Lecture 11: Interpretation Methods for Image Classification: Saliency Maps and Class Activation Maximization (CAM)

AC295

Advanced Practical Data Science

Pavlos Protopapas

Outline

1: Communications

2: Needs for Transparent Models

3: Interpretation Methods for Image Classification

4: Saliency Maps

5: Class Activation Maximization (CAM)

6: Feature Visualization

Communications

- **Milestone 3 (due 12/03):** Submit code with EDA + baseline model. A short writeup that explains what you have done so far and what you plan to achieve in next few weeks
- **Milestone 4 (due 12/11 & 12/15)** -
  1. Submit video presentation (6 mins per group) by 12/11
  2. Submit code and medium post and peer review by 12/15

We will have class showcase Tue 12/15 10:30 am where we will discuss a **few** projects.

Outline

1: Communications

2: Needs for Transparent Models

3: Interpretation Methods for Image Classification

4: Saliency Maps

5: Class Activation Maximization (CAM)

6: Feature Visualization

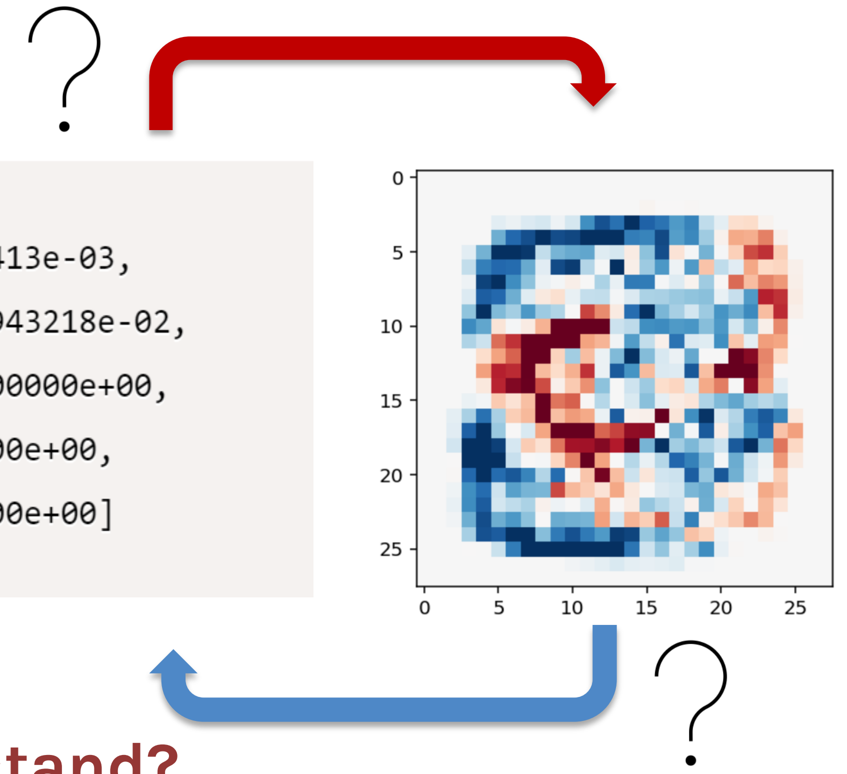
Why Care About Interpretability?

1. Help building **trust**:
    - Humans are reluctant to use ML for critical tasks
    - Fear of unknown when people confront new technologies
2. Promote **safety**:
    - Explain model's representation (i.e. important feature) providing opportunities to remedy the situation
3. Allow for **contestability**:
    - Black-box models don't decompose the decision into sub-models or illustrate a chain of reasoning

Defining Interpretability: Simulatibility

What is it? This property addresses whether a human could go through each step of the algorithm and check if each step is reasonable to them.

```
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, -1.47542413e-03,
-1.67811041e-04, -3.83280468e-02, -8.10846867e-02, -5.01943218e-02,
-2.90314621e-02, -2.65494116e-02, -8.29385683e-03, 0.00000000e+00,
0.00000000e+00, 1.67390785e-04, 3.92789141e-04, 0.00000000e+00,
0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00]
```

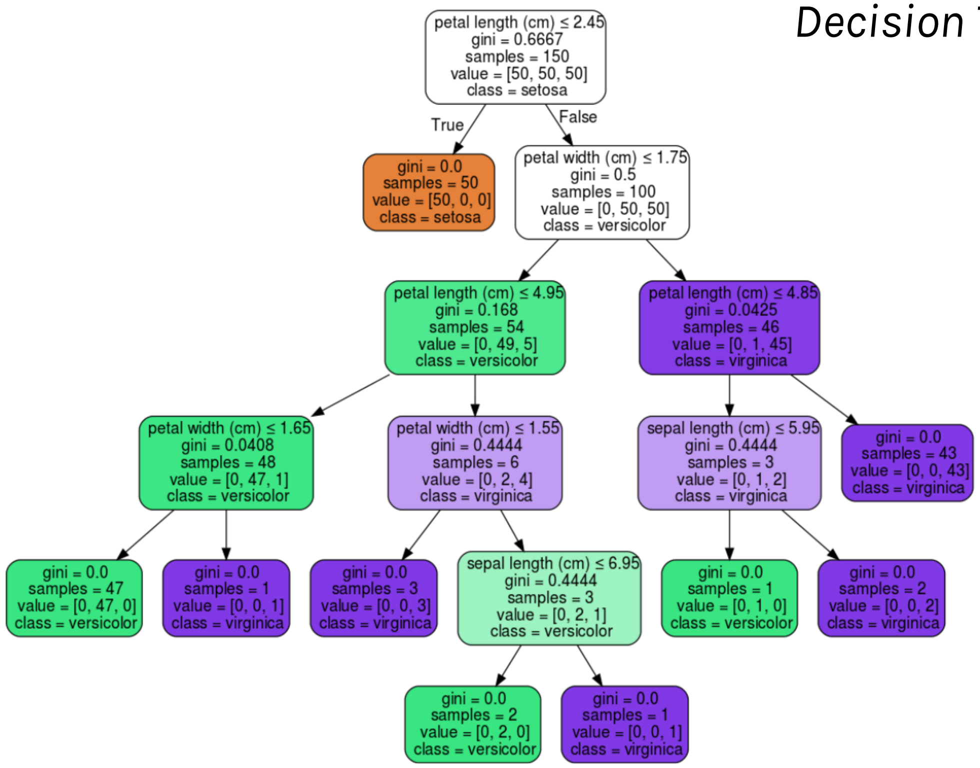


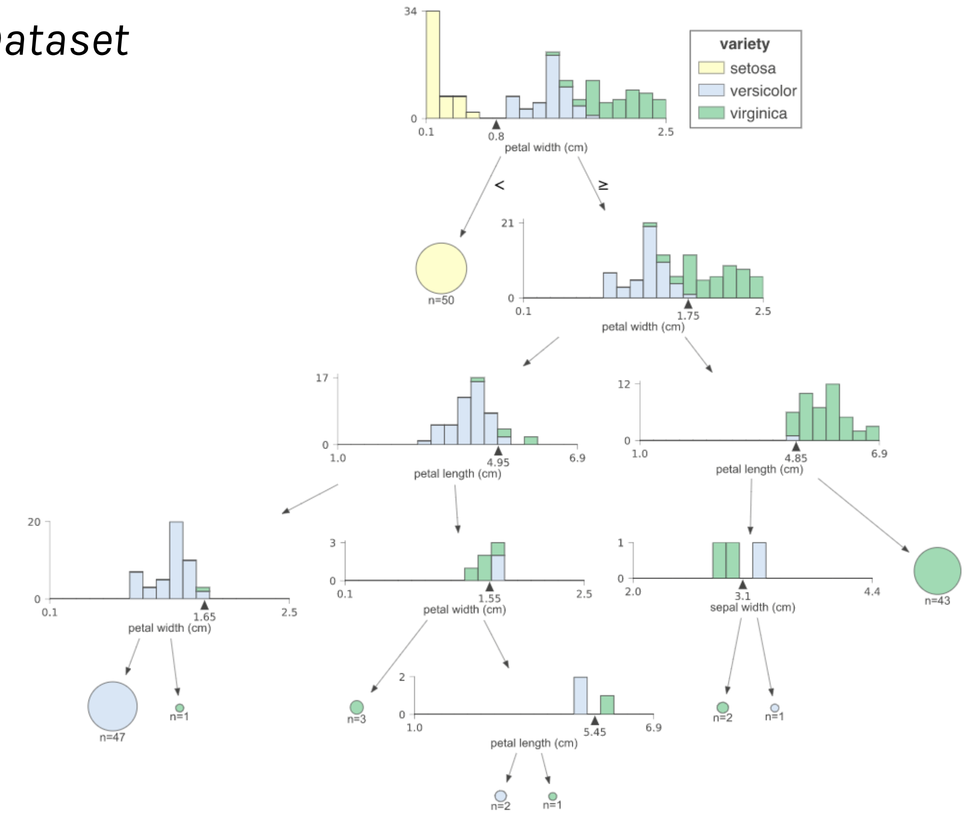
Bottom line, **is the model simple to understand?**

Defining Interpretability: Decomposability

What is it? This property addresses what the model is doing at each step. Decision are the most interpretable. How can we make it clear at each step?

Decision Trees on Iris Dataset



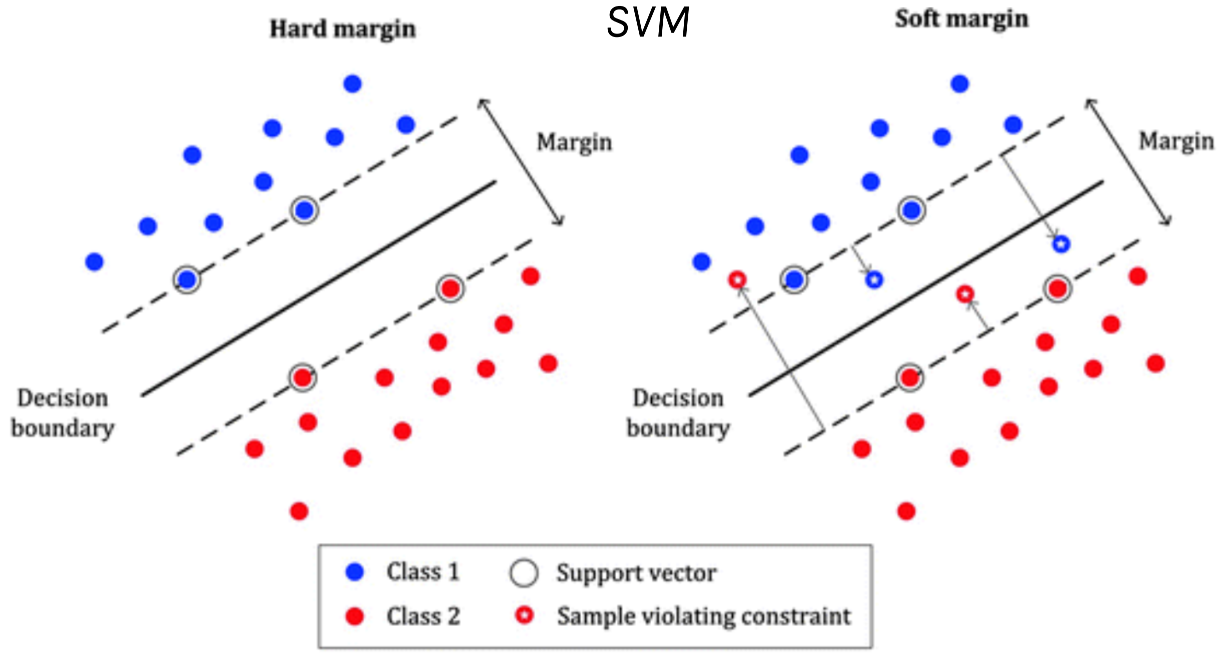


explained.ai, [How to Visualize Decision Trees](), sklearn versus dtreeviz

Defining Interpretability: Algorithmic Transparency

What is it? This property addresses whether the model has any desirable properties which are easy to understand. Good reparameterization, what else?

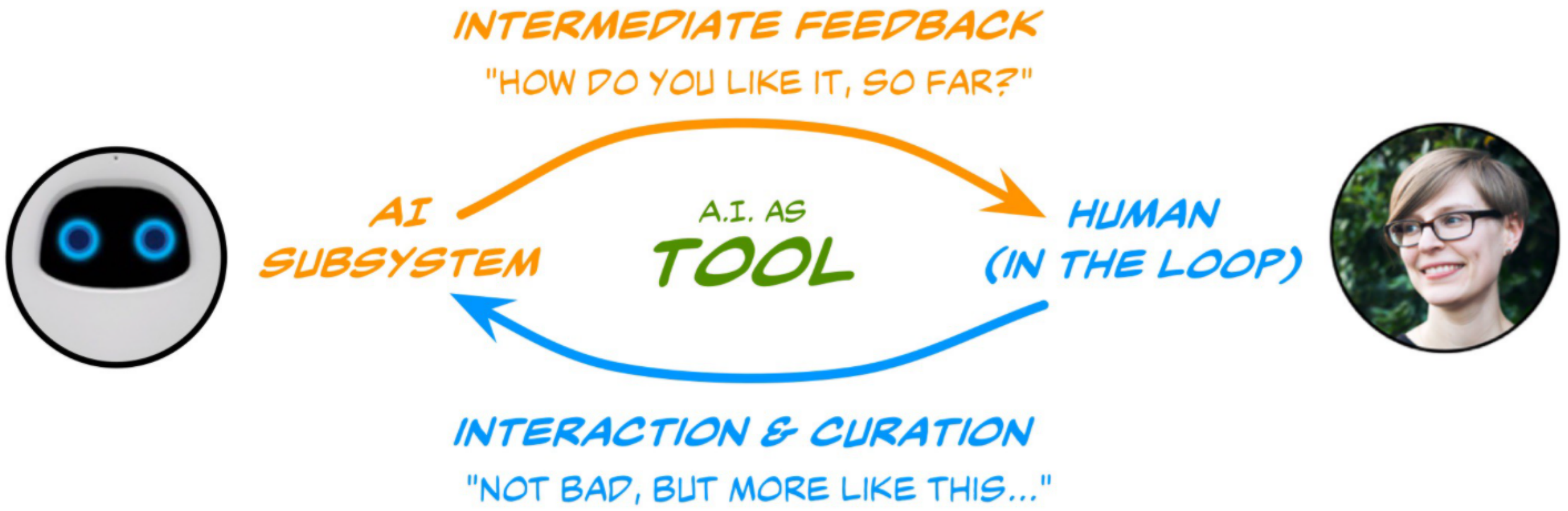
SVM



Bottom line, **only some models are informative from a simulatable perspective.**

Post-Hoc Interpretability

What is it? Things we can learn from the model after training has finished.



How can models incorporate human interpretation into their decision-making?

Stanford, [Humans in the Loop General Bluprint]

Why do we need Transparent Model?

How? We can apply different available **interpretation methods**. In this class we will cover those that **applies to image classification** networks:

1. Saliency Maps
2. Class Activation Maximization (CAM)

Outline

1: Communications

2: Needs for Transparent Models

3: Interpretation Methods for Image Classification

4: Saliency Maps

5: Class Activation Maximization (CAM)

6: Practical Real-World Implementation: Feature Visualization

Interpretation Methods for Image Classification

There are three common ways of categorizing Interpretation methods found in the literature and they are based on:

1. **Locality** (i.e. local, and global)
2. **Specificity** (i.e. model-specific, and model-agnostic)
3. **Signal flow** (i.e. Gradient-based backpropagation, and Perturbation-based forward propagation)

Let’s see them in more detail.

Salehi M., A Review of Different Interpretation Methods in Deep Learning, Medium, 2019

Interpretation Methods for Image Classification

Methods based on **locality differs on they way they look (or not) at a set of inputs**. Local methods explain the predictions by investigating the performance on a specific set of inputs, while the global one looks only at how the model works without probing its functionality.

1. **Local:** probe if the model uses the right cues in the input to come up with the correct prediction (e.g it finds the cat in the image and not the background);

2. **Global:** takes the most important feature for predicting the output of the model without feeding any datapoint (e.g. select latitude over longitude in housing price prediction task because of it’s best estimate).

Interpretation Methods for Image Classification

Methods based on **specificity look at the model used to train a set of inputs.** Model-specific look at the model functionality, while model-agnostic care about whether the function gives an output for each valid input it gets.

1. **Model-specific:** can only be applied to a specific group of models (e.g. a linear model cannot explain non-linear relationship in the input space);

2. **Model-agnostic:** can be applied to any machine learning model as black-box function for which not internal structure is provided as long as it provide let's say good prediction.

Interpretation Methods for Image Classification

Methods based on the **signal flow look at the information within the model.**

1. **Gradient-based backpropagation:** backpropagate a signal from the output towards the input to show how changing intermediate value would affect the output;

2. **Perturbation-based forward propagation methods:** perturb the input and probe its possible effects on the prediction of the network (e.g. occluding some part of the input image and investigating the changes in the output). It uses a forward pass to transfer the changes in input to the scores that are computed at the very last layers.

Outline

1: Communications

2: Needs for Transparent Models

3: Interpretation Methods for Image Classification

4: Saliency Maps

5: Class Activation Maximization (CAM)

6: Feature Visualization

Saliency Maps

Saliency map is the oldest and most frequently used explanation method for interpreting the predictions of convolutional neural networks (CNNs).

There are three main approaches to getting the saliency map:

1. Using **Deconvolutional Networks** proposed by Zeiler and Fergus 2013
2. Using a **Gradient Based Backpropagation method** by Symonian et al. 2013
3. Using a **Guided Backpropagation Algorithm** by Springenberg et al. 2014

Saliency Maps with DeconvNet

Zeiler and Fergus proposed a method able to recognize what features in the input image an intermediate layer of the network is looking for.

The method is known as **Deconvolutional Network** (DeconvNet) because it inverts the operations from a particular layer to the input using the activation of that specific layer.

To invert the process the authors used:

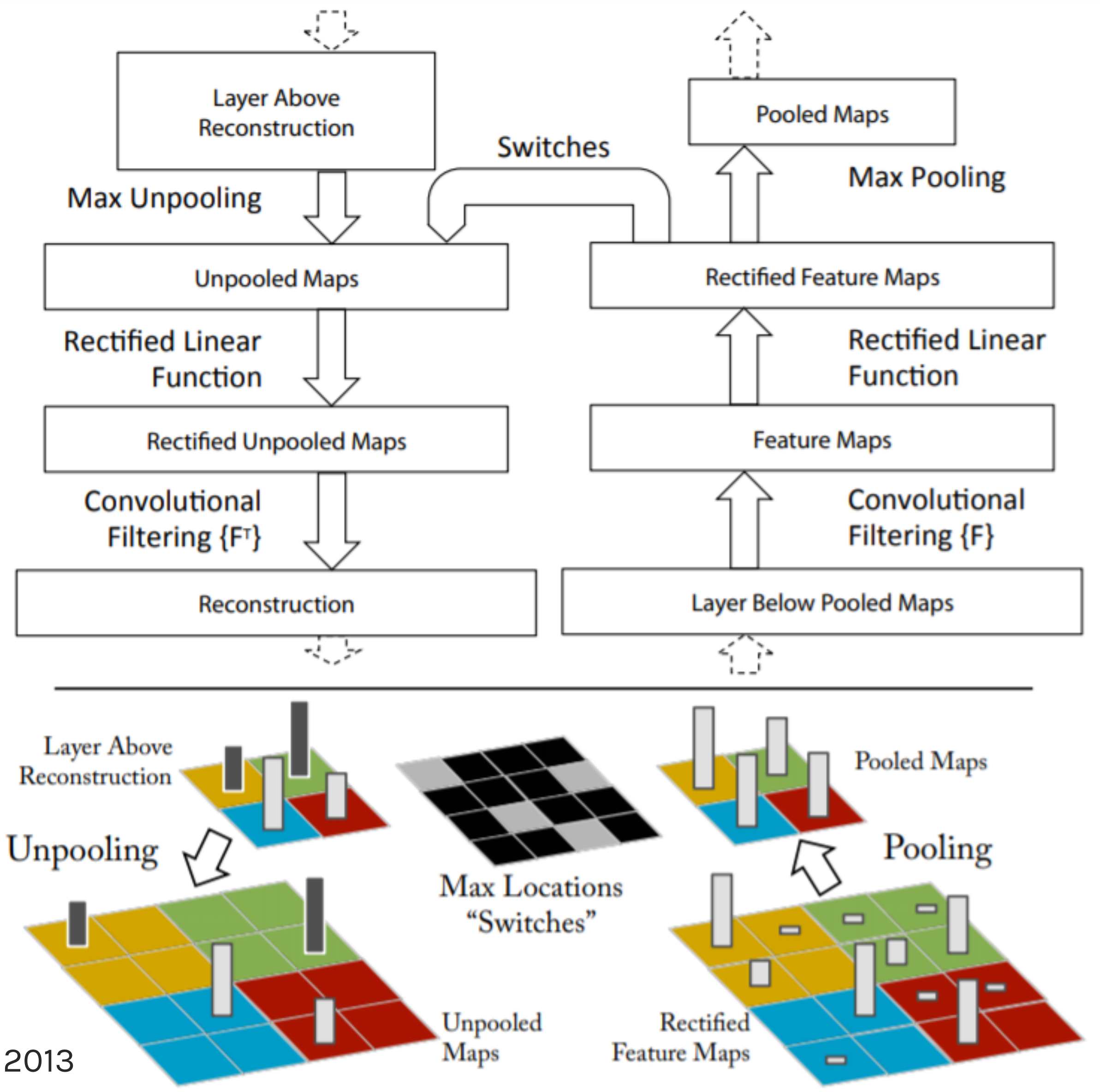
- **Unpooling** as the inverse of pooling
- **Inverse ReLU** removing the negative values as the inverse of itself

Zeiler and Fergus, Adaptive Deconvolutional Networks for Mid and High Level Feature Learning, 2011

Saliency Maps with DeconvNet <cont>

The whole process is illustrated in the figure:

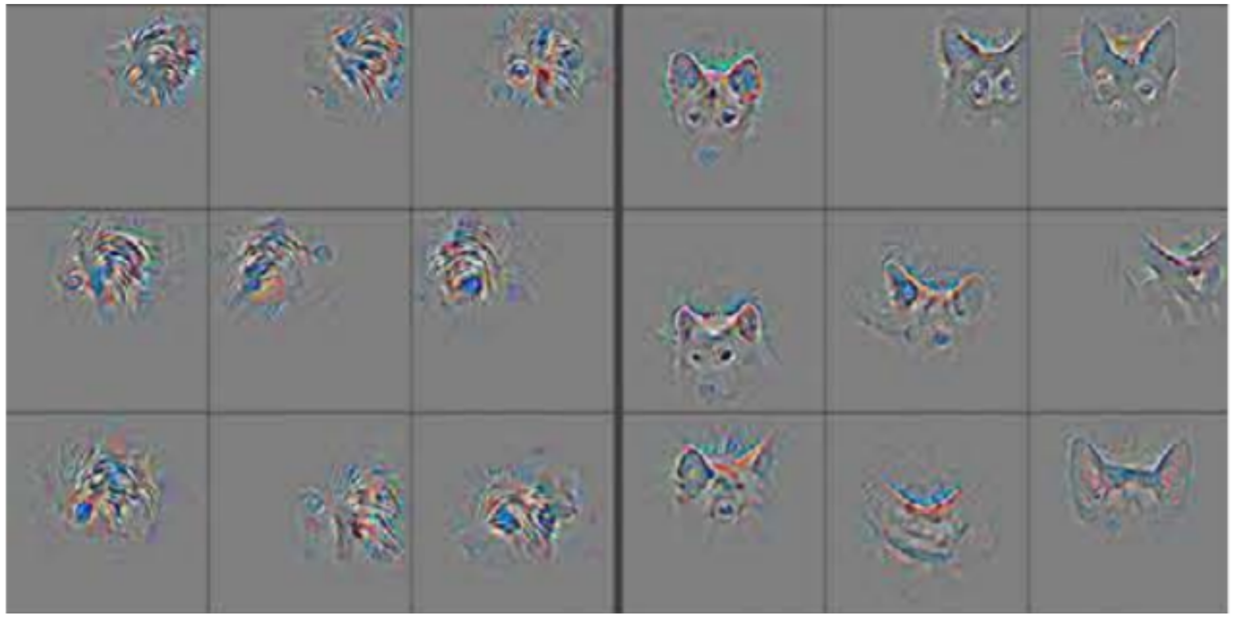
The authors used a module called **switch** to recover the positions of maxima in the forward pass because the pooling operation is non-invertible.



Zeiler and Fergus, Visualizing and Understanding Convolutional Networks, 2013

Saliency Maps with DeconvNet <cont>

Here are some of the results of the reconstruction of the input image from the activations of the fifth layer.

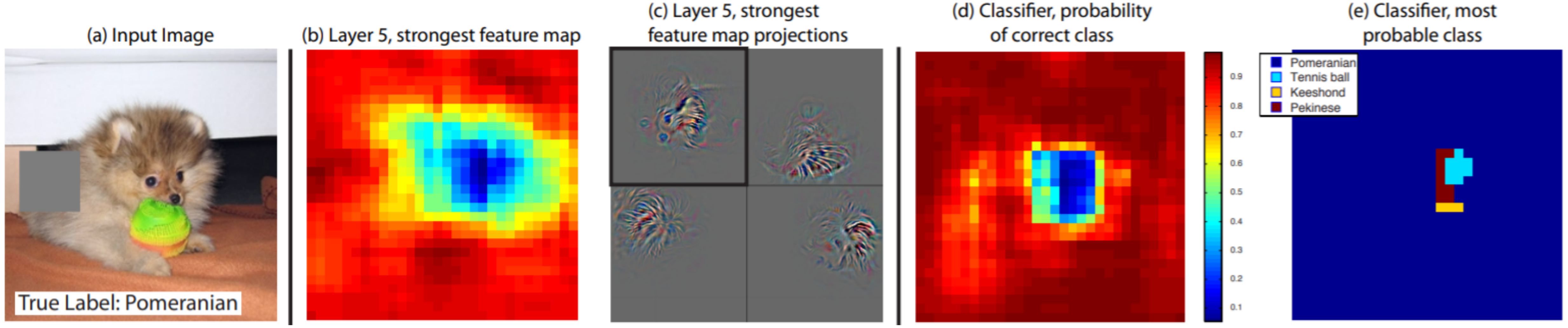


Zeiler and Fergus, [Visualizing and Understanding Convolutional Networks](), 2013

Saliency Maps with DeconvNet <cont>

Is if the model is truly identifying the location of the object in the image, or just using the surrounding context?

The examples show the model is localizing the objects within the scene, as the **probability of the correct class drops when the object is occluded.**



Zeiler and Fergus, Visualizing and Understanding Convolutional Networks, 2013

Saliency Maps with Gradient Based Backprop

Symonian et al. were the first to propose a method that uses the backpropagation algorithm to compute the gradients of logits wr.t. to the input of the network (for training the network, the gradients are computed w.r.t. the parameters of the network).

Using backpropagation, we can highlight pixels of the input image based on the amount of the gradient they receive, which shows their contribution to the final score.

Saliency Maps with Gradient Based Backprop

Class Model Visualization:

Given a learned classification ConvNet and a class of interest, the visualization method consists in numerically generating an image, which is representative of the class in terms of the ConvNet class scoring model.

Find an L2-regularised image, such that the score of the class c Sc is high:

$$\boxed{\arg\max_{I} S_c(I) - \lambda \|I\|^2{}_2}$$

Saliency Maps with Gradient Based Backprop

Image-Specific Class Saliency Visualization:

ConvNet can be queried about the spatial support of a particular class in a given image.

Given an image I_0, a class c, and a classification ConvNet with the class score function $S_c(I_0)$, we would like to rank the pixels of I_0 based on their influence on the score $S_c(I_0)$.

$$\boxed{S_c(I) \approx w^T I + b}$$

where

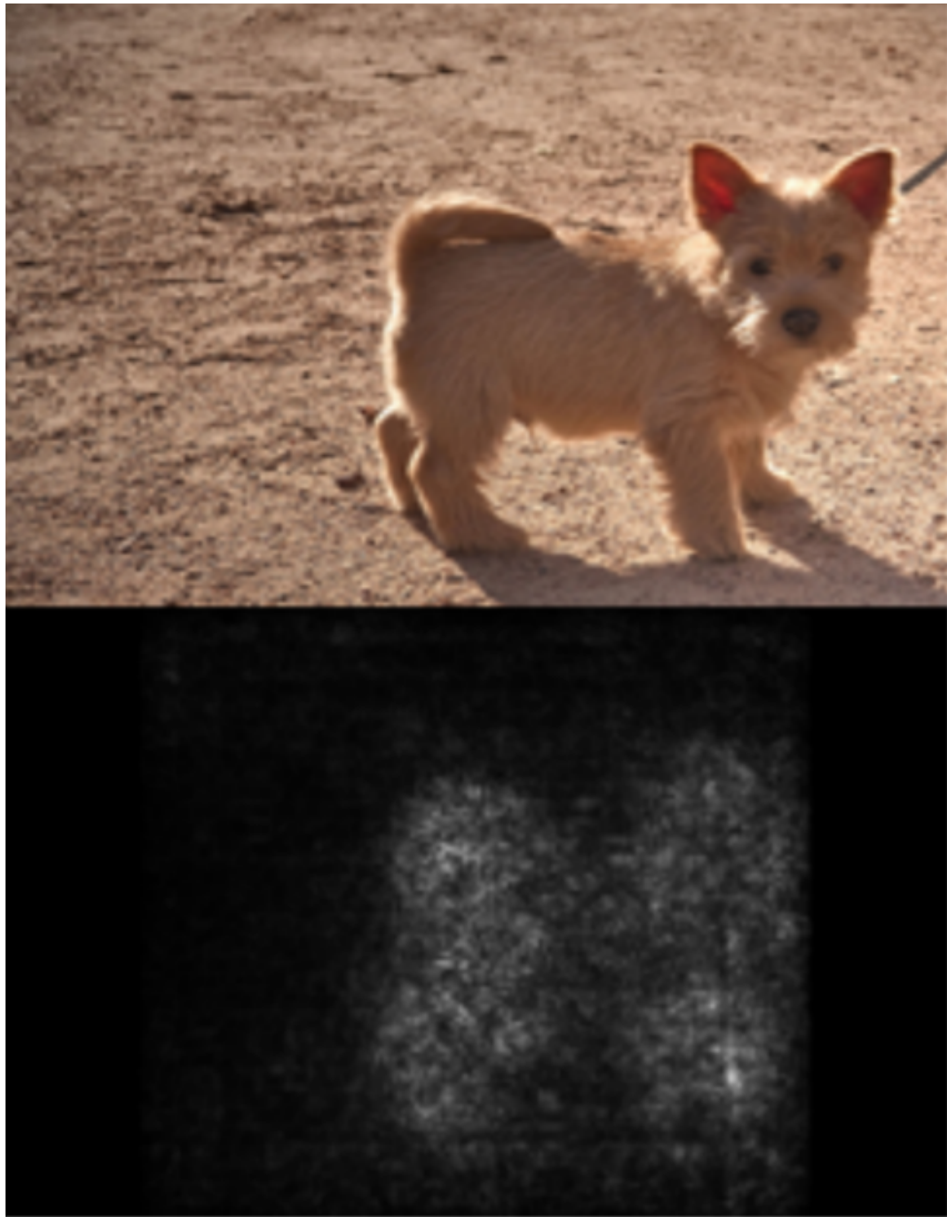
$$\boxed{w = \left.\frac{\partial S_c}{\partial I}\right|_{I_o}}$$

Saliency Maps with Gradient Based Backprop <cont>

The maps were extracted using a single back-propagation pass through a classification ConvNet.

No additional annotation (except for the image labels) was used in training.

Image-specific class saliency maps for the top-1 predicted class in ILSVRC-2013 test images.



Symonian et al, Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, 2013

Saliency Maps w Guided Backpropagation Algorithm

Springenberg et al. **combined DeconvNet and Gradient Based Backpropagation** and proposed the **Guided Backpropagation Algorithm** as a third way of getting saliency maps.

In fact, Instead of masking the importance signal based on the positions of negative values of the input in forward-pass (backpropagation) or the negative values from the reconstruction signal flowing from top to bottom (deconvolution), they **mask the signal if each one of these cases occurs**.

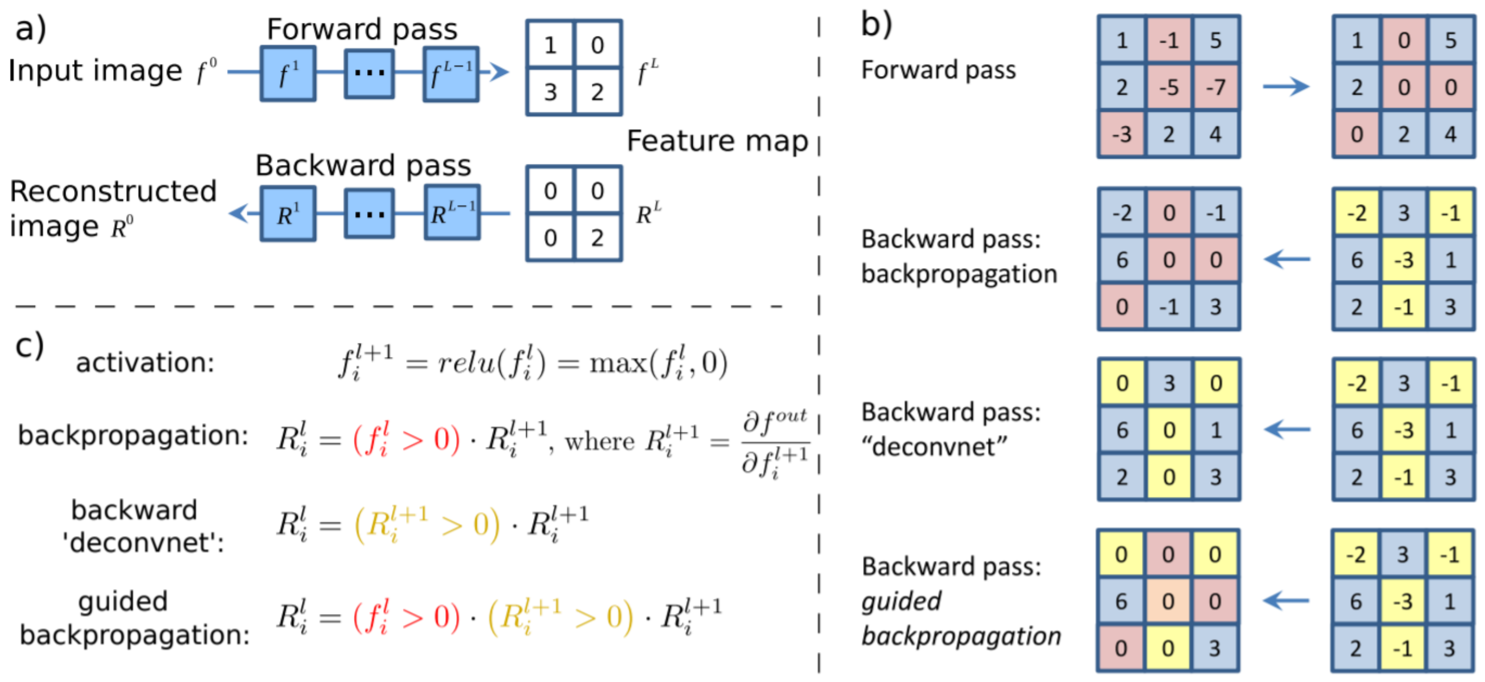
Saliency Maps with Gradient Based Backprop <cont>

The figure presents the activations of high layer neurons used by the algo:

a) Given an input image, perform the forward pass to the layer we are interested in, set to zero all activations except one and propagate back to the input image.

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: 'deconvnet' and guided backpropagation do not compute a true gradient but rather an imputed version).


a)
Forward pass
Input image f^0
f^1
f^{L-1}
f^L
Feature map
Backward pass
Reconstructed image R^0
R^1
R^{L-1}
R^L
b)
Forward pass
Backward pass: backpropagation
Backward pass: "deconvnet"
Backward pass: guided backpropagation
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': $R_i^l = (R_i^{l+1} > 0) \cdot R_i^{l+1}$

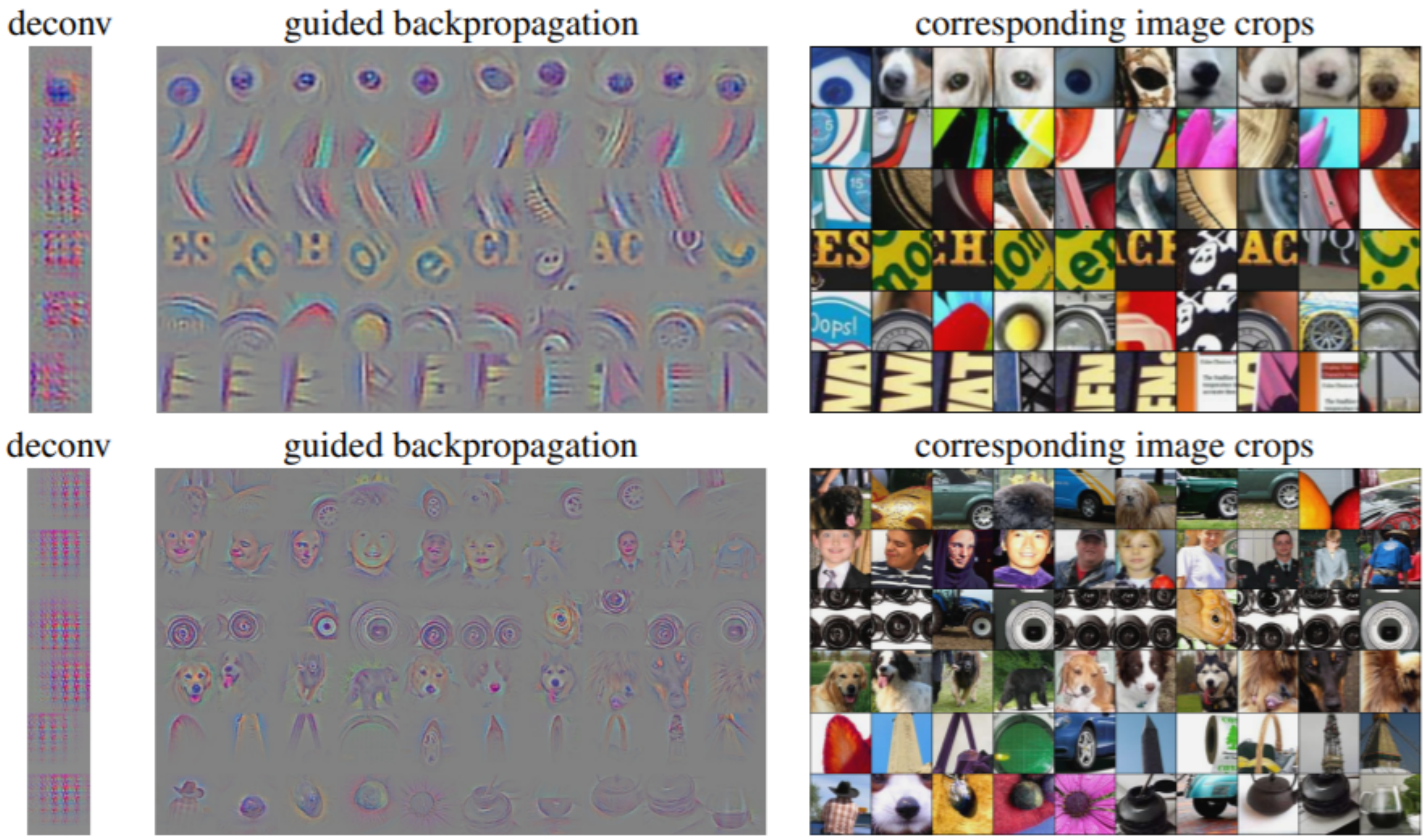
guided backpropagation: $R_i^l = (f_i^l > 0) \cdot (R_i^{l+1} > 0) \cdot R_i^{l+1}$

Springenberg et al., Striving for Simplicity, 2014

Saliency Maps with Gradient Based Backprop <cont>

Visualization of patterns learned by the layer conv6 (top) and layer conv9 (bottom) of the network trained on ImageNet. Each row corresponds to one filter.

The visualization using "guided backpropagation" is based on the top 10 image patches activating this filter taken from the ImageNet dataset.



Springenberg et al., Striving for Simplicity, 2014

Saliency Maps Recap

Saliency map is an interpretable technique to investigate hidden layers in CNNs. It is **a local gradient-based backpropagation interpretation method**, and it could be used for any arbitrary artificial neural network (**model-agnostic**).

However, some **limitations** of the method has been raised because:

- Saliency maps are not always reliable. Indeed, subtracting the mean and normalizations, can make undesirable changes in saliency maps as shown by Kindermans et al. 2018;
- Saliency maps ae vulnerable to adversarial attacks Ghorbani et al. 2019.

Outline

1: Communications

2: Needs for Transparent Models

3: Interpretation Methods for Image Classification

4: Saliency Maps

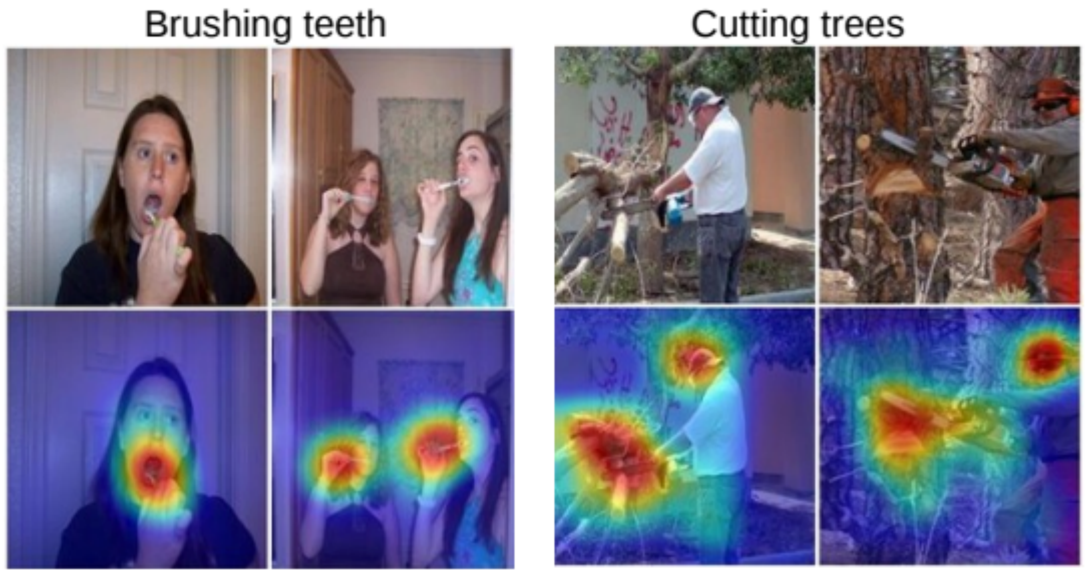
5: Class Activation Maximization (CAM)

6: Feature Visualization

Class Activation Maximization (CAM)

Class Activation Maximization is another explanation method for interpreting convolutional neural networks (CNNs) introduced by Zhou et al. 2016.

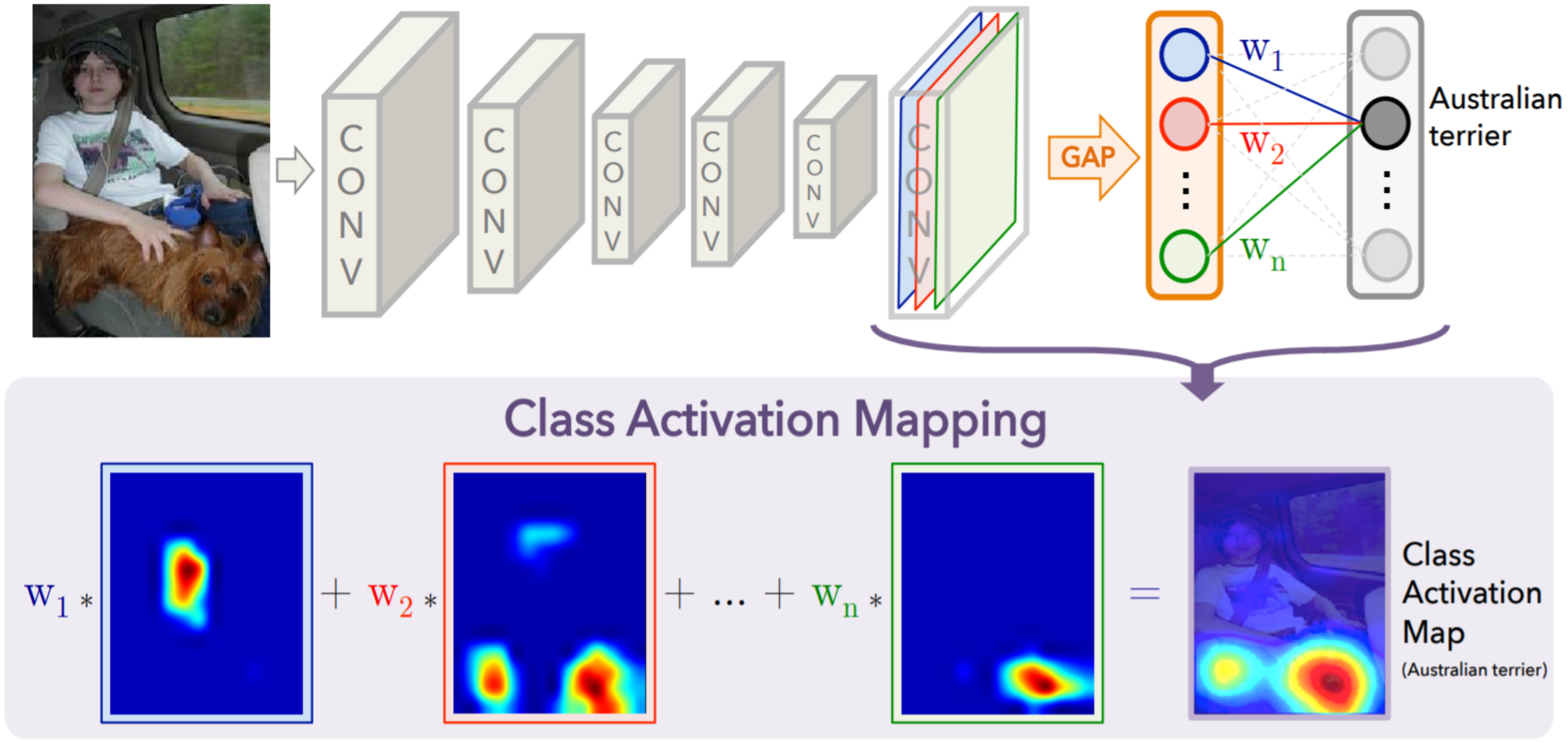
They proposed a network where the fully connected layers at the very end of the model has been replaced by a layer named **Global Average Pooling (GAP)** and combined with a class activation mapping (CAM) technique. .



Class Activation Maximization (CAM)

The GAP averages the activations of each feature map and concatenates and outputs them as a vector.

Then, a weighted sum of the resulted vector is fed to the final softmax loss layer.



Zhou et al., Learning Deep Features for Discriminative Localization, 2016

More on Class Activation Maximization

Other approaches to Class Activation Maximization have been developed by Selvaraju et al. 2016:

- **Grad-CAM**: is a more versatile version of CAM that can produce visual explanations for any arbitrary CNN, even if the network contains a stack of fully connected layers too (e.g. the VGG networks);
- **Guided Grad-CAM:** by adding an element-wise multiplication of guided-backpropagation visualization.

Class Activation Maximization (Grad-CAM)

Similarly to saliency maps:

1. we let the gradients of any target concept score flow into the final convolutional layer;

2. compute an importance score based on the gradients;

3. produce a coarse localization map highlighting the important regions in the image for predicting that concept.

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

More formally, at first, the gradient of the logits of the class c w.r.t the activations maps of the final convolutional layer is computed and then the gradients are averaged across each feature map to give us an importance score.

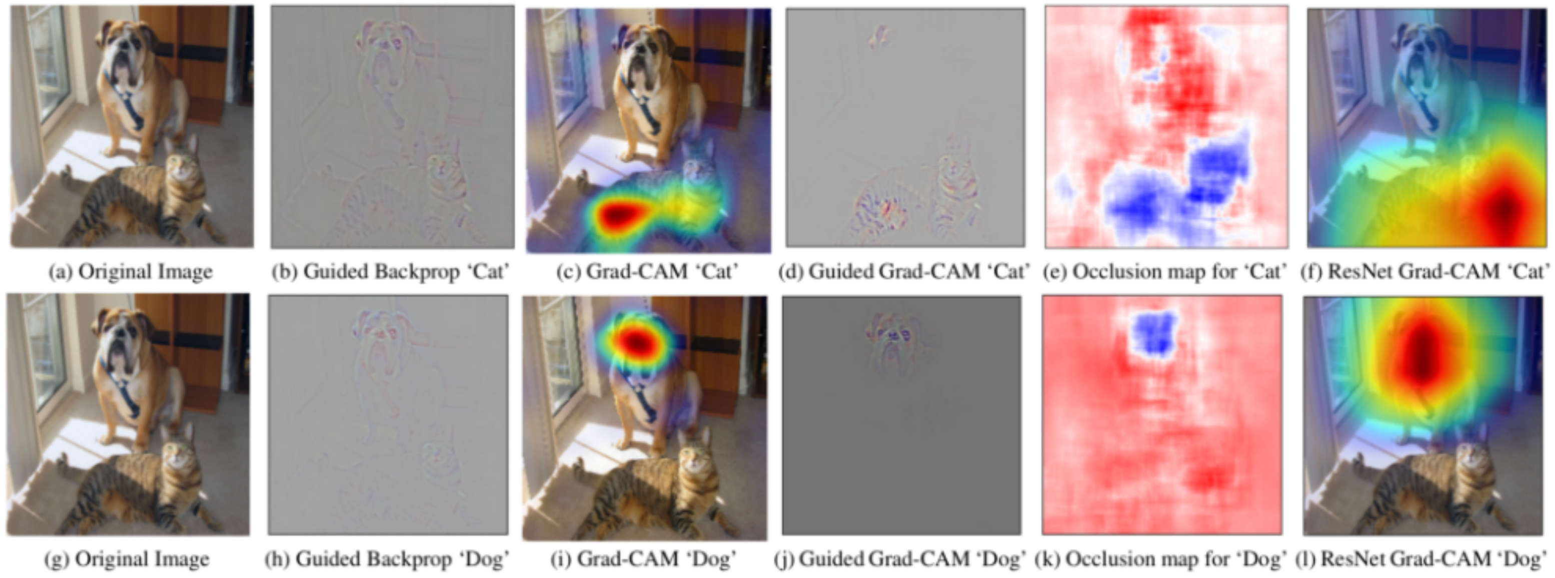
$$L_{\text{Grad-CAM}}^c = ReLU \underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$

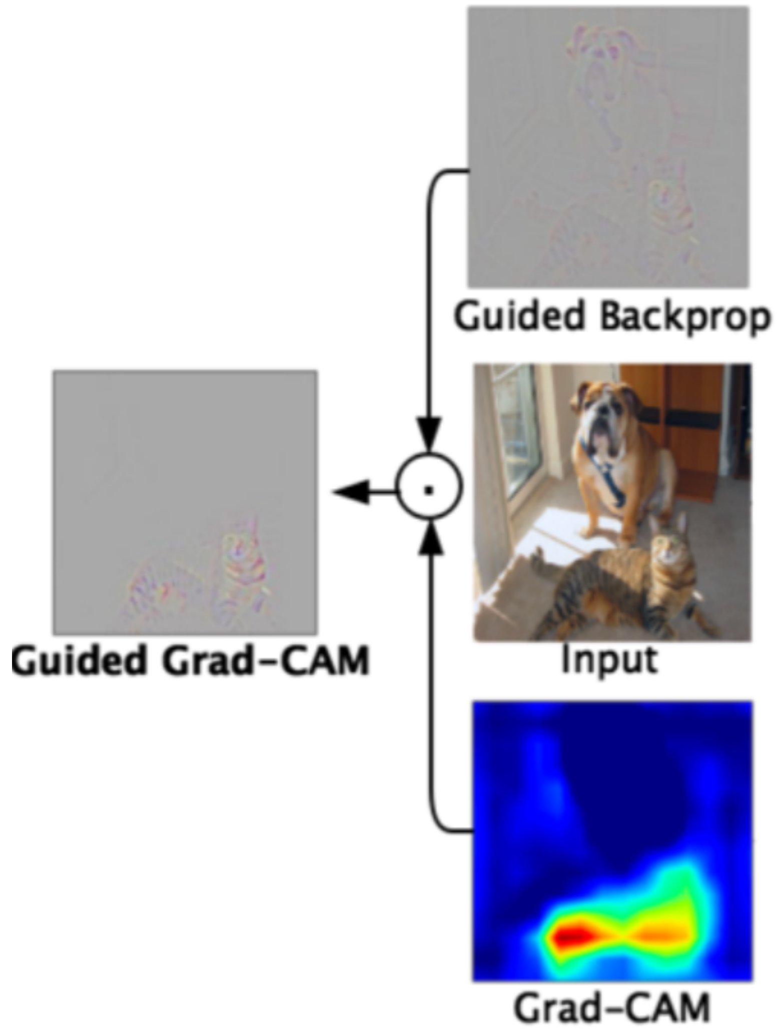
K: index of the activation map in the last convolutional layer;
c: is the class of interest;
α: computed above shows the importance of feature map *k* for the target class *c*.

Class Activation Maximization (Guided Grad-CAM)

As Grad-CAM can only produce coarse-grained visualizations, the authors have also combined guided-backpropagation with their method and propose Guided Grad-CAM.

They simply do that by an element-wise multiplication of guided-backpropagation visualization as shown in the figure.


(a) Original Image
(b) Guided Backprop 'Cat'
(c) Grad-CAM 'Cat'
(d) Guided Grad-CAM 'Cat'
(e) Occlusion map for 'Cat'
(f) ResNet Grad-CAM 'Cat'
(g) Original Image
(h) Guided Backprop 'Dog'
(i) Grad-CAM 'Dog'
(j) Guided Grad-CAM 'Dog'
(k) Occlusion map for 'Dog'
(l) ResNet Grad-CAM 'Dog'



Class Activation Maximization Recap

Both CAM and Grad-CAM are **local backpropagation-based** interpretation methods. They are **model-specific** as they can be used exclusively for interpretation of convolutional neural networks.

Outline

1: Communications

2: Needs for Transparent Models

3: Interpretation Methods for Image Classification

4: Saliency Maps

5: Class Activation Maximization (CAM)

6: Feature Visualization

Feature Visualization

There is a growing sense that neural networks need to be interpretable to humans. As it matures, two major threads of research have begun:

- **Feature visualization:** what a network is looking for by generating examples;
- **Attribution**: what part of an example is responsible for the network activating a particular way.

Next tutorial focus on feature visualization a) showing how to plot feature maps and b) how to visualize activation maximization using kearas.viz

Olah, et al., Feature Visualization, Distill, 2017

Feature Visualization

There is a growing sense that neural networks need to be interpretable to humans. As it matures, two major threads of research have begun:

- **Feature visualization:** what a network is looking for by generating examples;
- **Attribution**: what part of an example is responsible for the network activating a particular way.

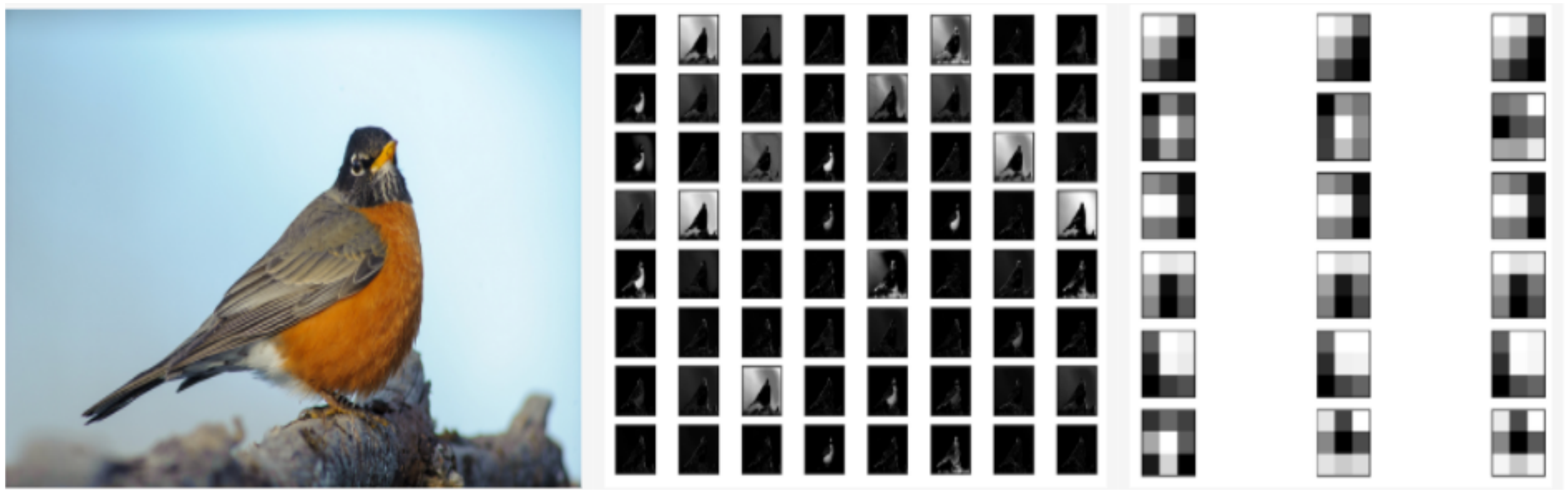
Next tutorial focus on feature visualization a) showing how to plot feature maps and b) how to visualize activation maximization using kearas.viz

Olah, et al., Feature Visualization, Distill, 2017

What to Visualize for CNNs?

The first things to try are:

1. Visualize the result of applying a learned filter to an image;
2. Visualize the filters themselves.

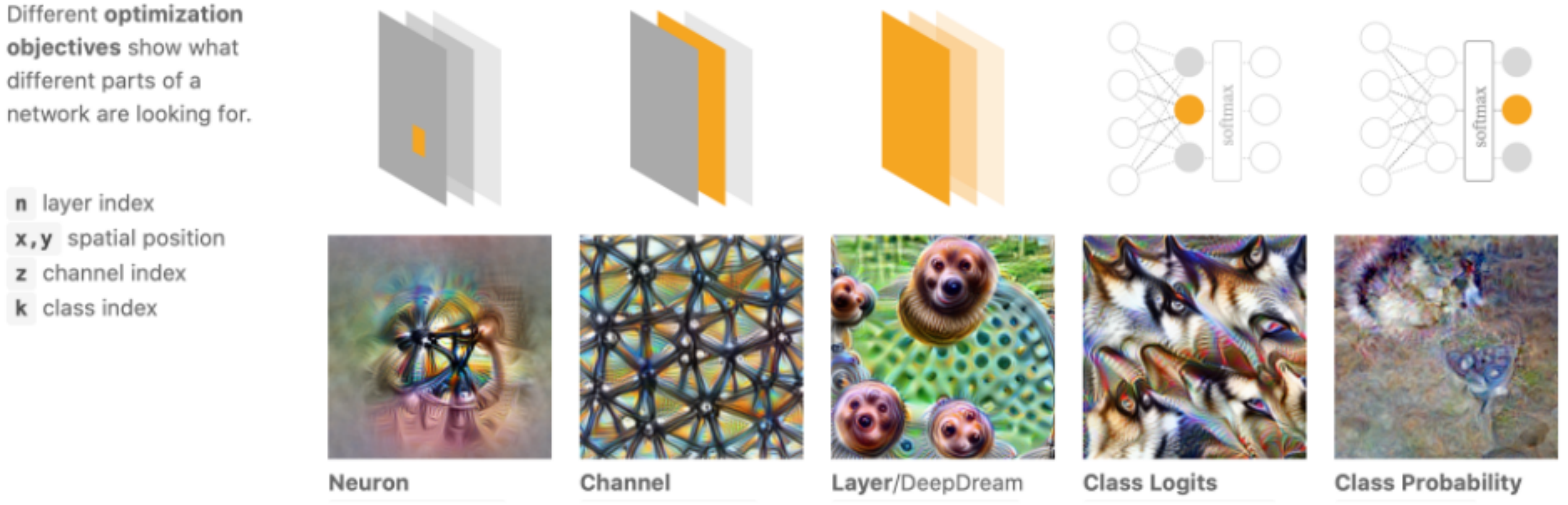


Unfortunately, these simple visualizations don't shed much light on what the model has learned (will see this in the notebook).

Activation Maximization

Rather than visualizing a particular filter or the representation of a particular image at a hidden layer, we can visualize the image that maximize the output and hence the impact of that specific filter or layer.

That is we find image x* that maximizes activation of a filter while holding the network weights fixed.

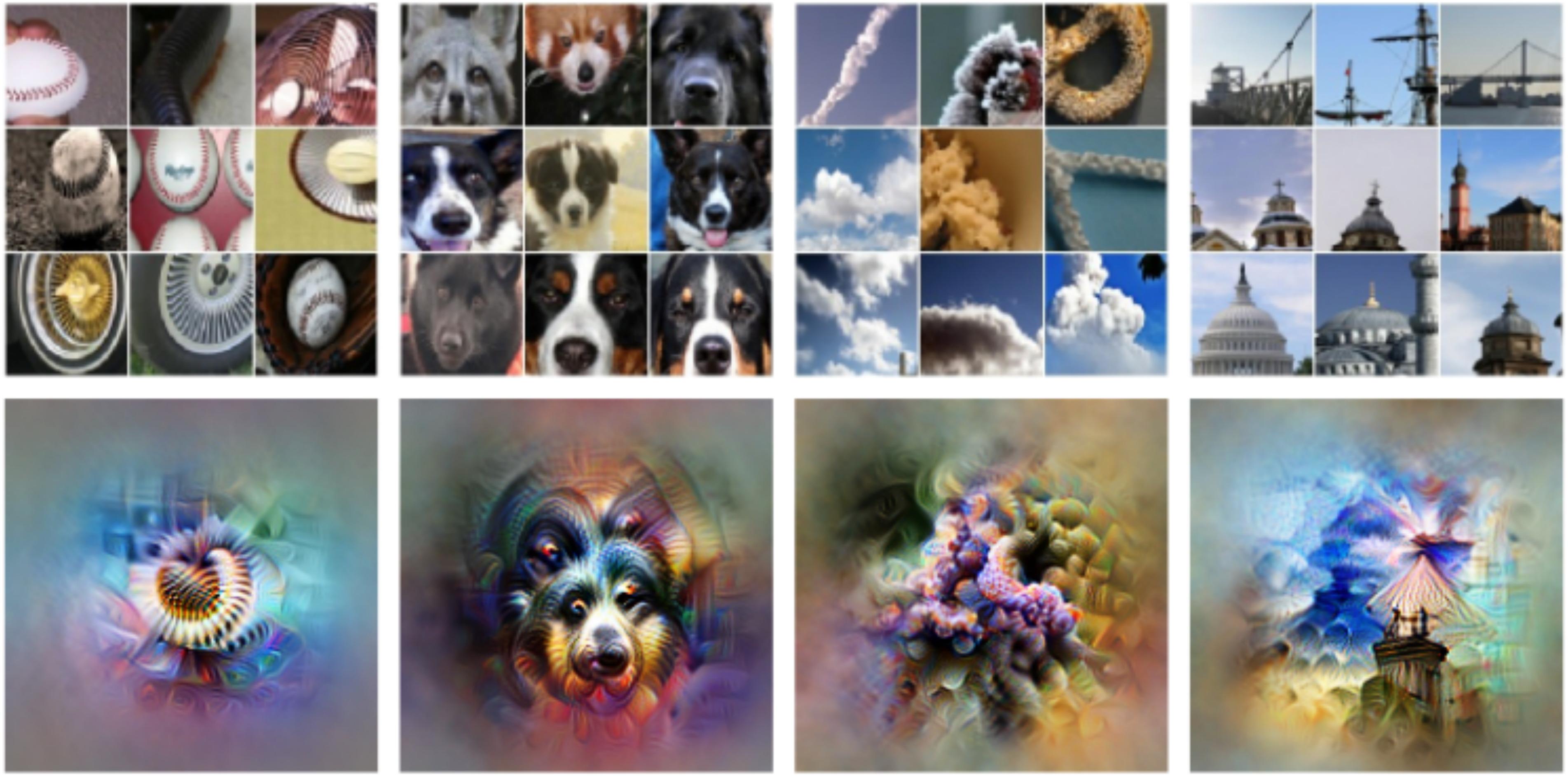


Olah, et al., [Feature Visualization](), Distill, 2017

Visualizing Convolutional Features by Activation Maximization

Dataset Examples show us what neurons respond to in practice

Optimization isolates the causes of behavior from mere correlations. A neuron may not be detecting what you initially thought.



Baseball—or stripes?

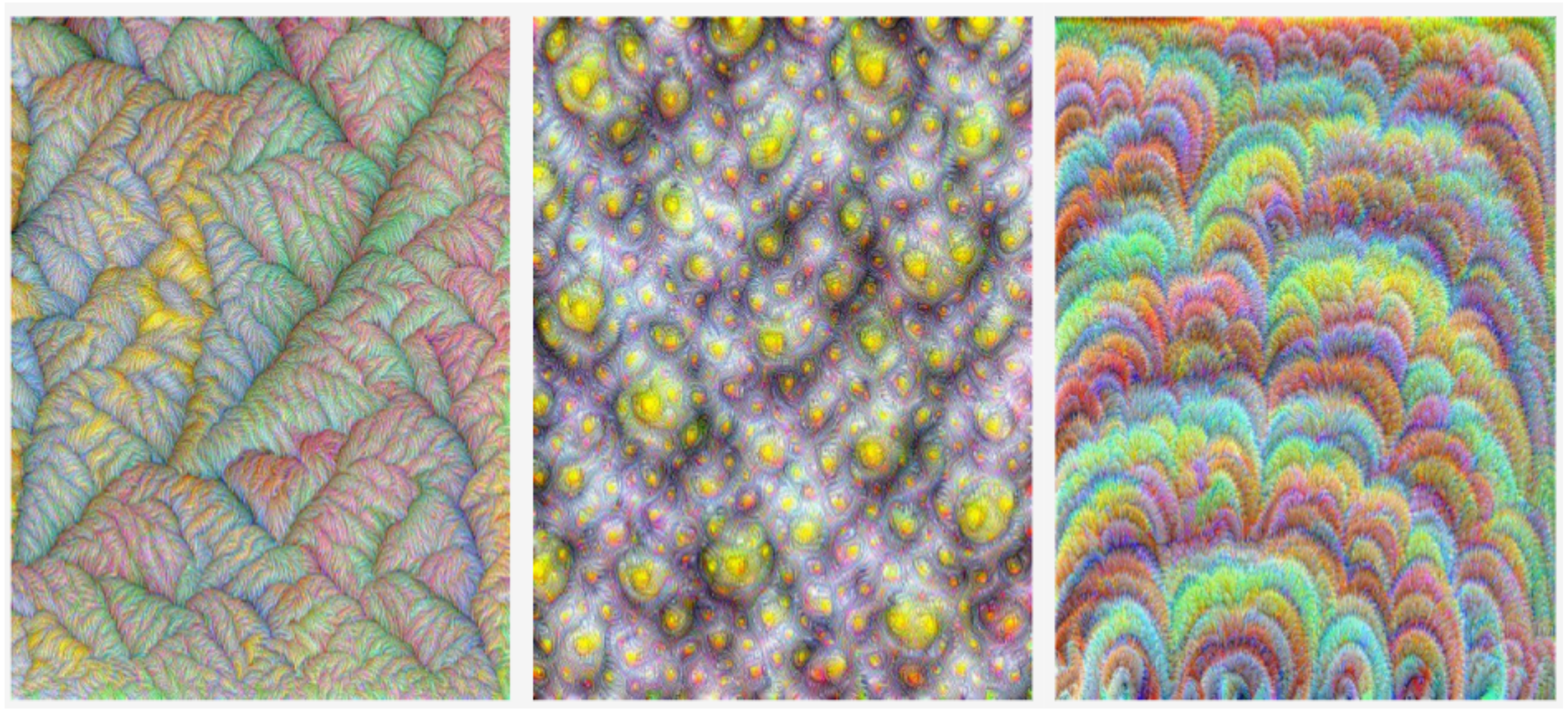
Animal faces—or snouts?

Clouds—or fluffiness?

Buildings—or sky?

Olah, et al., [Feature Visualization](), Distill, 2017

Visualizing Convolutional Filters by Activation Maximization



Yosinski, J., et al., [Deep Visualization Toolbox](), 2015

To the notebook!

Feature Visualization for CNNs

THANK YOU