict the effects of masking some features [14]

Mask Based Methods





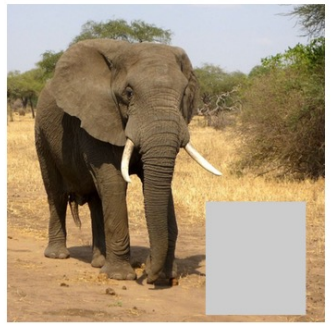


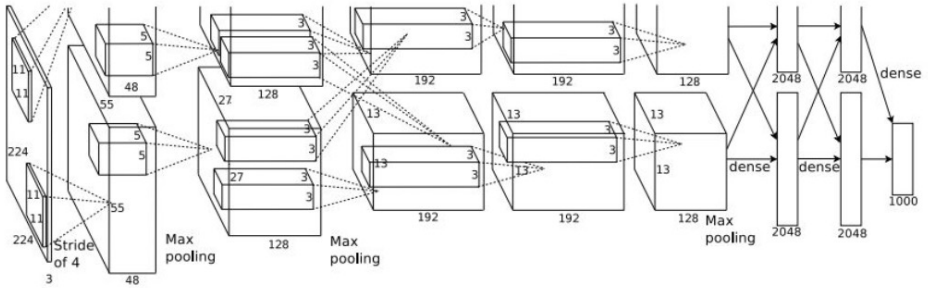


Zeiler and Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014

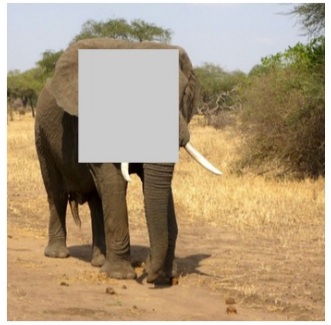
Which pixels matter: Occlusion

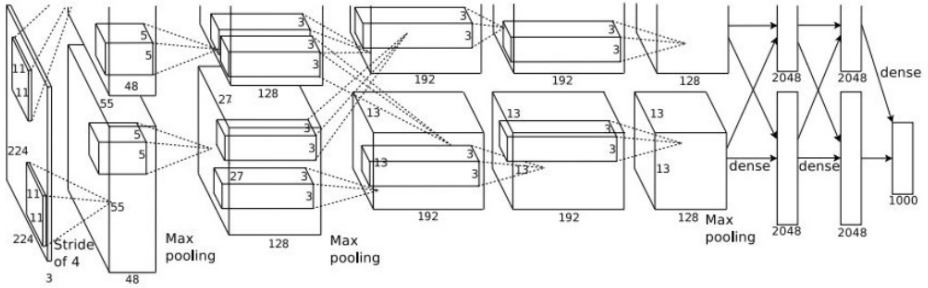
Mask part of the image before feeding to CNN,
check how much predicted probabilities change





P(elephant) = 0.95





P(elephant) = 0.75

Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014

Local Interpretable Model-Agnostic Explanations (LIME)

- It is surrogate-based explanation technique
  - A linear surrogate model is used to explain the prediction of the target model
  - Approximate the target model in the neighborhood of the interested input

- It does not directly explain the prediction of the target model, but rather the predictions of a surrogate model

Local Interpretable Model-Agnostic Explanations (LIME)

1. generates samples in the neighborhood of the interested input
   - Find their corresponding output using the target model

2. fitting a linear model to these samples
   - approximates the target model in this local vicinity by a simple linear function

uses local linear approximations to help understand how black box models function internally

– learning a sparse linear model for a given prediction

Interpretability Approaches

- **Visualization**
  - **Feature Interpretability**
    - visualizing the features that the model has learned to understand what it is learning
  - **Activation Maximization**
    - maximizing the activation of a neuron or layer to understand what the model is learning

- **Attribution (Saliency Maps)**
  - which parts of the input are most important

- **Interpretable Models**

Types of interpretations

- An Interpretable model could clearly show the reasons for decision
- Model-based interpretability vs. post-hoc interpretability
- Accuracy-interpretability trade off

Desiderata in model-based interpretability

- Sparsity
- Simulatability
- Modularity

Summary

- Feature interpretability
- Attribution methods
  - Gradient
  - Input ×Gradient
  - Guided Backpropagation
  - CAM
  - GradCAM
- Evaluation Metrics
- Black-box methods