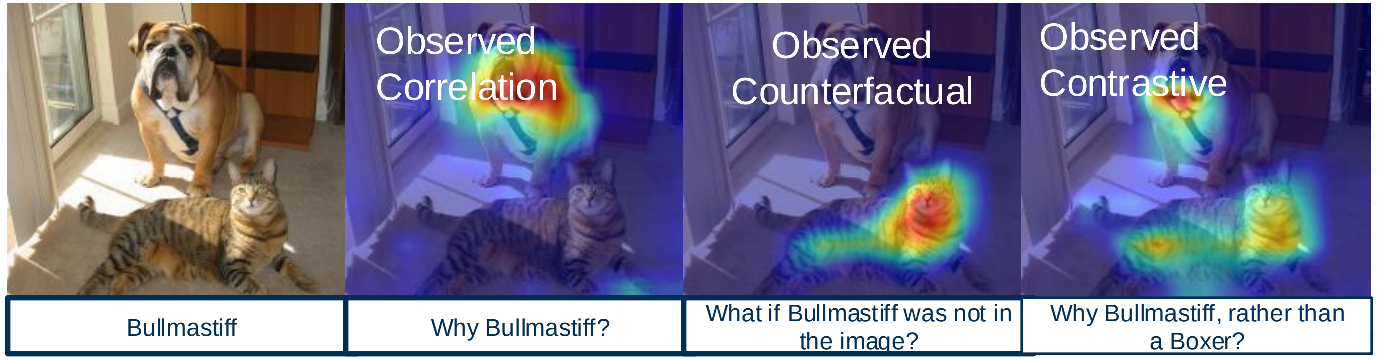
ECE 4252/8803: Fundamentals of Machine Learning (FunML)
Fall 2024

Lecture 22: Explainability in Neural Networks



OLIVES @GeorgiaTech
Georgia Tech

Overview

In this Lecture..

Explainability

- Definition
- Why Explainability

Visualization of Convolutional Neural Networks

Types of Explanations

Explanatory Evaluation

What is Explainability?

The ability of an entity to explain or justify its decisions or predictions in human-understandable terms

Explainability

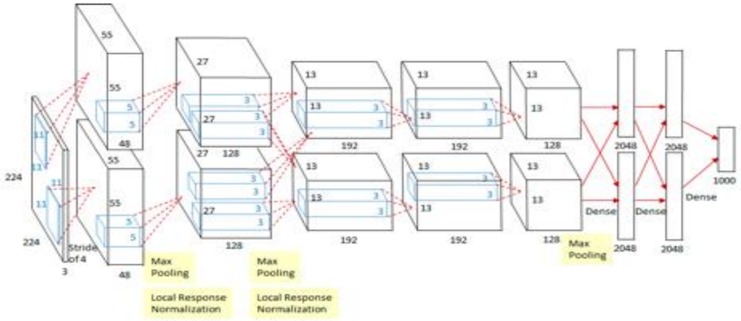
Why Explainability?

Explainability matters:

establish **trust** in deep learning systems by developing *transparent* models that can explain *why they predict what they predict* **to humans**

Algorithm

Data



Output

class scores

Explainability is useful in:

- Medical: help doctors diagnose
- Seismic: help interpreters label seismic data
- Autonomous Systems: build appropriate trust and confidence

Deep models act as algorithms that take data and output something **without** being able to **explain** their methodology

Overview

In this Lecture..

Explainability

Visualization of Convolutional Neural Networks

- Explainability in CNNs
- Visualizing filters
- Dimensionality Reduction with Last layer Embeddings
- Visualizing activations
- Gradient-based visualizations
- Saliency Maps and Intermediate feature visualization
- Grad-CAM visualization and explanations

Types of Explanations

Explanatory Evaluation

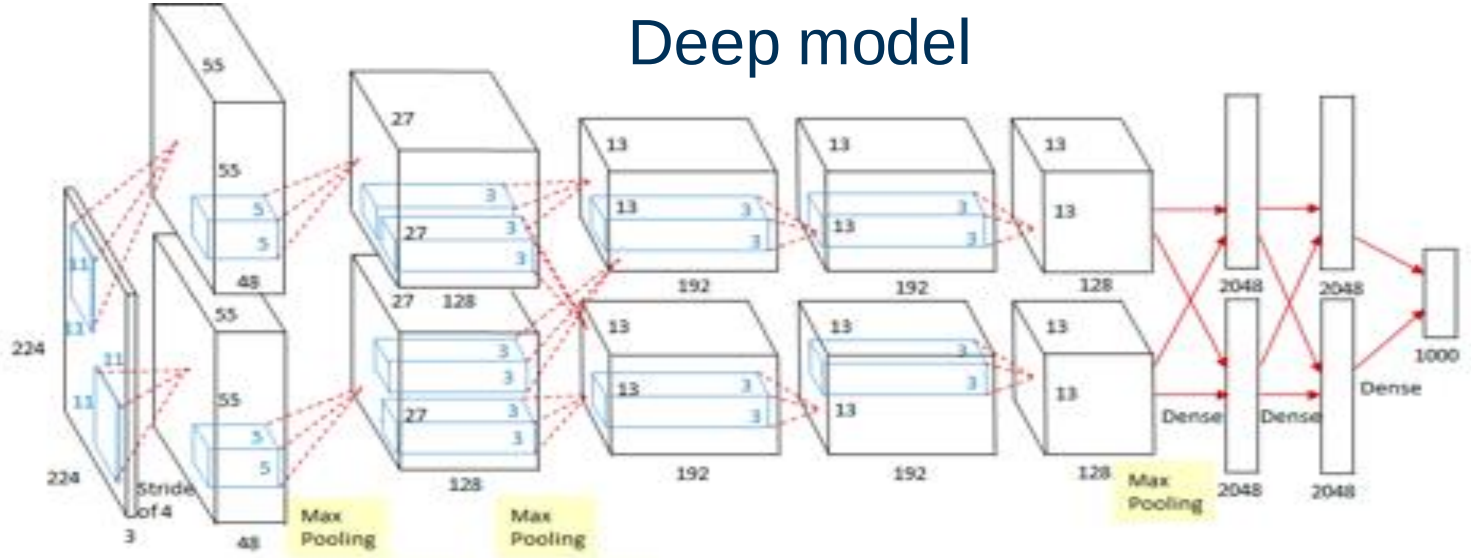
Explainability

Explainability in CNNs

Data

Input Image:
3 x 224 x 224

Deep model



https://medium.com/coinmonks/paper-review-of-alexnet-caffenet-winner-in-ilsvrc-2012-image-classification-b93598314160

Output

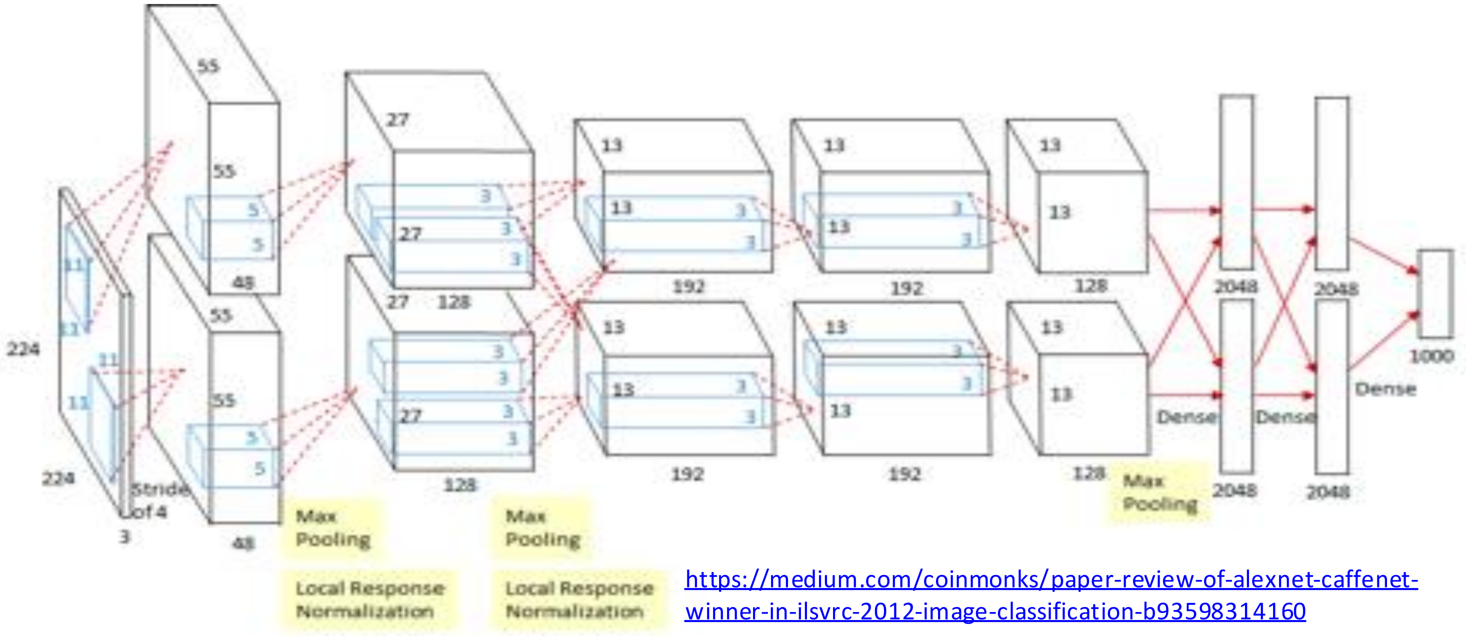
Output class scores:
1000 numbers
(trained on ImageNet)

Not Explainable

Krizhevsky et al. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012)

Explainability
Explainability in CNNs

Input Image:
3 x 224 x 224



https://medium.com/coinmonks/paper-review-of-alexnet-caffenet-winner-in-ilsvrc-2012-image-classification-b93598314160

Output class scores:
1000 numbers
(trained on ImageNet)

What are these layers looking for?
Are they explainable?

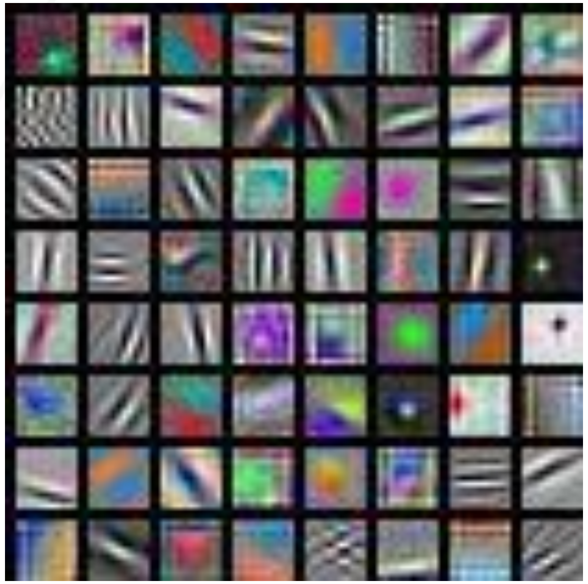
Material adapted from : Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Krizhevsky et al. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012)

Visualizing CNNs

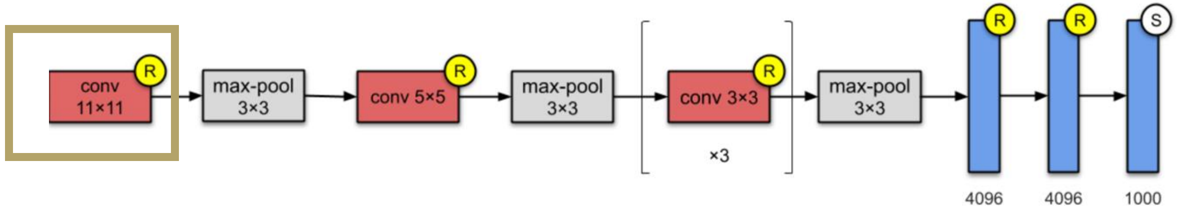
Visualizing Filters in First Layers

Filters = Weights



AlexNet:
64 x 3 x 11 x 11

Input Image:
3 x 224 x 224



AlexNet

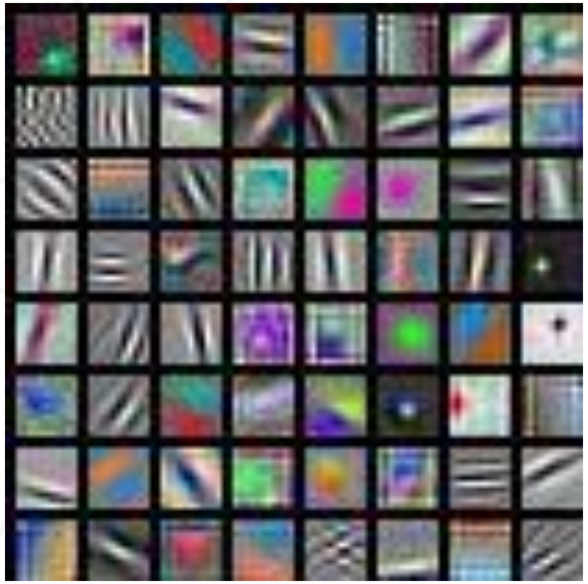
Filters always extend the full depth of the input volume

- 64 filters in the first convolutional layer
- Filter size: 11 x 11 x 3 (visualize as RGB images)
- Filters are looking for ***low-level*** **oriented edges, color blobs, textures, background etc.**

Material adapted from : Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

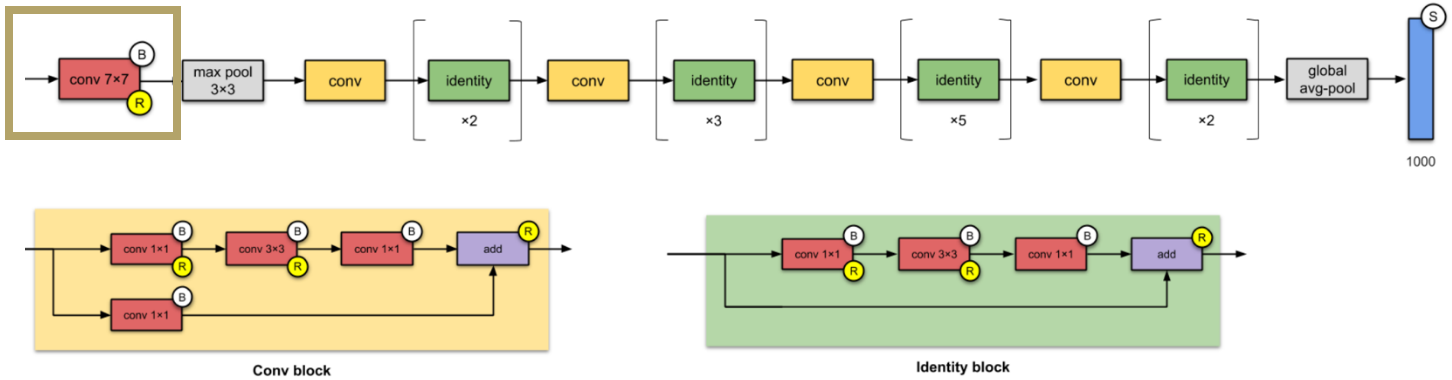
Krizhevsky et al. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012)

Visualizing CNNs
Visualizing Filters in First Layers



AlexNet:
64 x 3 x 11 x 11

ResNet-18:
64 x 3 x 7 x 7



ResNet

- Filters in the first convolutional layers **across different architectures learn similar patterns**

Material adapted from : Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Krizhevsky et al. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012)

OLIVES @GeorgiaTech
Georgia Tech

Visualizing CNNs

Visualizing Filters in Intermediate Layers

Visualize the filters
(raw weights)

Conv layer 1 weights
16 x 3 x 7 x 7
(visualize as RGB images)

Conv layer 2 weights
20 x 16 x 7 x 7
(visualize as 16 grayscale images)

Filters in **higher** convolutional **layers are not as interpretable as** filters in **the first layer**

Conv layer 3 weights
20 x 20 x 7 x 7
(visualize as 20 grayscale images)

Material adapted from : Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Visualizing CNNs

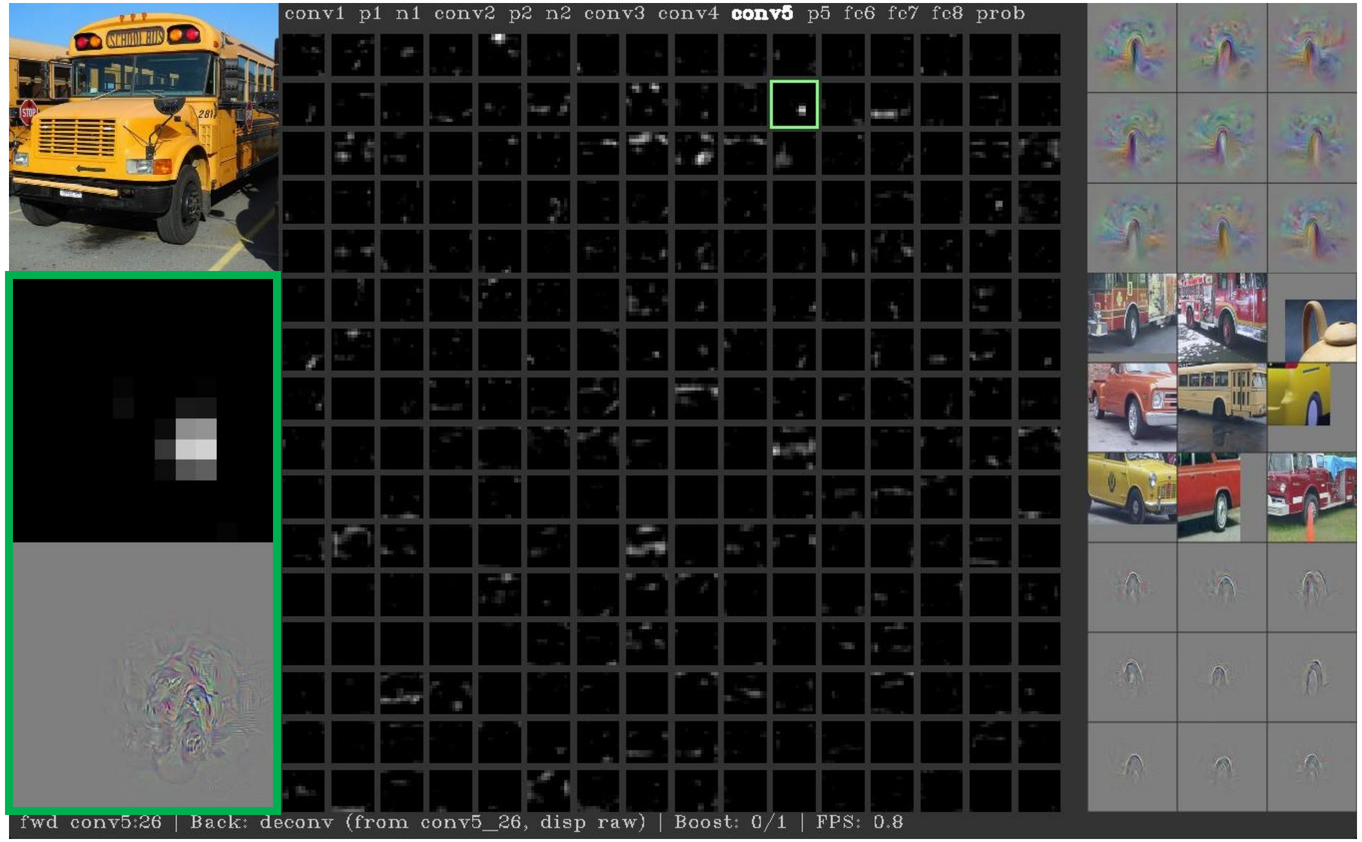
Visualizing Filters in Intermediate Layers

Intermediate layers:

- Weights: not very interpretable
- Activations: interpretable

The filter in conv 5 layer is activated when it sees a wheel

Conv 5



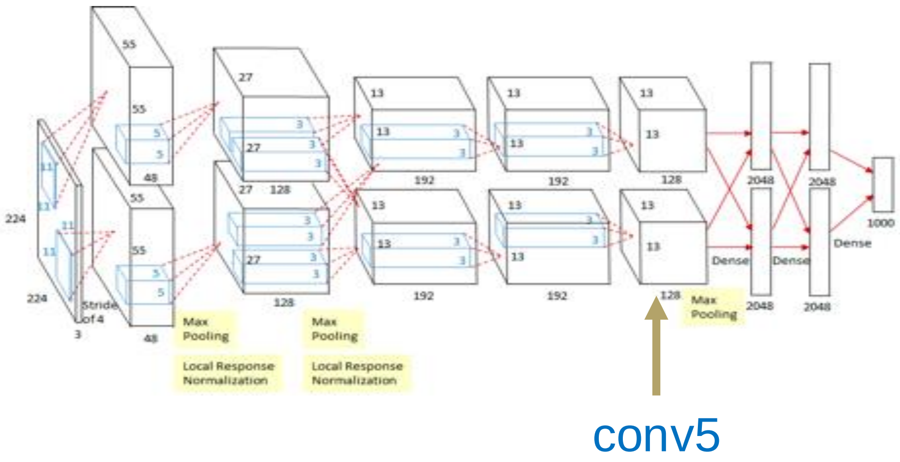
AlexNet

Material adapted from : Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

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

Visualizing CNNs
Visualizing Filters in Intermediate Layers

- Apart from visualizing the intermediate activations, we can also visualize what **similar visual patterns** in images that cause the maximum activations of certain neurons



- Maximally Activating Patches:
    - Image patches in the input that cause the maximum activations of certain filters

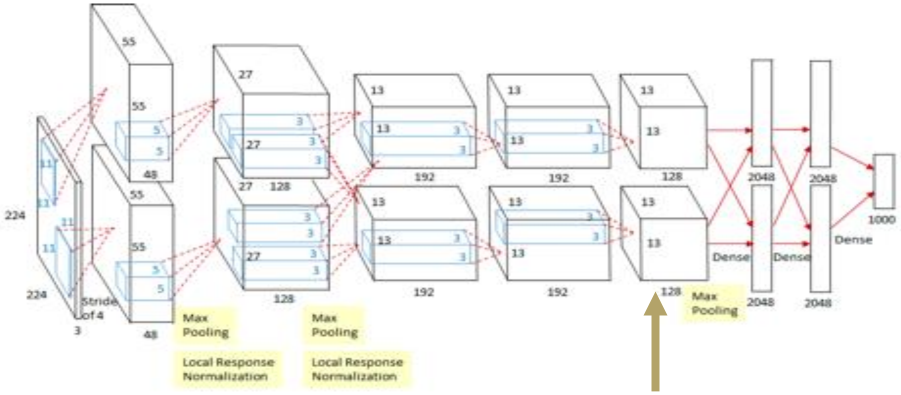
Material adapted from : Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

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

Visualizing CNNs

Visualizing Maximally Activating Patches

- Maximally Activating Patches:
  - Image patches in the input that cause the maximum activations of certain filters
- Obtaining Maximally Activating Patches:
  - Pick activations in a layer, e.g., conv5 (128 x 13 x 13), pick one of the channels 17/128
  - Feed forward many images through the network, record values of the chosen channel
  - Visualize image patches that correspond to **maximal activation**
- Maximally activating patches share **similar visual patterns**



Maximally activating patches
Each row corresponds to a particular neuron in conv5

Material adapted from : Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

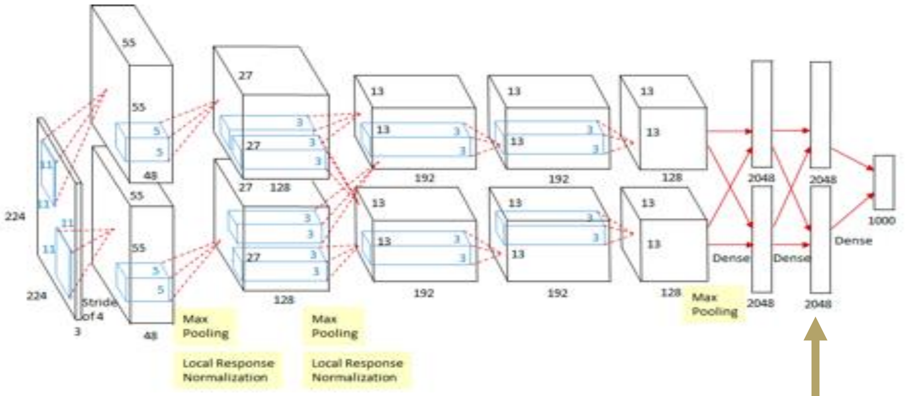
Springenberg et al, “Striving for Simplicity: The All Convolutional Net”, *ICLR* Workshop 2015

Visualizing CNNs

Visualizing Last Layer Activations

- We can group the images that have similar class-specific information by exploring last layer activations

- Last layer activations (embedding):
  - 4096-dimensional feature vector for an image (layer immediately before the classifier)
  - Representations of *entire input images* instead of specific patches
  - Similar embeddings correspond to same classes of input images

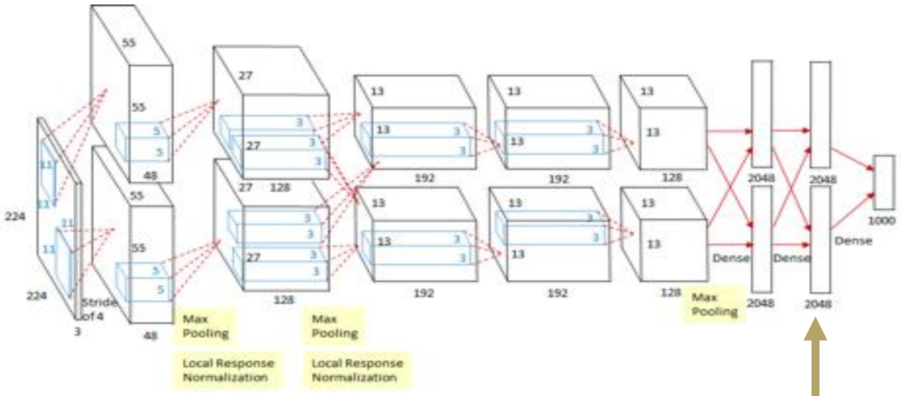


AlexNet fully-connected 7

Visualizing CNNs

Visualizing Last Layer Activations

- Last layer activations (embedding):
  - 4096-dimensional feature vector for an image (layer immediately before the classifier)
  - Representations of *entire input images* instead of specific patches
  - Similar embeddings correspond to same classes of input images
- Feed forward many images through the network, collect the final layer feature vectors
- Visualize input images that have similar last layer embeddings



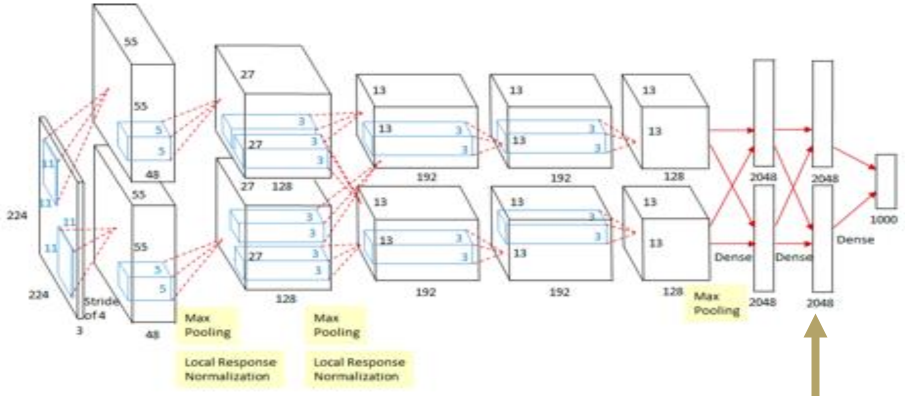
AlexNet

fully-connected 7

Material adapted from : Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Visualizing CNNs

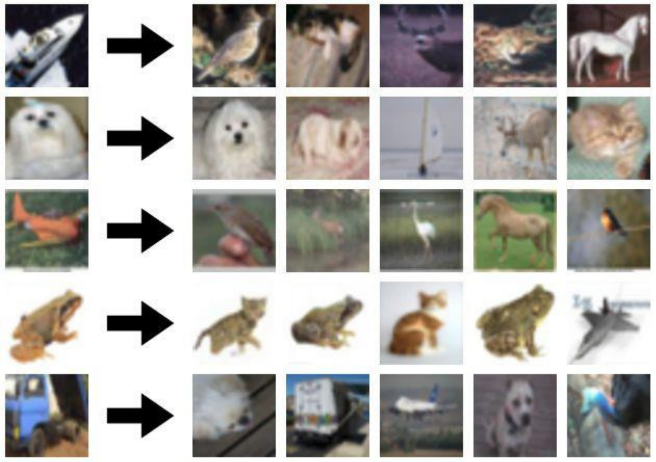
Visualizing Last Layer Activations



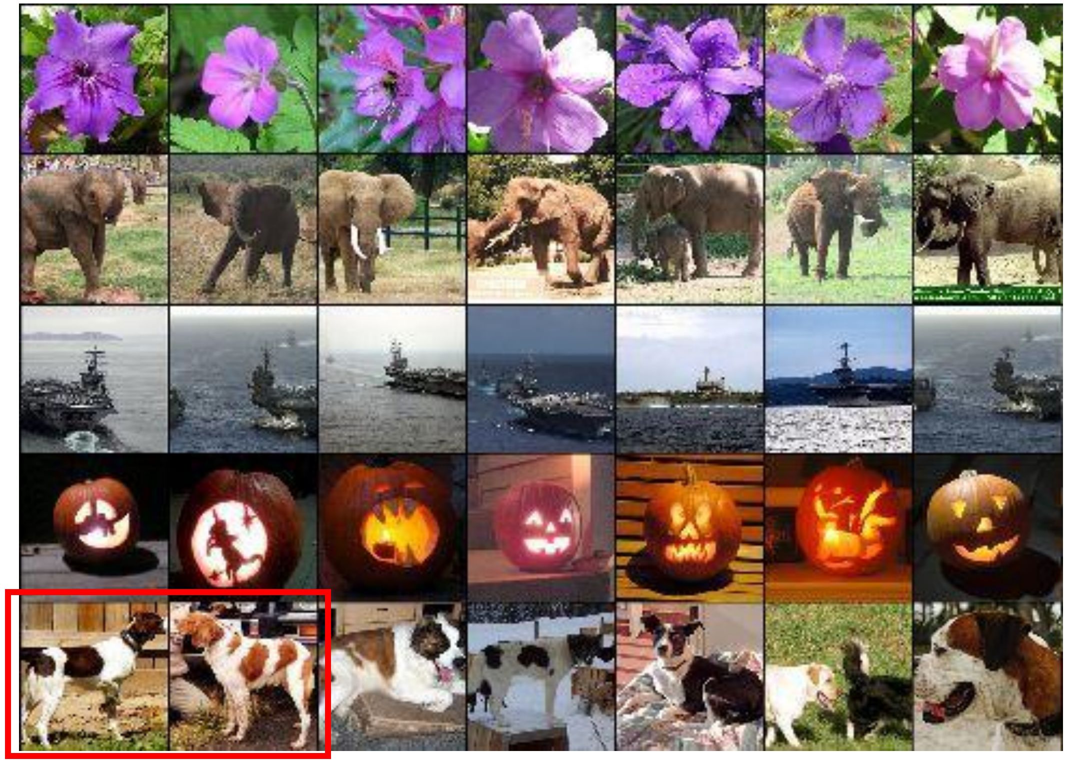
fully-connected 7

- Last layer embedding:
  - 4096-dimensional feature vector for an image

Recall: Nearest neighbors in **pixel** space



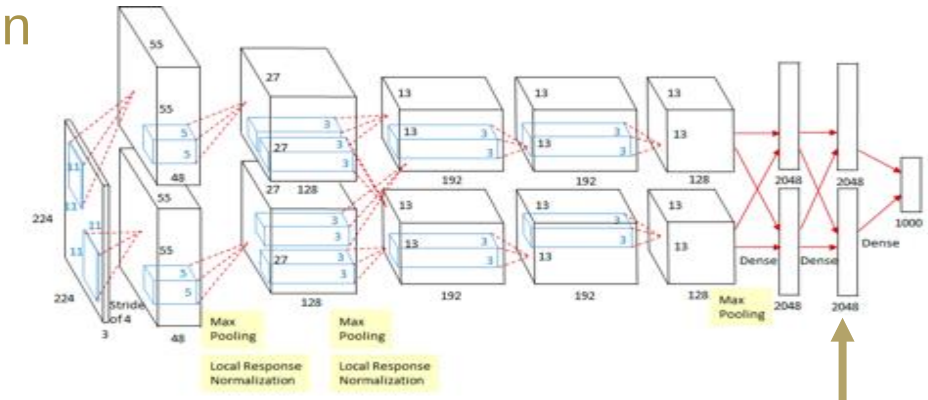
Test image L2 Nearest neighbors in **feature** space



- The **features** of the two dogs **share L2 similarity**
- The images do not share L2 similarity due to horizontally flipped poses

Visualizing CNNs

Visualizing Last Layer Activations via Dimensionality Reduction



fully-connected 7

- Last layer embedding:
  - 4096-dimensional feature vector for an image

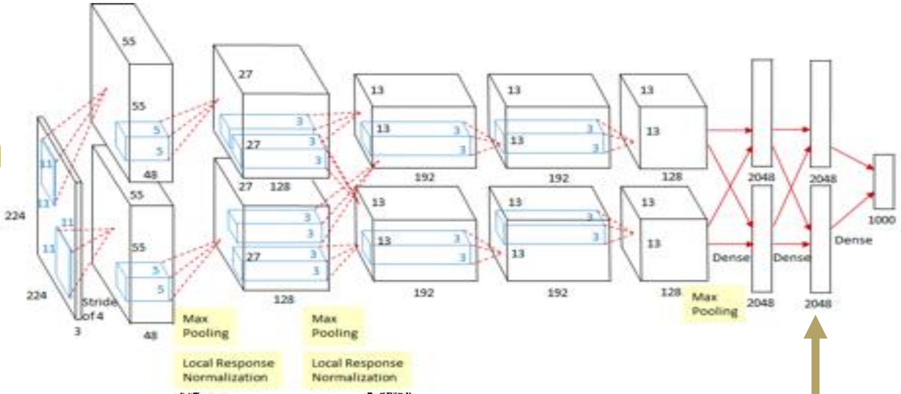
- Visualize the “feature space” by reducing dimensionality of feature vectors from 4096 to 2 dimensions
  - Done via dimensionality reduction techniques – PCA, Isomap, UMAP, tSNE etc.
  - Each 2-dim feature correspond to an input image

Material adapted from : Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Van der Maaten and Hinton, “Visualizing Data using t-SNE”, JMLR 2008

Visualizing CNNs

Visualizing Last Layer Activations via Dimensionality Reduction

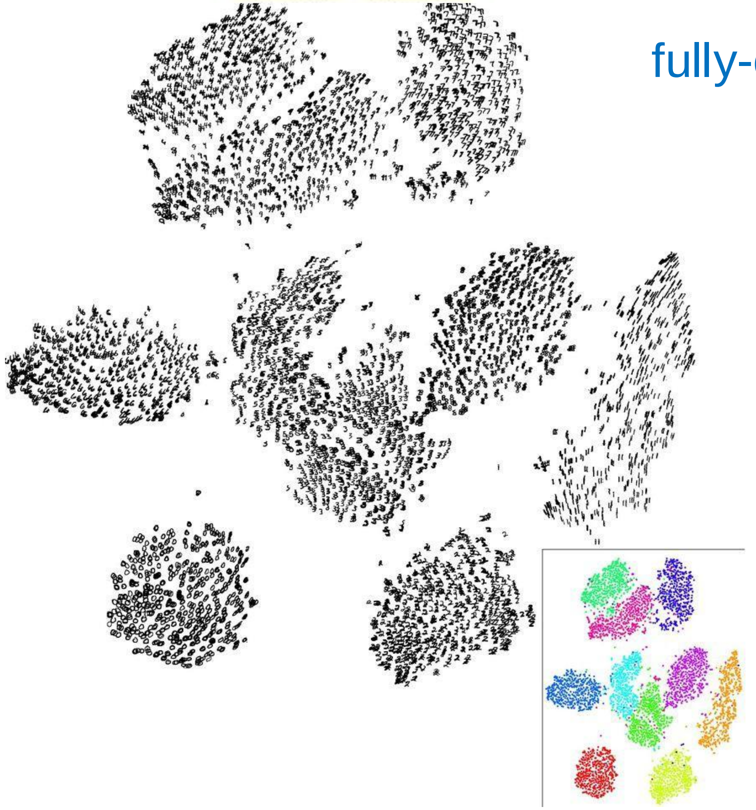


fully-connected 7

- Last layer embedding:
  - 4096-dimensional feature vector for an image

- Dimensionality reduction using t-SNE
  (t-distributed stochastic neighbor embedding):

- Embed ***high-dimensional data*** points so that **locally, pairwise distances are conserved** i.e., similar things end up in clusters, while dissimilar things end up wherever



t-SNE visualization of MNIST

Material adapted from : Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

OLIVES @GeorgiaTech

Georgia Tech

Van der Maaten and Hinton, “Visualizing Data using t-SNE”, JMLR 2008

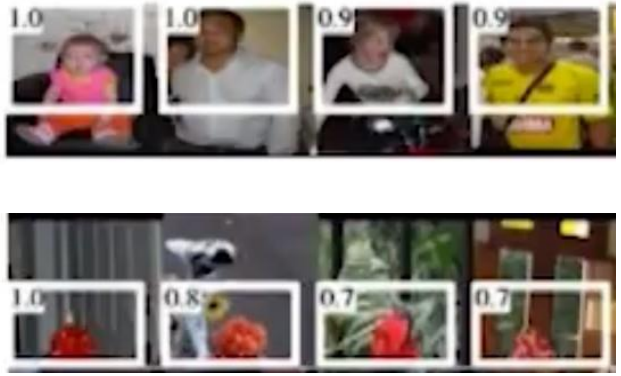
Visualizing CNNs
Summary

We have been visualizing:

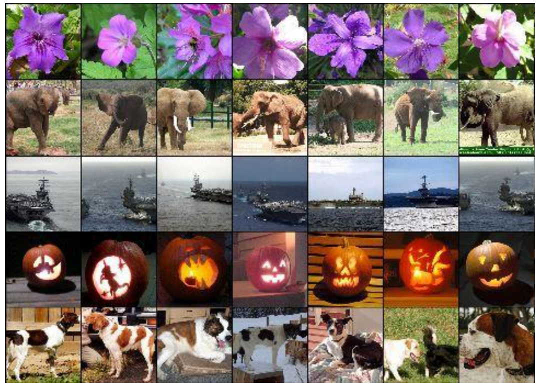
- Weights (filters) in conv layers
- Maximally activating patches
- Nearest neighbor images in features space
- Last layer embeddings

Next, we will look into how pixels affect model decisions

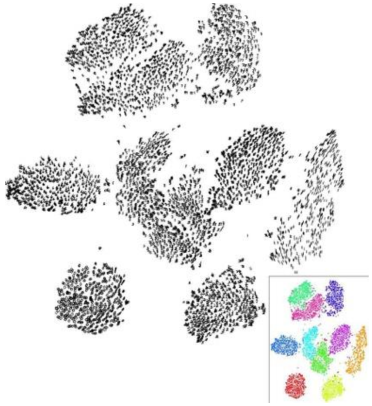
filters



Maximally activating patches



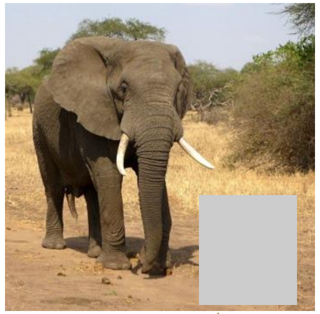
Nearest neighbor images

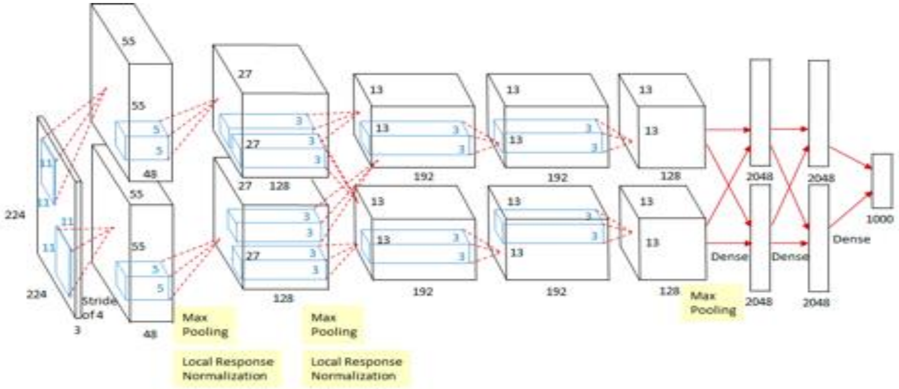


Last layer embeddings

Saliency Methods
Saliency via Occlusion

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





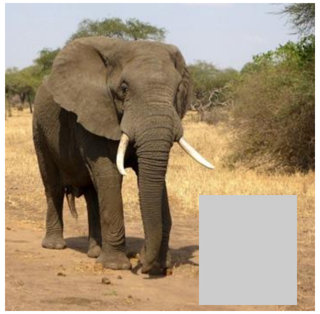
P(elephant) = 0.95

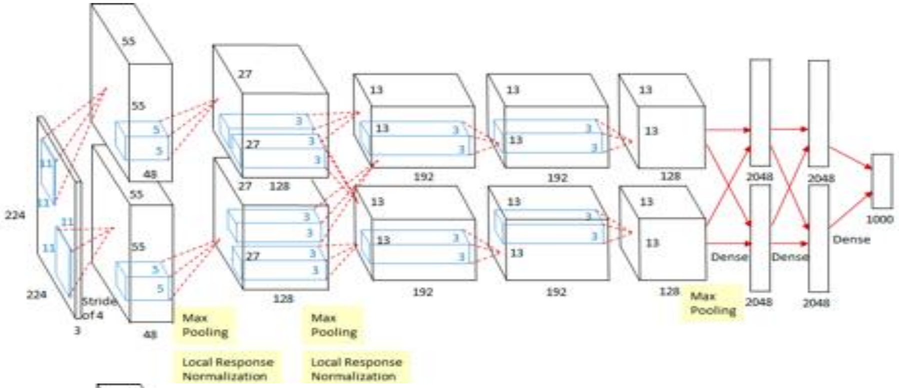
A gray patch or patch of average pixel value of the dataset
Note: not a black patch because the input images are centered to zero in the preprocessing.

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

Saliency Methods
Saliency via Occlusion

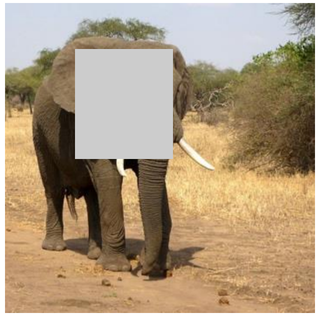
- **Intervention**: Mask part of the image before feeding to CNN, check how much predicted probabilities change

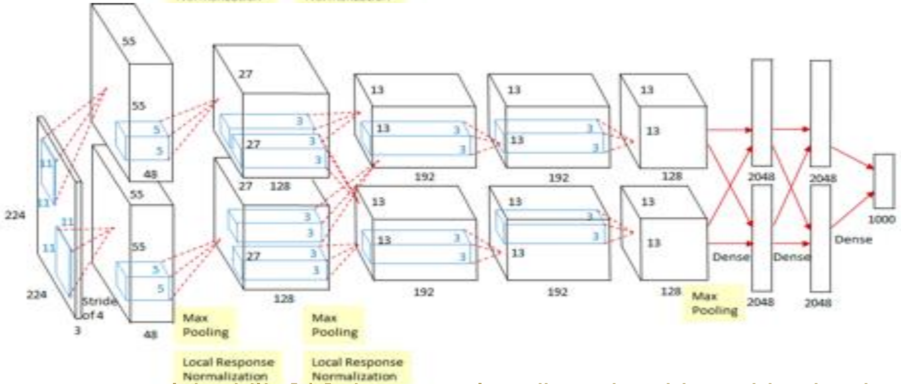




P(elephant) = 0.95

These pixels affect decisions more





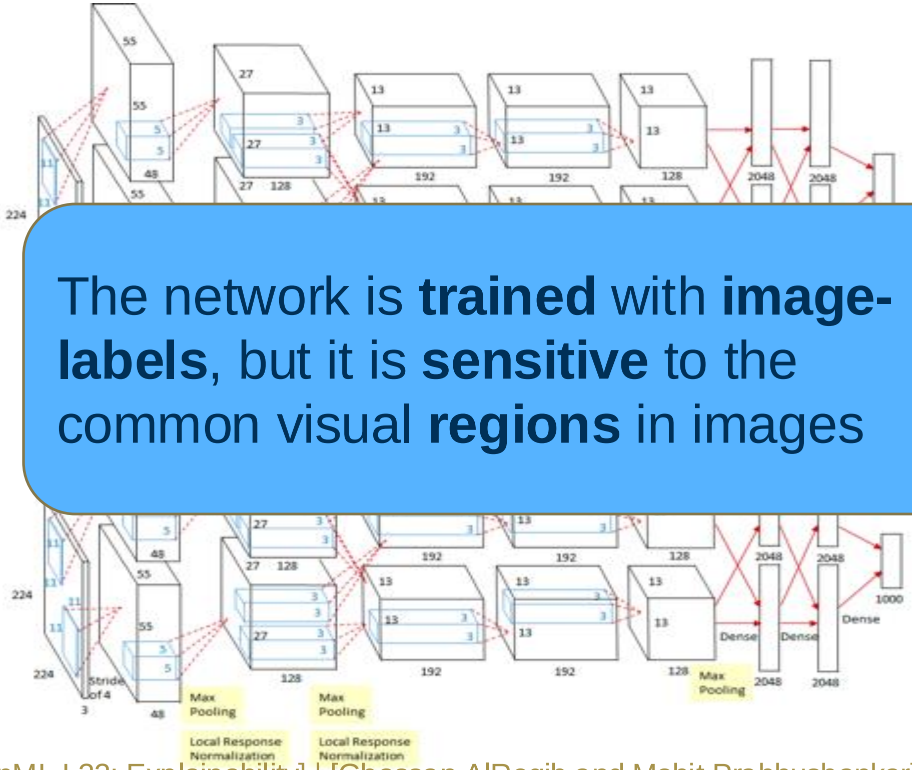
P(elephant) = 0.75

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

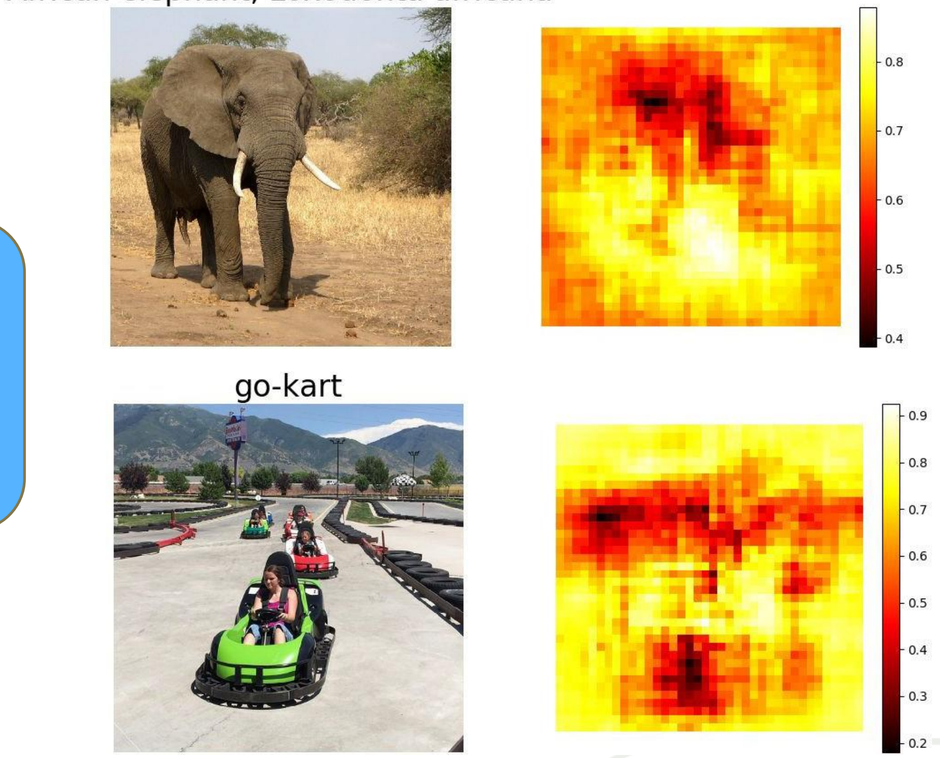
Saliency Methods
Saliency via Occlusion

- Visualizing the heatmap of pixels that cause decrease in probabilities when blocked

These pixels affect decisions more



The network is **trained** with **image-labels**, but it is **sensitive** to the common visual **regions** in images



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

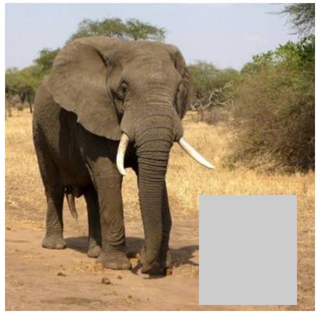
Saliency Methods
Saliency via Occlusion

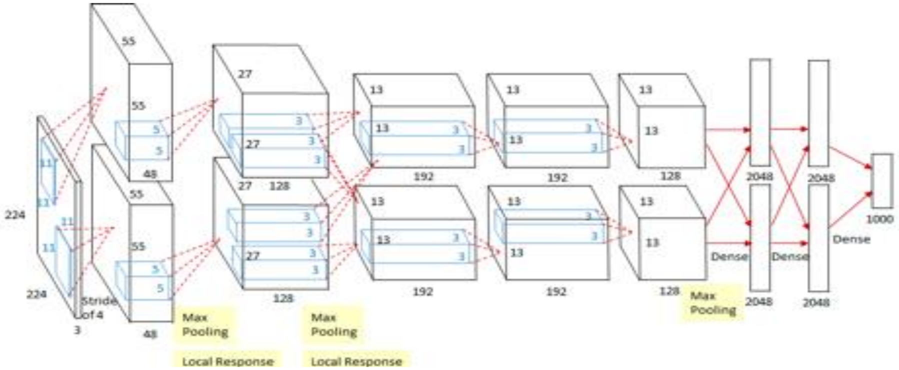
- Visualizing the heatmap of pixels that cause decrease in probabilities when blocked

Very expensive

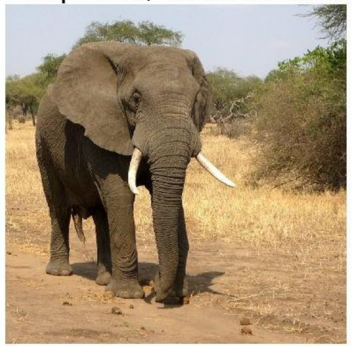
Faithful

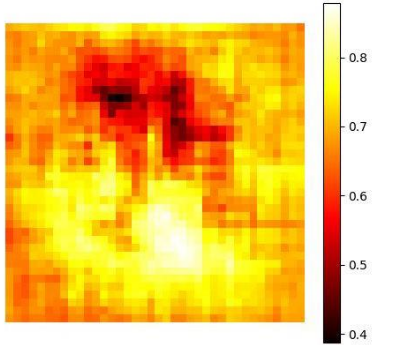
These pixels affect decisions more



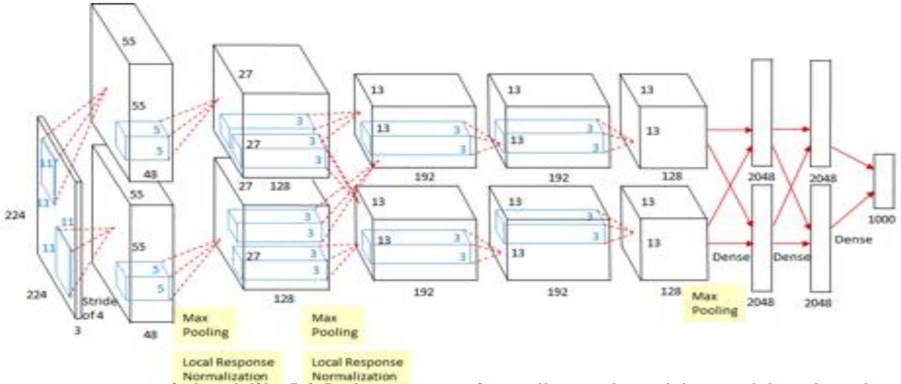


African elephant, Loxodonta africana

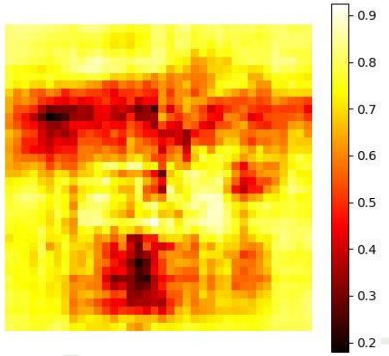




go-kart



go-kart



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

- Saliency via Occlusion:
  - a very expensive approach especially for high resolution input images
- We want to find a less expensive approach
- Recall **feature importance** in logistic regression:

$$P(y = 1|\boldsymbol{x}) = \sigma(\boldsymbol{w}^T\boldsymbol{x} + b)$$
$$P(y = 0|\boldsymbol{x}) = 1 - \sigma(\boldsymbol{w}^T\boldsymbol{x} + b)$$

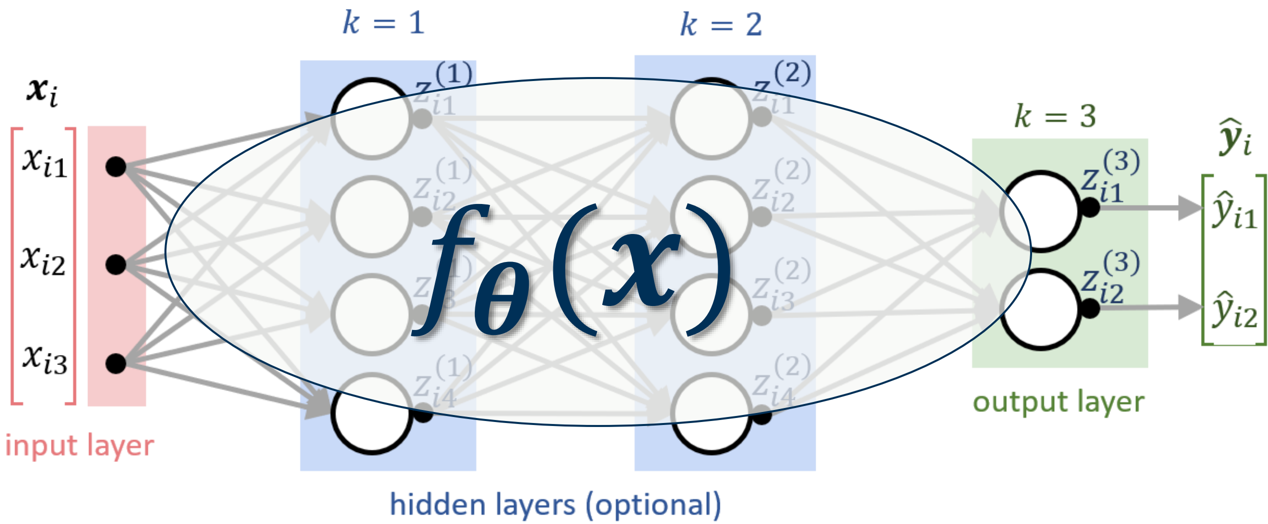
  - for each feature x_i, a **weight** w_i represents its **importance**
- We want to generate pixel saliency maps by deep models as feature importance maps

Saliency Methods

Saliency via Gradients

- Generate pixel saliency maps by deep models as feature importance maps
- *Highly non-linear mapping* function $f_{\boldsymbol{\theta}}: \mathcal{X} \rightarrow \mathcal{Y}$:

$$\widehat{\boldsymbol{Y}} = \varphi\left(\varphi\left(\varphi\left(\boldsymbol{X}\left(\boldsymbol{W}^{(1)}\right)^T + \boldsymbol{o}(\boldsymbol{b}^{(1)})^T\right)\left(\boldsymbol{W}^{(2)}\right)^T + \boldsymbol{o}(\boldsymbol{b}^{(2)})^T\right)\left(\boldsymbol{W}^{(3)}\right)^T + \boldsymbol{o}(\boldsymbol{b}^{(3)})^T\right)$$



- Assume that we can 'linearize' the model using Taylor series

$$\widehat{\boldsymbol{Y}} \approx \boldsymbol{X}(\boldsymbol{W})^T + \boldsymbol{o}(\boldsymbol{b})^T$$

$$f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots,$$

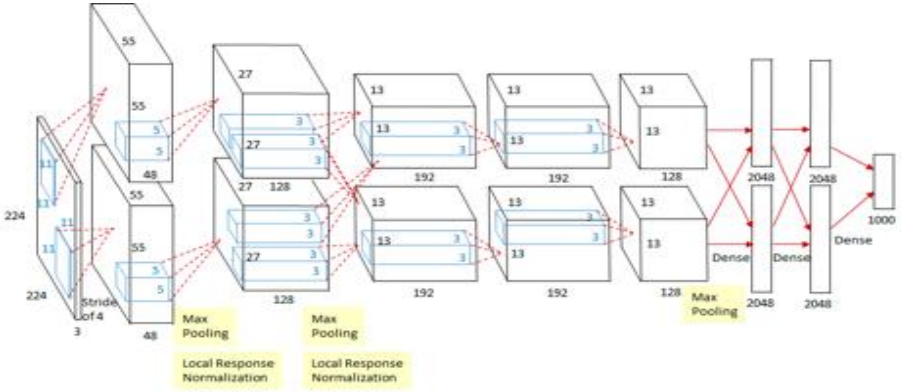
$$\boldsymbol{W} \approx \frac{\partial \widehat{\boldsymbol{Y}}}{\partial \boldsymbol{X}}$$

saliency approximated by **gradients w.r.t. input**, which can be **obtained via backpropagation**

Saliency Methods

Saliency via Gradients

Forward pass: Compute probabilities



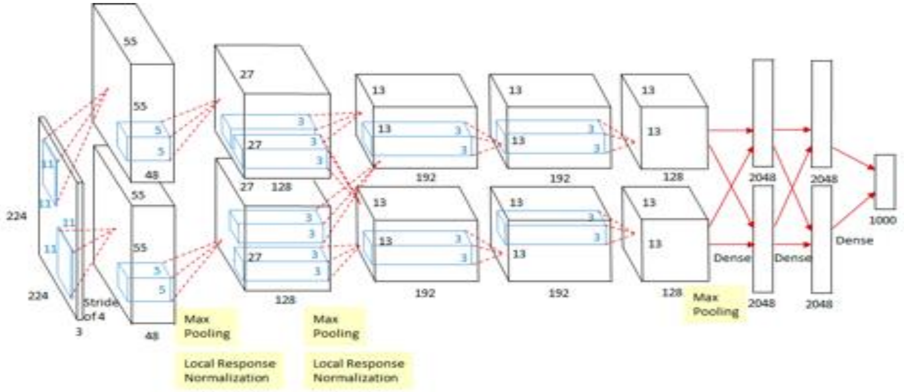
$$\hat{\boldsymbol{y}} = \begin{bmatrix} 0.05 \\ 0.10 \\ \mathbf{0.85} \end{bmatrix}$$ Dog

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

Saliency Methods

Saliency via Gradients

Forward pass: Compute probabilities

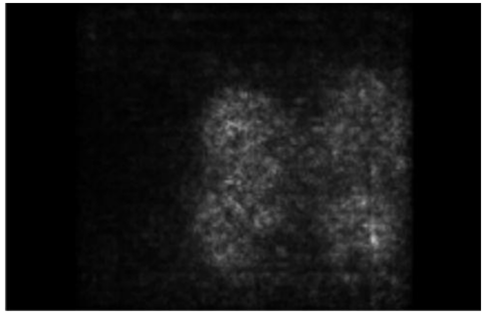


$$\hat{\boldsymbol{y}} = \begin{bmatrix} 0.05 \\ 0.10 \\ \mathbf{0.85} \end{bmatrix}$$

$\hat{y}_c = 0.85$ (dog)

Backward pass: Compute gradients

$$\frac{\partial \hat{y}_c}{\partial X}$$



Saliency map

Compute **gradient** of (unnormalized) class score **with respect to image pixels**

Then visualize the max of absolute value over RGB channels

OLIVES @GeorgiaTech

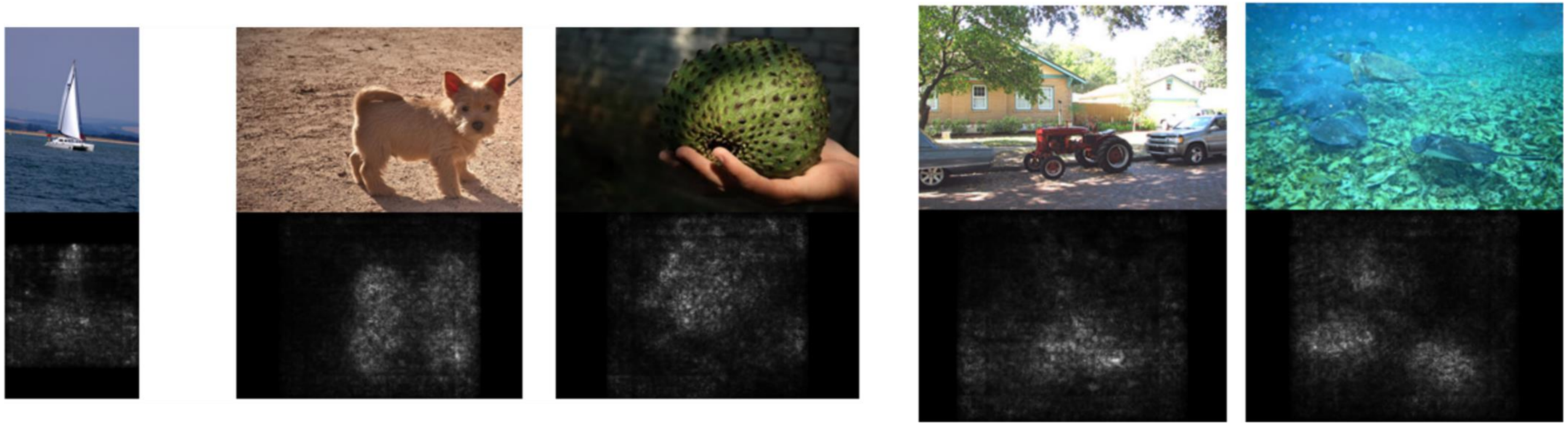
Georgia Tech

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

Saliency Methods

Saliency via Gradients

- Examples of saliency maps via backpropagation



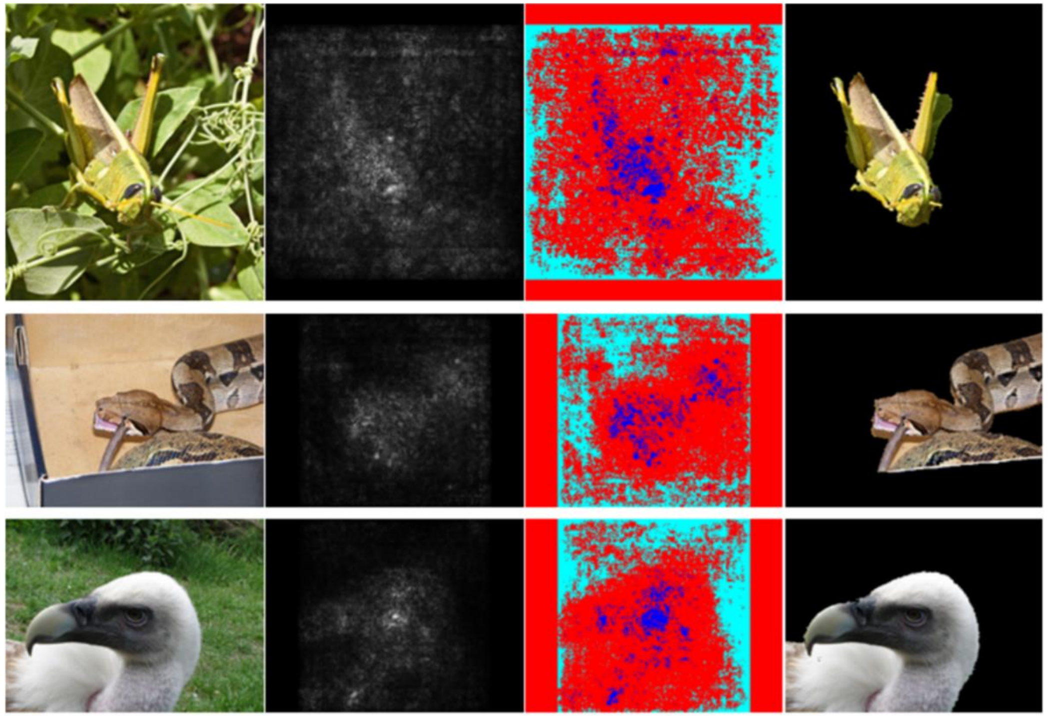
Simonyan, Vedaldi, and Zisserman, “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”, ICLR Workshop 2014.
Zeiler and Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014

Saliency Methods

Weakly-supervised Segmentation

- Saliency maps can be used to help unsupervised semantic segmentation

Gradients w.r.t. pixels



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

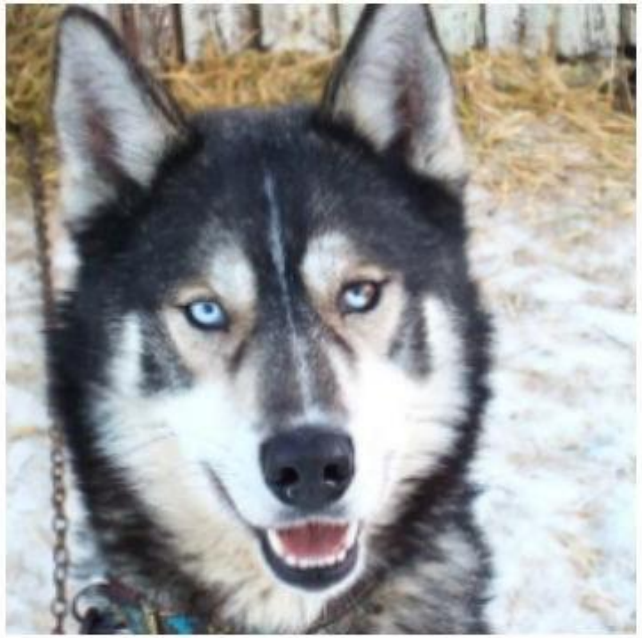
Saliency Methods

Uncovering Biases

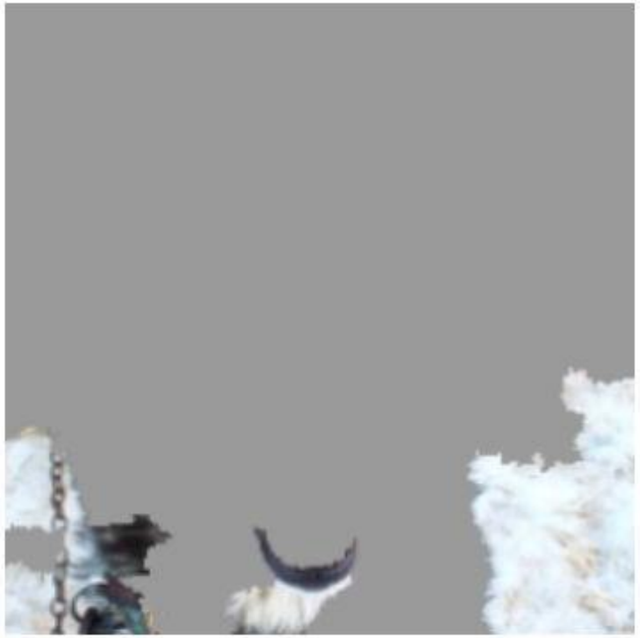
- Saliency Maps also find biases

When snow is presented in most of wolf images, network may use these snow pixels as salient regions for prediction

Wolf vs. dog classifier is actually a snow vs. no-snow classifier



(a) Husky classified as wolf



(b) Explanation

snow pixels as salient regions

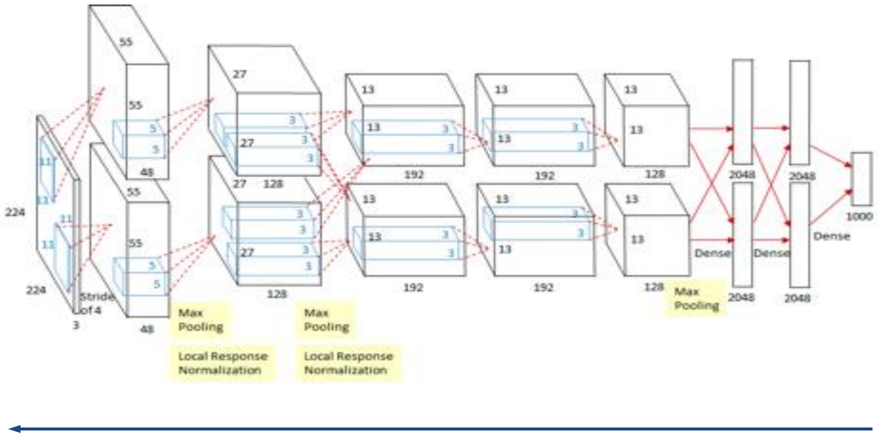
Ribeiro et al, ""Why Should I Trust You?" Explaining the Predictions of Any Classifier", ACM KDD 2016

Gradient-based Explanations

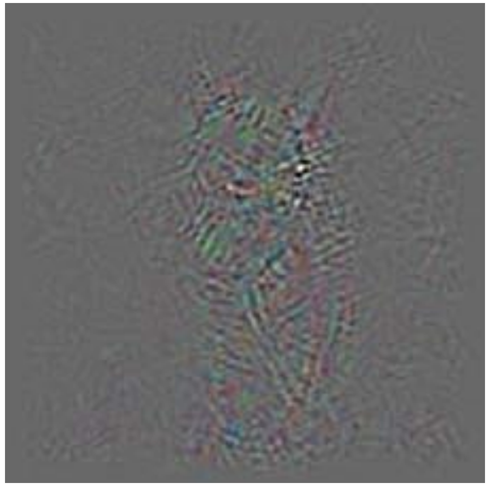
Vanilla Backpropagation

- Saliency map by vanilla backprop can be noisy

- We want cleaner saliency maps:
    - *Modifying (rectifying) the backpropagation* empirically produce better visualizations



Backward pass: Compute gradients



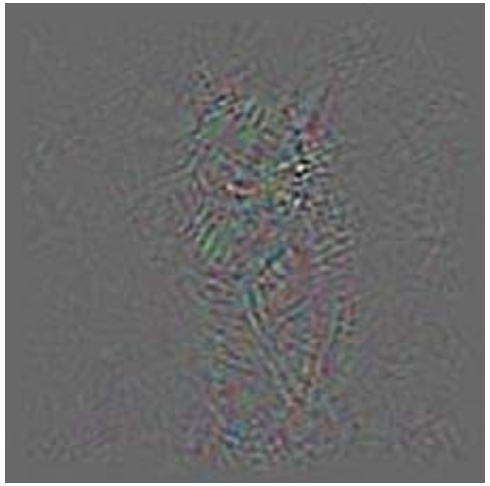
Saliency map by vanilla backprop

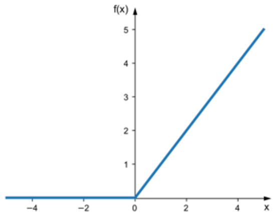
Gradient-based Explanations

Vanilla Backpropagation

Going backward performing all the operations of the network (Unpooling, Filtering…), and for ReLU non-linearities, **only pass gradients to regions of positive activations**

Saliency map by vanilla backprop





$$h^{l+1} = \max\{0, h^l\}$$

Forward pass

h^l

1	-1	5
2	-5	-7
-3	2	4

→

1	0	5
2	0	0
0	2	4

h^{l+1}

$$\frac{\partial L}{\partial h^l} = [\![h^l > 0]\!] \frac{\partial L}{\partial h^{l+1}}$$

Backward pass: backpropagation

-2	0	-1
6	0	0
0	-1	3

←

-2	3	-1
6	-3	1
2	-1	3

$\frac{\partial L}{\partial h^{l+1}}$

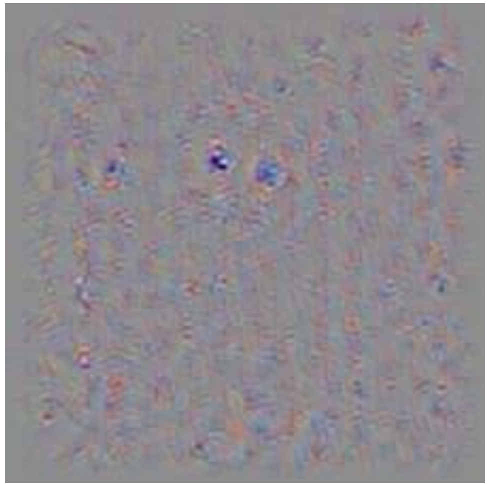
Gradients from the later layer

positive activations in the previous layer

Gradient-based Explanations

Deconvnet Backpropagation

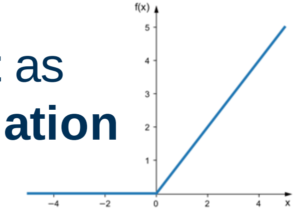
Saliency map by Deconv backprop



Cleaner saliency map

- The way DeconvNet Backpropagation handle the ReLU non-linearities is different as they propose to **only propagate positive gradient**
- *Rectifying the backpropagation* empirically produce better saliency visualizations

We can think of **Deconvnet** as **rectified gradients propagation**



$$\frac{\partial L}{\partial h^l} = [[\frac{\partial L}{\partial h^{l+1}} > 0]] \frac{\partial L}{\partial h^{l+1}}$$

Backward pass: "deconvnet"

0	3	0
6	0	1
2	0	3

←

-2	3	-1
6	-3	1
2	-1	3

$\frac{\partial L}{\partial h^{l+1}}$ Gradients from the later layer

positive gradient in the **later layer**

Simonyan et al., Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, *2014*

Gradient-based Explanations

Guided Backpropagation

Saliency map by
Guided Backpropagation

Cleaner saliency map

- Guided backpropagation propose to **propagate positive gradient and rectified by positive activations**
- **Non-intuitive** approach but **empirically** produce better saliency visualizations

$$h^{l+1} = \max\{0, h^l\}$$

Forward pass

$$h^l \quad \begin{bmatrix} 1 & -1 & 5 \\ 2 & -5 & -7 \\ -3 & 2 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 5 \\ 2 & 0 & 0 \\ 0 & 2 & 4 \end{bmatrix} \quad h^{l+1}$$

$$\frac{\partial L}{\partial h^l} = [[h^l > 0]][[\frac{\partial L}{\partial h^{l+1}} > 0]]\frac{\partial L}{\partial h^{l+1}}$$

Backward pass: *guided backpropagation*

$$\begin{bmatrix} 0 & 0 & 0 \\ 6 & 0 & 0 \\ 0 & 0 & 3 \end{bmatrix} \leftarrow \begin{bmatrix} -2 & 3 & -1 \\ 6 & -3 & 1 \\ 2 & -1 & 3 \end{bmatrix} \quad \frac{\partial L}{\partial h^{l+1}}$$

Gradients from the later layer

positive activations in the previous layer

positive gradient in the later layer

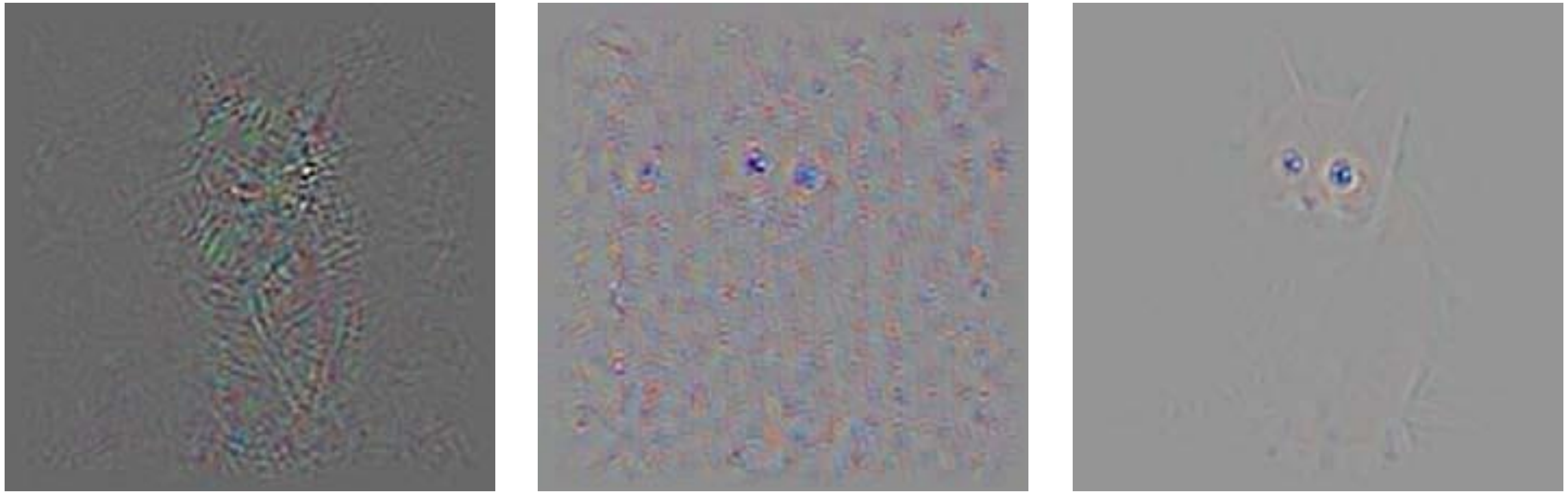
Springenberg, Dosovitskiy, et al., Striving for Simplicity: The all convolutional net, 2015

Gradient-based Explanations

Vanilla vs Deconvnet vs Guided Backpropagation

- Guided Backprop tends to be “cleanest”

Backprop Deconv Guided Backprop



Gradient-based Explanations

Problems with Guided Backpropagation

- Not very “class-discriminative”

GB for “airliner”

GB for “bus”

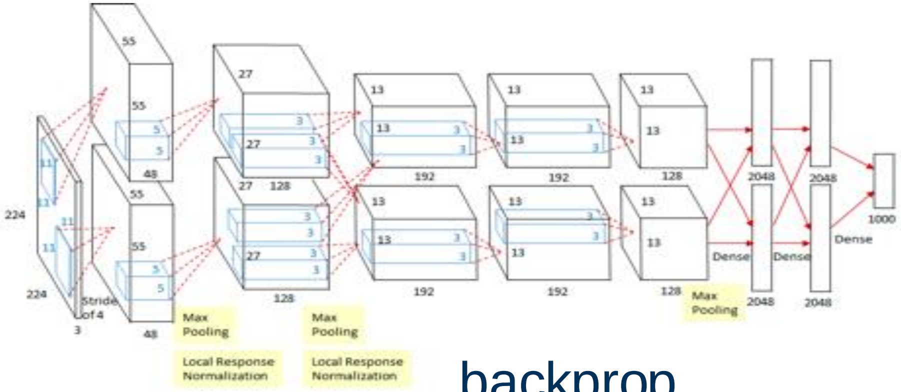
- Less related to the decision-making of neural networks

Nie, et al. "A theoretical explanation for perplexing behaviors of backpropagation-based visualizations." *International Conference on Machine Learning*. PMLR, 2018.

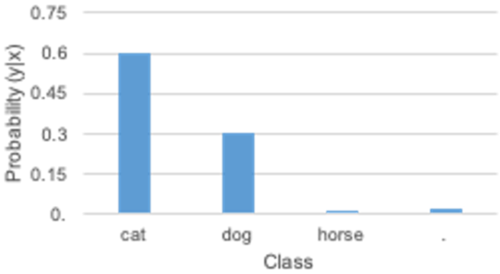
Gradient-based Explanations

Problems with Guided Backpropagation

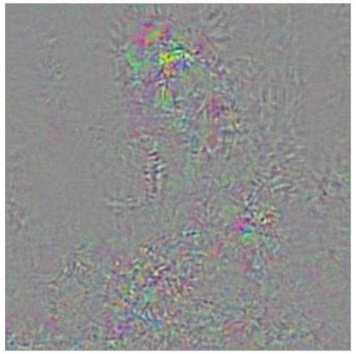
x



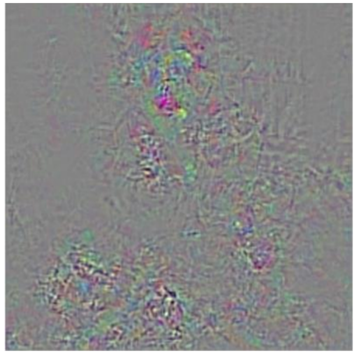
y



backprop



Backprop for \`cat’



Backprop for \`dog’

Noisy

$$w_c = \frac{\partial y_c}{\partial x}\bigg|_{x=x_0}$$

Guided Backprop for \`cat’

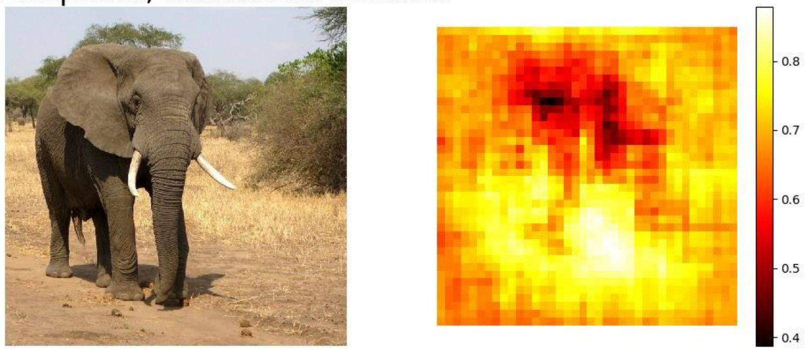
Guided Backprop for \`dog’

Not Class-Discriminative

Gradient-based Explanations
Summary

Saliency maps:
- Intervention-based:
  - perturbing pixels and see how the decision change
  - expensive
- Gradient-based:
  - approximates feature importance by backpropagation
  - computational efficient
  - not reflect decision making

African elephant, Loxodonta africana



Expensive

Guided Backprop for \`cat’

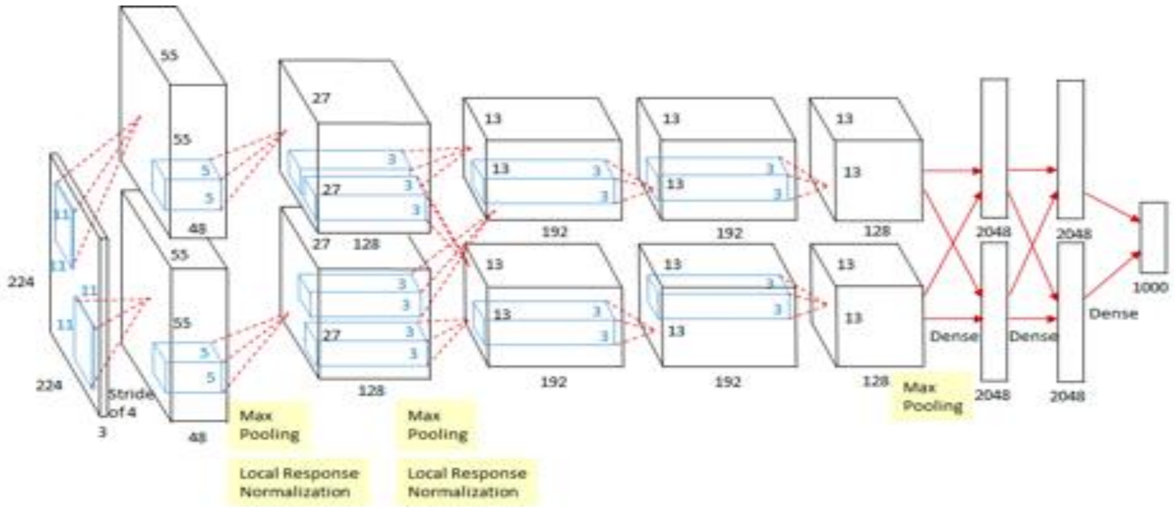
Guided Backprop for \`dog’

Not Class-Discriminative

Gradient-based Class Activation Map (Grad-CAM)

Motivation

- To find the important activations that are responsible for a particular class
- We want the activations:
    - **Class-discriminative** to reflect decision-making
    - **Preserve spatial information** to ensure spatial coverage of important regions



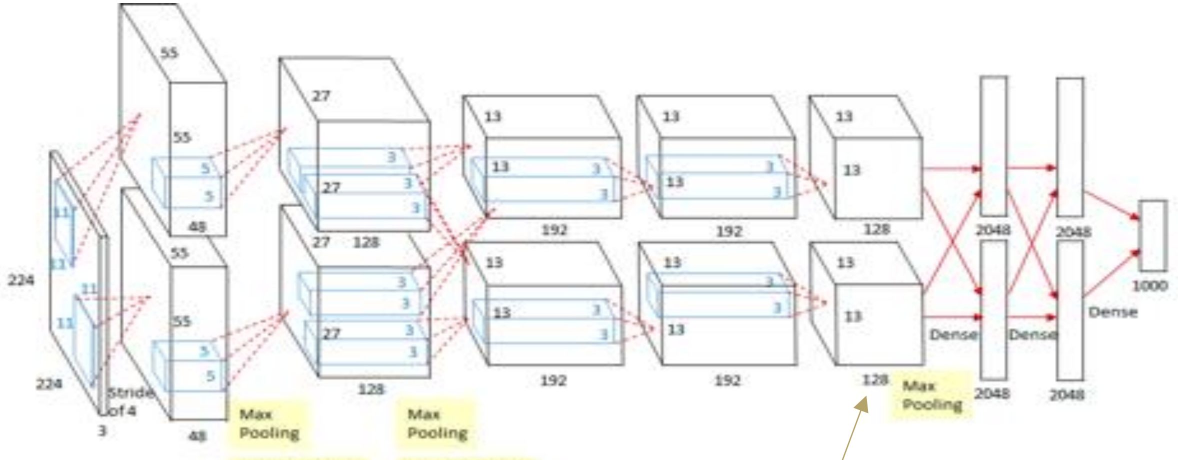
Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Gradient-based Class Activation Map (Grad-CAM)

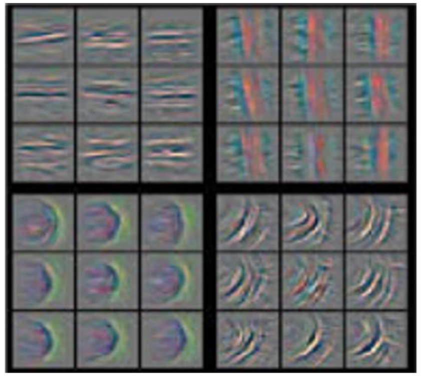
Motivation

- To find the important activations that are responsible for a particular class
- We want the activations:
  - Class-discriminative
  - Preserve spatial information

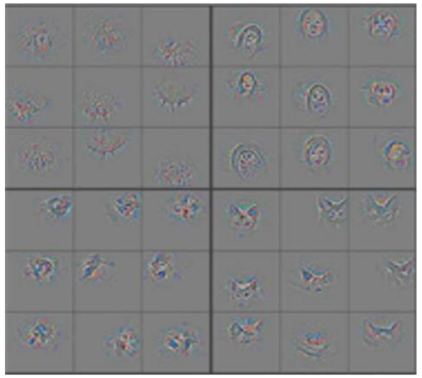
- Which layer to perturb:
  - **Higher layer** capture **class specific information**
  - **Spatial information** is **lost** in **fully-connected** layers
  - **Last convolutional layer** forms a best compromise between high-level semantics and detailed spatial resolution



Last convolutional layer



Low-level features captured by lower-layers



High-level features captured by higher-layers

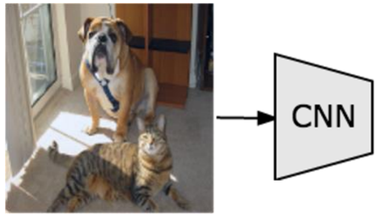
Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Gradient-based Class Activation Map (Grad-CAM)

Method

- Given an image, feed forward through CNN

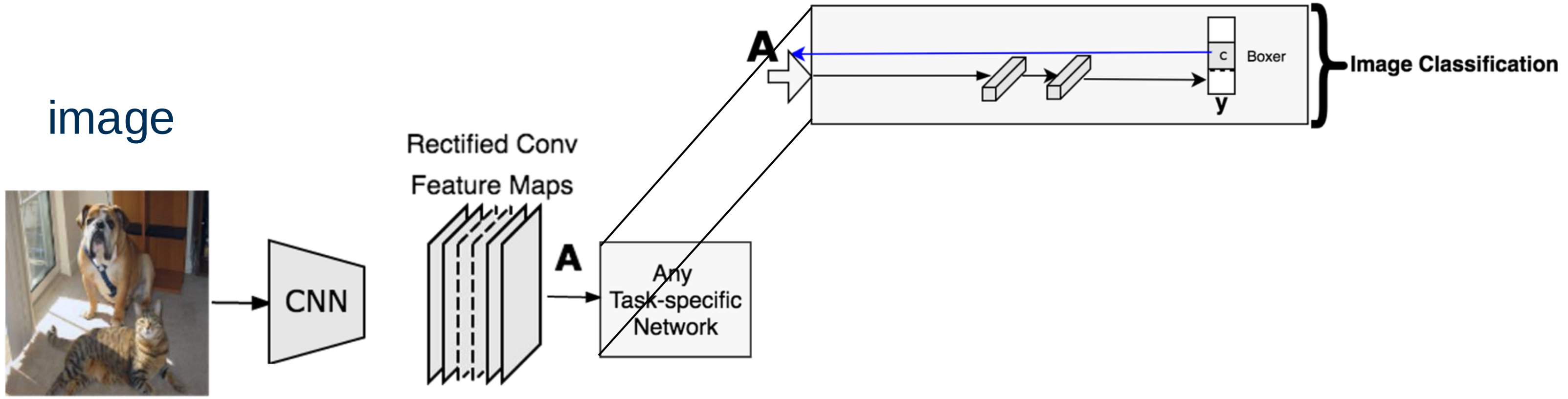
image



Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Gradient-based Class Activation Map (Grad-CAM)
Method

- Given an image, feed forward through CNN
- Final convolutional layer output feature maps for later task-specific layers, i.e., fc layer for classification

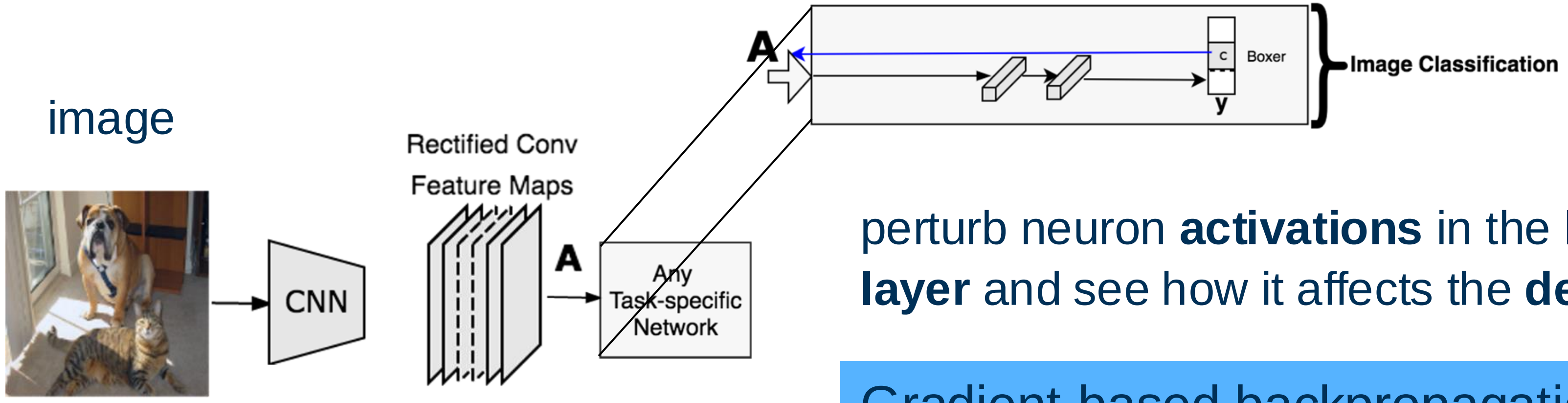


Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Gradient-based Class Activation Map (Grad-CAM)

Method

- Given an image, feed forward through CNN
- Final convolutional layer output feature maps for later task-specific layers, i.e., fc layer for classification



perturb neuron **activations** in the **last conv layer** and see how it affects the **decision**

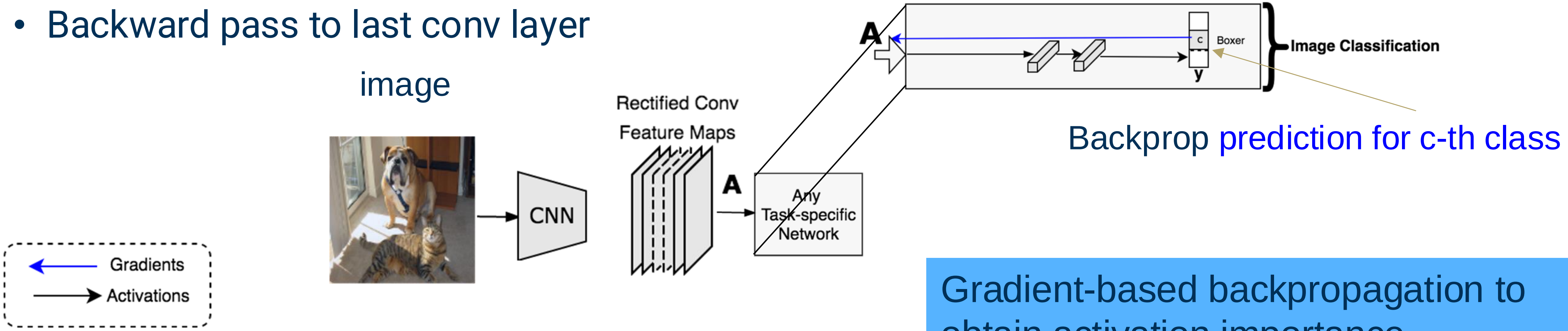
Gradient-based backpropagation to obtain activation importance

Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Gradient-based Class Activation Map (Grad-CAM)

Method

- Given an image, feed forward through CNN
- Final convolutional layer output feature maps for later task-specific layers, i.e., fc layer for classification
- Backward pass to last conv layer



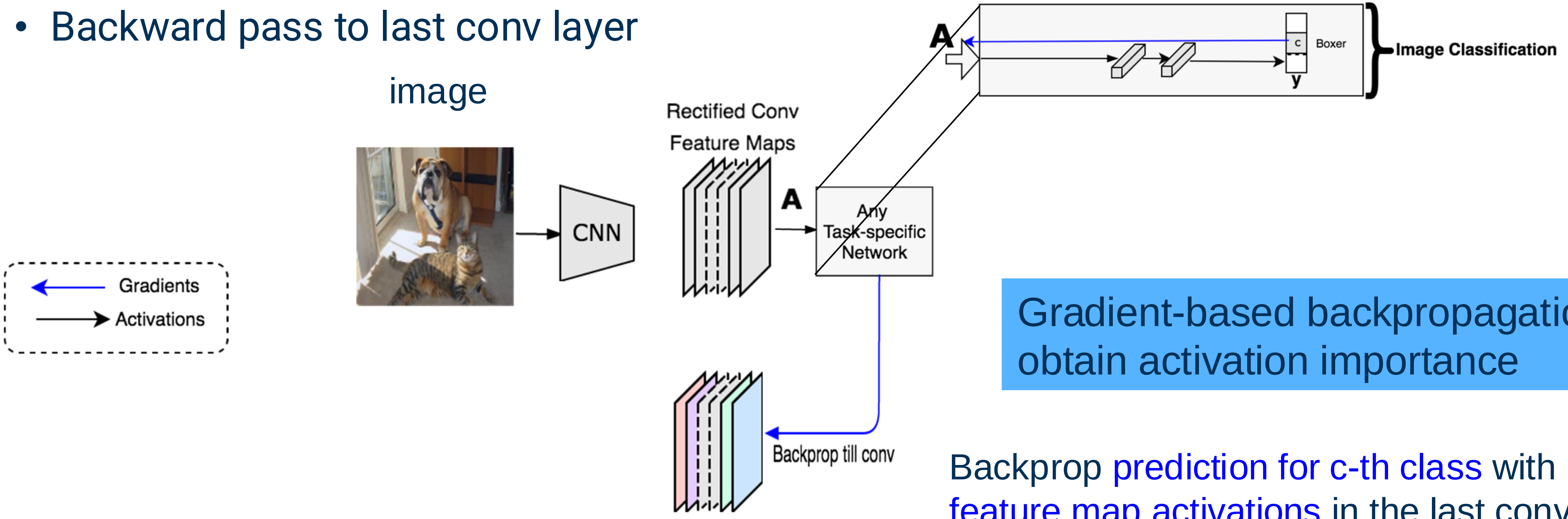
Gradient-based backpropagation to obtain activation importance

Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Gradient-based Class Activation Map (Grad-CAM)

Method

- Given an image, feed forward through CNN
- Final convolutional layer output feature maps for later task-specific layers, i.e., fc layer for classification
- Backward pass to last conv layer



Gradient-based backpropagation to obtain activation importance

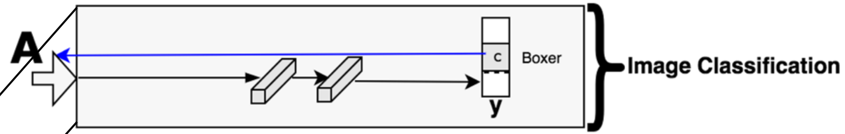
Backprop prediction for c-th class with respect to feature map activations in the last conv layer

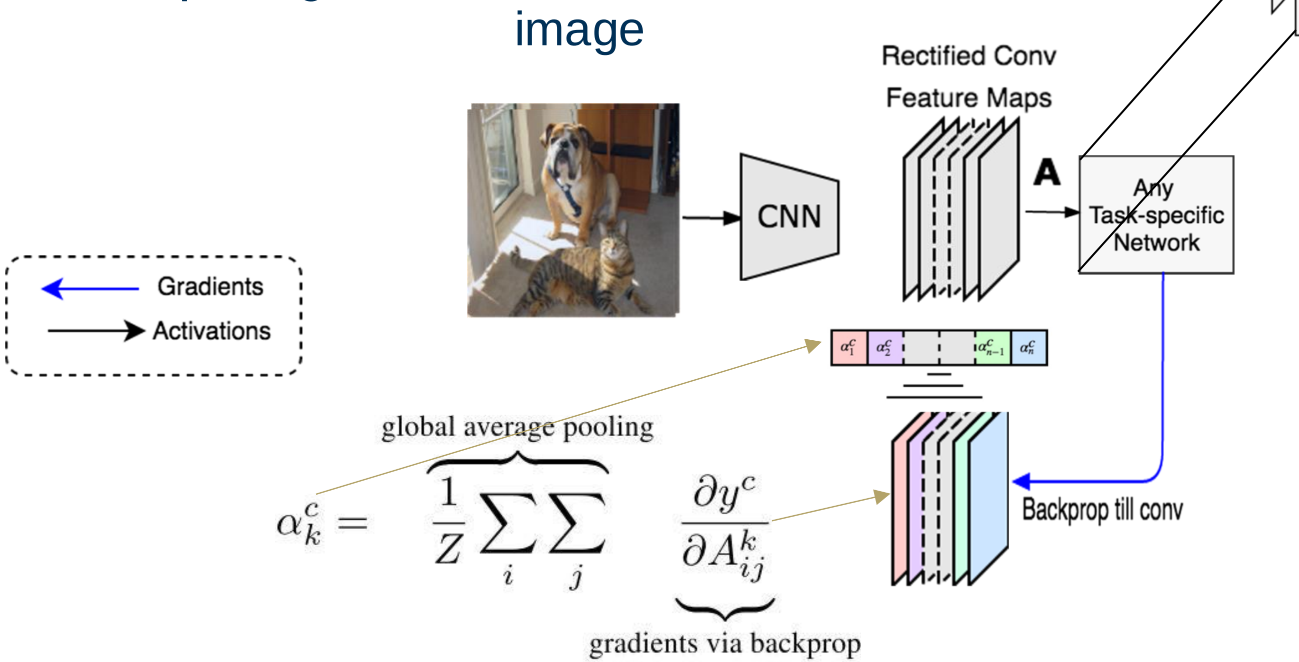
Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Gradient-based Class Activation Map (Grad-CAM)

Method

- Given an image, feed forward through CNN
- Final convolutional layer output feature maps for later task-specific layers
- Backward pass to last conv layer
- Compute gradients w.r.t. last conv activations





$\frac{\partial y^c}{\partial A^k}$: gradients of prediction for c-th class with respect to k-th feature map activations A^k in the last conv layer

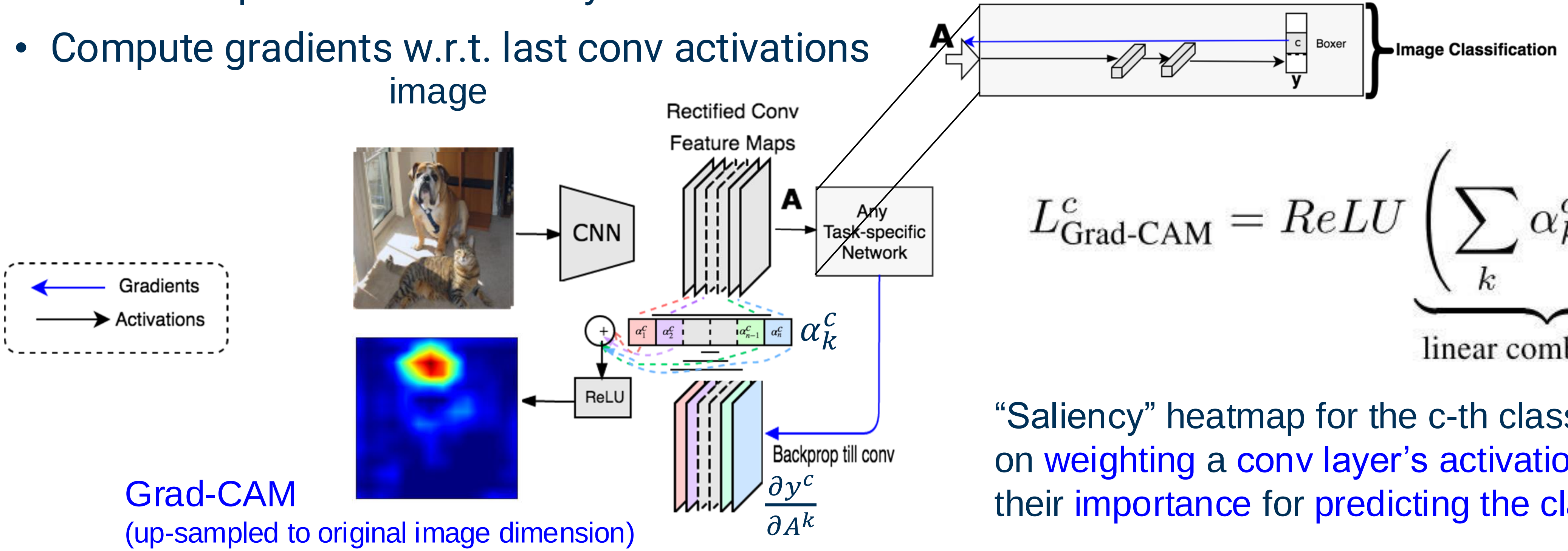
α_k^c is the scalar importance of k-th feature map obtained by averaging $\frac{\partial y^c}{\partial A^k}$ spatially

Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Gradient-based Class Activation Map (Grad-CAM)

Method

- Given an image, feed forward through CNN
- Final convolutional layer output feature maps for later task-specific layers
- Backward pass to last conv layer
- Compute gradients w.r.t. last conv activations



Grad-CAM
(up-sampled to original image dimension)

$$L^c_{\text{Grad-CAM}} = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$

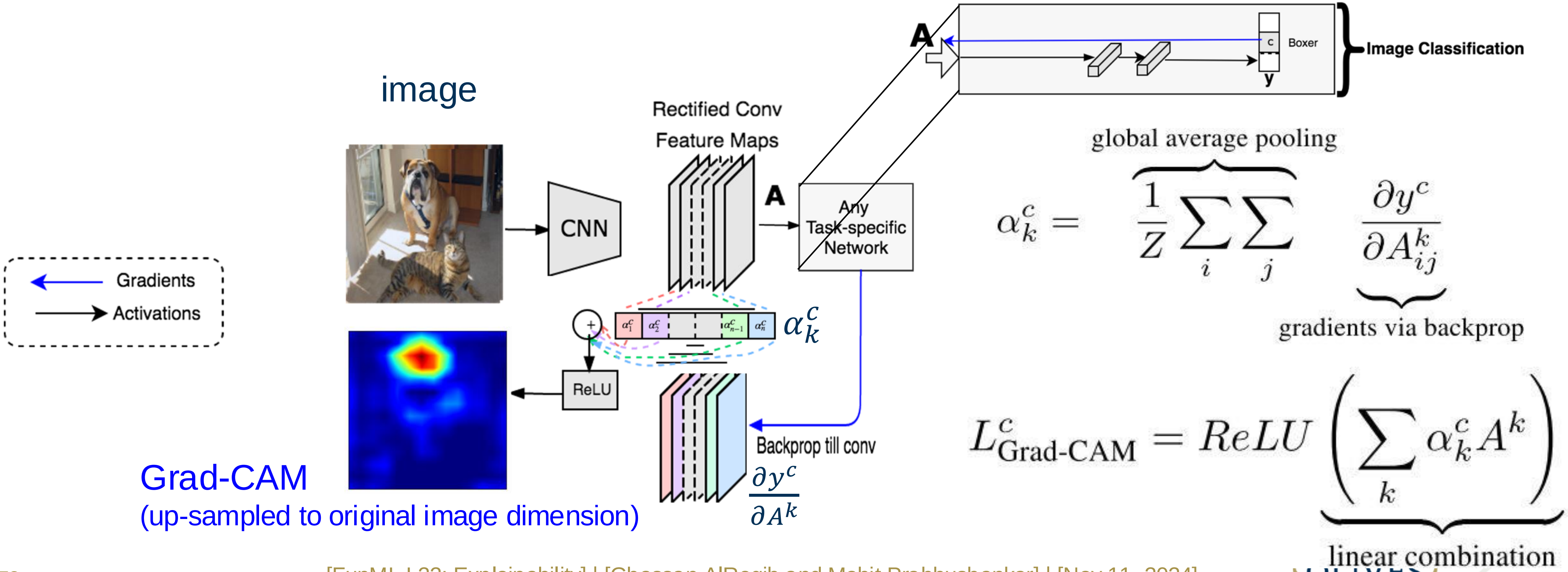
"Saliency" heatmap for the c-th class is based on weighting a conv layer's activation maps by their importance for predicting the class

Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Gradient-based Class Activation Map (Grad-CAM)

Method

Grad-CAM uses the **gradient** information flowing into the **last convolutional layer** of the CNN to assign **importance values** to each activation **for a particular decision of interest.**

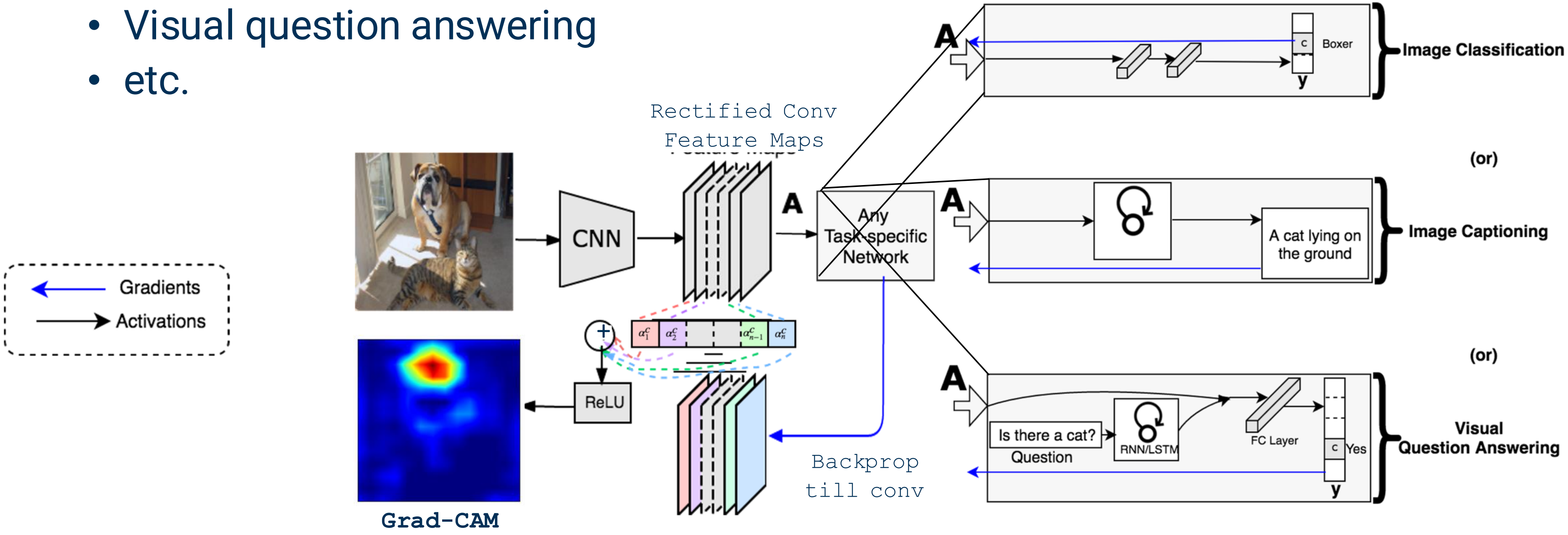


Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Gradient-based Class Activation Map (Grad-CAM)

Method

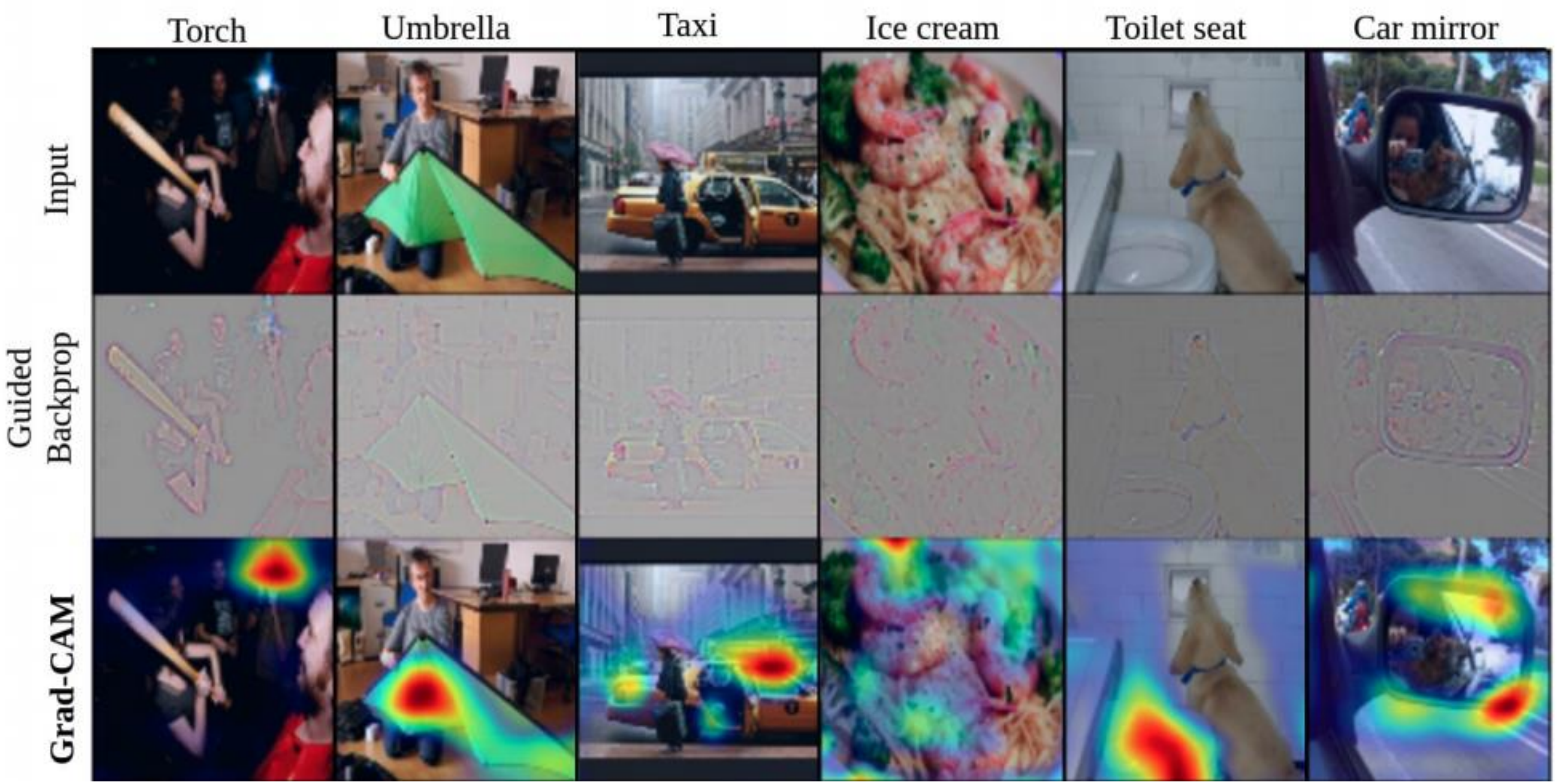
- Grad-CAM generalizes to different tasks:
  - Image classification
  - Image captioning
  - Visual question answering
  - etc.



Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Gradient-based Class Activation Map (Grad-CAM)

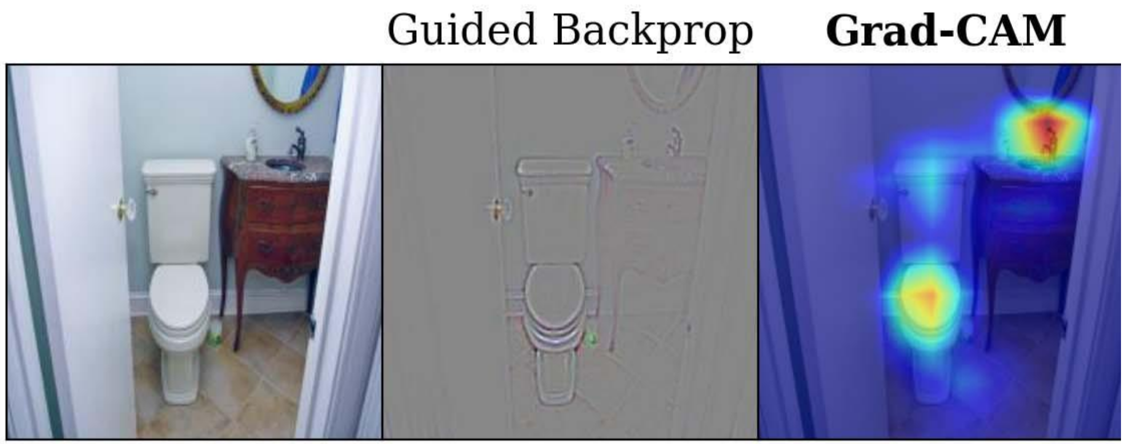
Method



Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Gradient-based Class Activation Map (Grad-CAM)

Method



A bathroom with a toilet and a sink

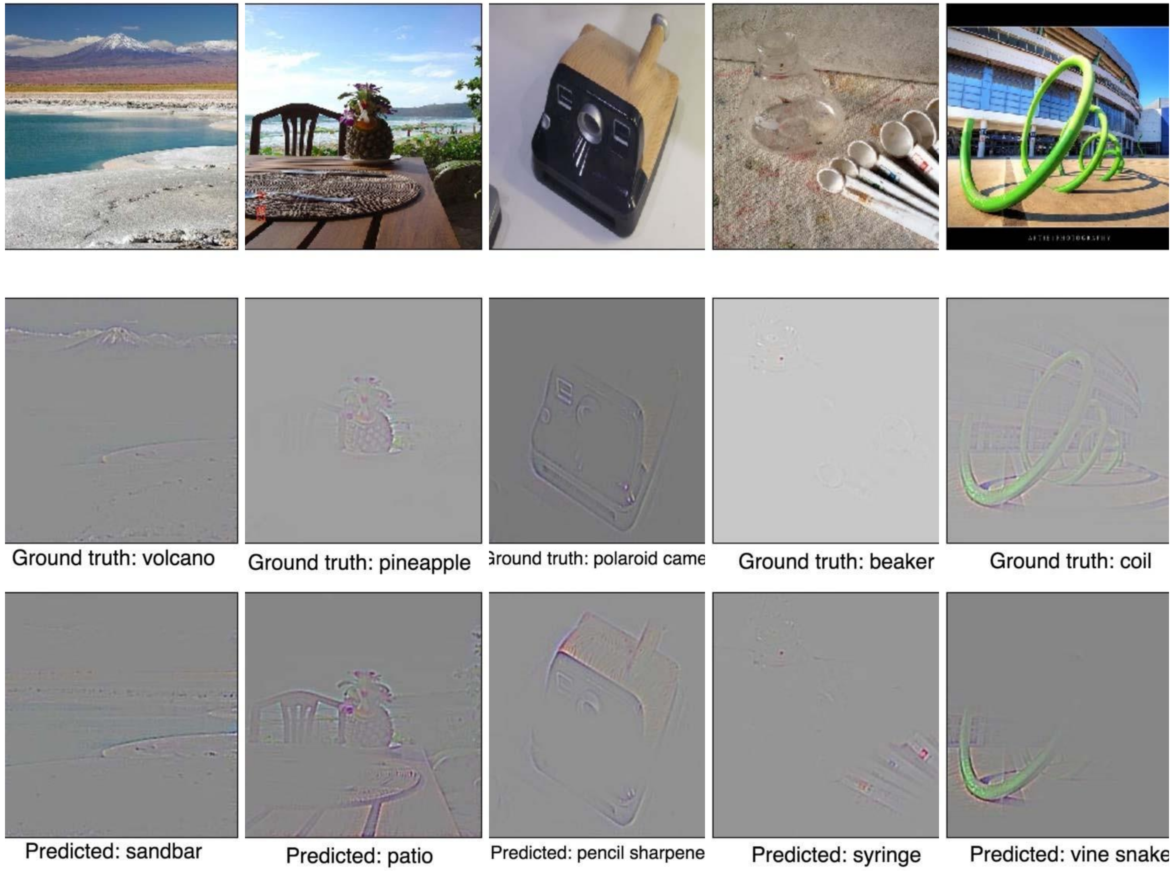
A horse is standing in a field with a fence in the background

Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Gradient-based Class Activation Map (Grad-CAM)

Method

- Even unreasonable predictions seem justifiable



Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.

Appendix
Notations

- x_i: a single feature
- $\boldsymbol{x}_i$: feature vector (a data sample)
- $\boldsymbol{x}_{:,i}$: feature vector of all data samples
- $\boldsymbol{X}$: matrix of feature vectors (dataset)
- N: number of data samples
- $\boldsymbol{W}$: weight matrix
- $\boldsymbol{b}$: bias vector
- $\boldsymbol{v}(t)$: first moment at time t
- $\boldsymbol{G}(t)$: second moment at time t
- $\boldsymbol{H}(\boldsymbol{\theta})$: Hessian matrix
- P: number of features in a feature vector
- α: learning rate
- Bold letter/symbol: vector
- Bold capital letters/symbol: matrix