Towards Automatic Concept-based Explanations

Published **@NeurIPS 2019**

Ghorbani A., Wexler J., Zou J.Y., Kim B.

Outline

- Is saliency a well defined problem?
- What are the Concept Activation Vectors?
- Towards concept-based explanations

Is saliency a well defined problem?

What is saliency for DNNs?

3 different categories…
They all do the same…
Infer insights about the model by ranking the **input** features

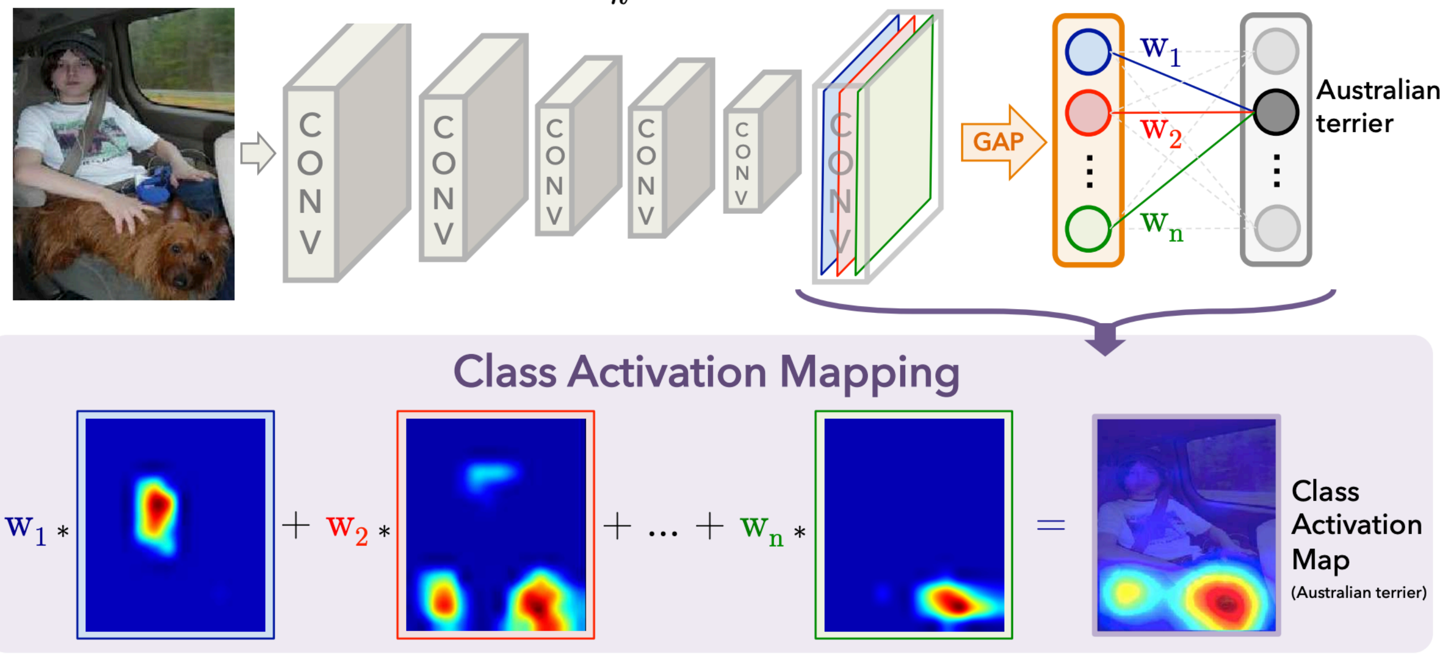
3 default axioms:

1. Completeness
2. Implementation invariance
3. Sensitivity

Class Activation Maps (CAM), Zhou et al., 2016

Idea: Project back the weights of the output layer on to the convolutional feature maps

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



Grad-CAM,
Selvaraju et al., 2017

Idea: Don't use weights and activations,
use the gradients.

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

logits

“The (un)reliability of saliency methods”, Kindermans et al., 2017

A **simple** input **transformation** causes most **saliency** methods to **fail**!

1 New axiom:

input invariance

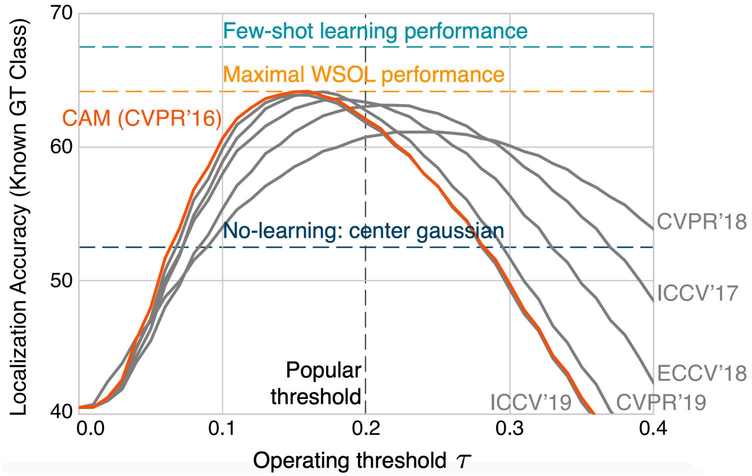
“Local explanation methods for deep neural networks lack sensitivity to parameter values”, Adebayo et al., 2018

“DNNs with **randomly-initialized** weights produce **explanations** that are both visually and quantitatively **similar** to those produced by DNNs with **learned weights**”

“Evaluating Weakly Supervised Object Localization Methods Right”, Choe et al., 2020

Insignificant improvements since Zhou et al., **2016** !!!

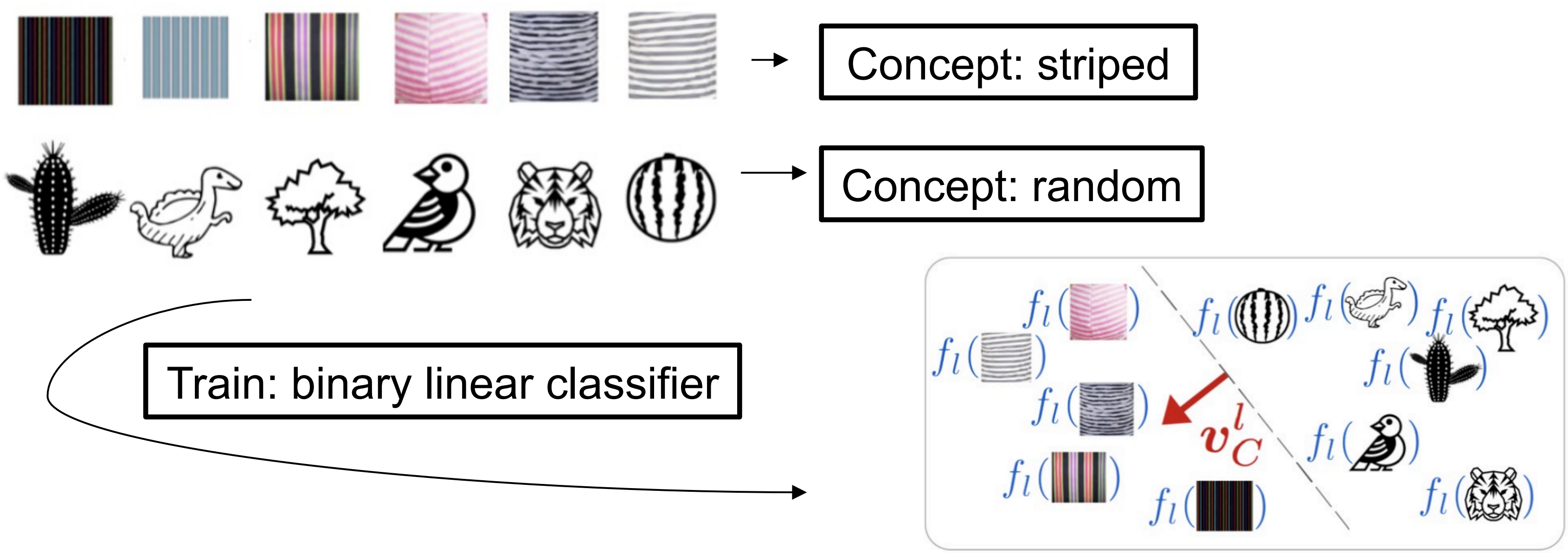
It’s all about **hyper-parameter tuning**!



Concept Activation Vectors (TCAV), Kim et al., 2018

What are Concept Activation Vectors (CAVs)?

It is the normal to a hyperplane separating examples with and without a concept.



Concept Activation Vectors (TCAV), Kim et al., 2018

Idea:

Project the **derivatives** to the **direction** of the **concept**

$$\underbrace{S_{C,k,l}(\boldsymbol{x})}_{} = \overbrace{\nabla h_{l,k}(f_l(\boldsymbol{x})) \cdot \boldsymbol{v}_C^l}^{}$$

$S_{C,kl}(x)$ can **quantitatively** measure the **sensitivity** of model predictions with respect to **concepts**

Concept Activation Vectors (TCAV), Kim et al., 2018

Testing with CAVs (TCAV):

$$\text{TCAVQ}_{C,k,l} = \frac{|\{\boldsymbol{x} \in X_k : S_{C,k,l}(\boldsymbol{x}) > 0\}|}{|X_k|}$$

The Fraction of k-class inputs whose l-layer activation vector was positively influenced by the concept C.

i.e. the average positive effect of a concept

Concept Activation Vectors (TCAV), Kim et al., 2018

With TCAVs we can:

- Sort images with respect to their relation to the concept
- Reveal biases
- See which layer learns which concept

Drawbacks:

- The user must specify the concept (this can be quite vague)
- Introduces human bias in the explanation process

"Towards Automatic Concept-based Explanations", Ghorbani et al., 2019

Concept-based Explanation **Desiderata**:

1. **Meaningfulness:** An **example** of a concept is semantically **meaningful on its own**.

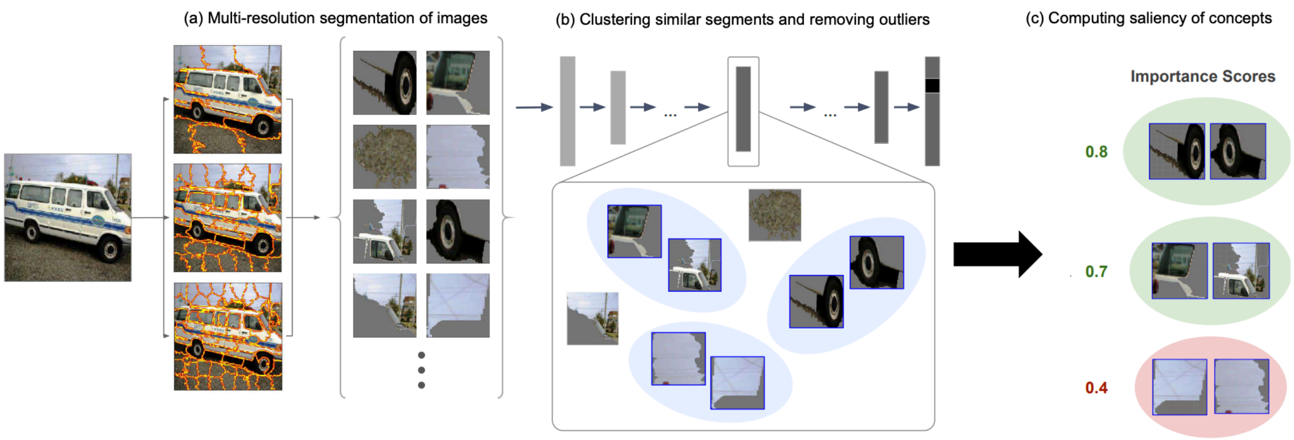
2. **Coherency:** Examples of a concept should be **perceptually similar** to each other and **dissimilar** from examples of other concepts.

3. **Importance:** A concept is "important" for the prediction of a class if its **presence** is **necessary** for the true **prediction** of samples in that class.

“Towards Automatic Concept-based Explanations”, Ghorbani et al., 2019

Explanations in 3 steps:

1. Image **segmentation** using different scales.
2. **Clustering** of similar **segments** as examples of the same **concept**.
3. Testing with Concept Activation Vectors (**TCAVs**).

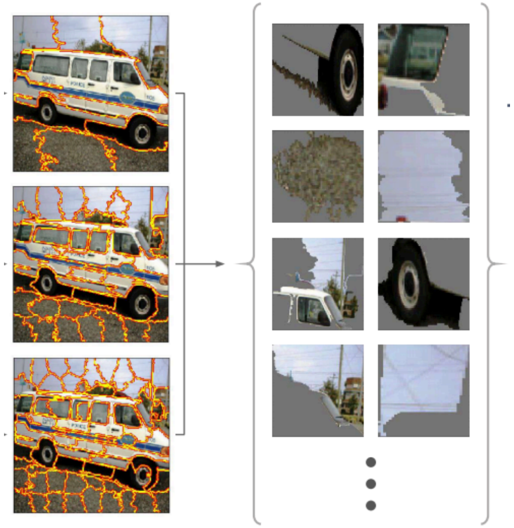


“Towards Automatic Concept-based Explanations”, Ghorbani et al., 2019

1. Image **segmentation** using different scales.

Procedure:

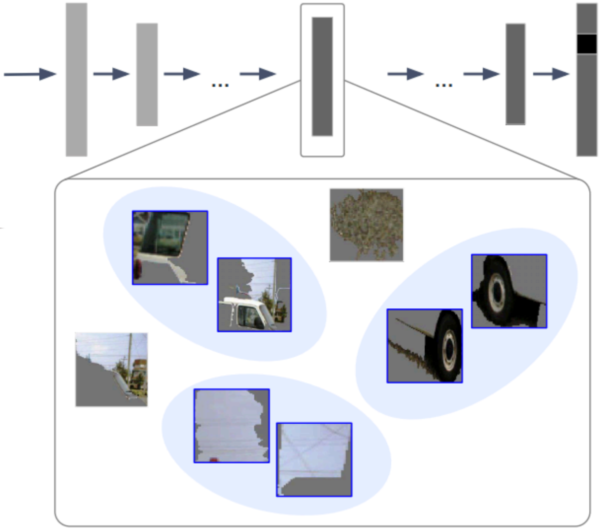
i. Take all **images** from a **class**.
ii. Rescale them to **3** different **resolutions**.
iii. Use SLIC to get **segments**.



“Towards Automatic Concept-based Explanations”, Ghorbani et al., 2019

2. **Clustering** of similar **segments** as examples of the same **concept**.

Procedure:
i. Take a model **pretrained** on ImageNet.
ii. Compute the segment’s **activations** at mid-high level layers*.
iii. Do **K-means** clustering (Euclidean distance**) of the **segments**.
iv. Remove **outliers**.



* Earlier layers are better at similarity of textures and colors while latter ones are better for object.

** The Euclidean distance in the activation space of final layers is an effective perceptual similarity metric.

“Towards Automatic Concept-based Explanations”, Ghorbani et al., 2019

3. Testing with Concept Activation Vectors (**TCAVs**).

Procedure:

i. Take all the **clusters**.

ii. Treat them as **concepts**.

iii. Apply **relative TCAVs**:

Train the binary classifier using a **1-vs-all*** setting.

* Use one concept as primary and the rest as random images.

“Towards Automatic Concept-based Explanations”, Ghorbani et al., 2019

“Towards Automatic Concept-based Explanations”, Ghorbani et al., 2019

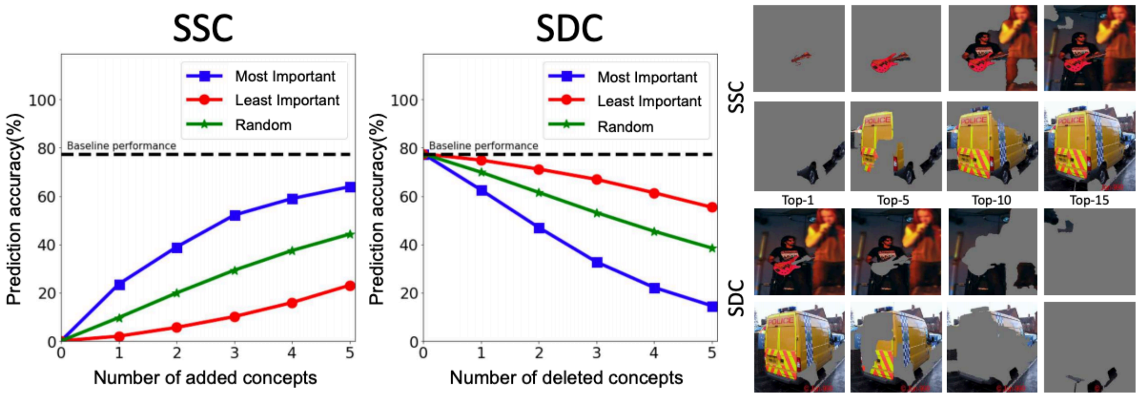


Figure 4: **Importance** For 1000 randomly sampled images in the ImageNet validation set, we start removing/adding concepts from the most important. As it is shown, the top-5 concepts is enough to reach within 80% of the original accuracy and removing the top-5 concepts results in misclassification of more than 80% of samples that are classified correctly. For comparison, we also plot the effect of adding/removing concepts with random order and with reverse importance order.

“Towards Automatic Concept-based Explanations”, Ghorbani et al., 2019

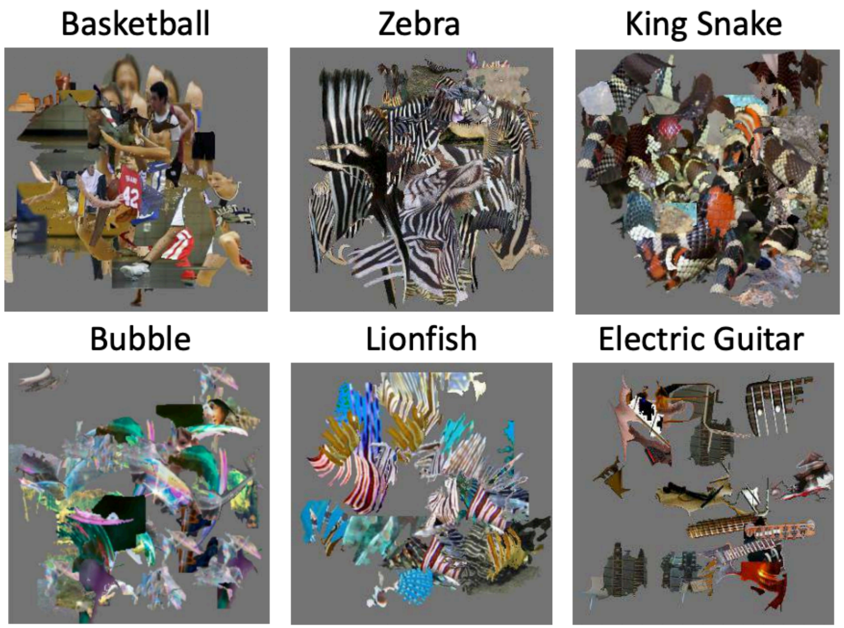


Figure 6: **Stitching important concepts** We test what would the classifier see if we randomly stitch important concepts. We discover that for a number classes this results in predicting the image to be a member of that class. For instance, basketball jerseys, zebra skin, lionfish, and king snake patterns all seem to be enough for the Inception-V3 network to classify them as images of their class.

"Towards Automatic Concept-based Explanations", Ghorbani et al., 2019

- Why do they test only on ImageNet?

  Feature extraction etc. are using ImageNet
- The human experiments are not so well designed.

  e.g. clustered segments vs random ones
- What if we change the the K in K-means?

  They use K=25
- What if we remove/add more scales?
- Inherits all the bad aspects of:
  segmentation, clustering, similarity metric, TCAVs.

  = the method is too noisy
- What happened to the Implementation invariance?

Thank you